

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

В І С Н И К

НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Видається з 1964 р.

№ 744

КОМП’ЮТЕРНІ НАУКИ
ТА ІНФОРМАЦІЙНІ
ТЕХНОЛОГІЇ

Відповідальний редактор – д-р техн. наук, професор Ю.М. Рашкевич

Львів
Видавництво Львівської політехніки
2012

Вісник Національного університету “Львівська політехніка” “Комп’ютерні науки та інформаційні технології” входить до переліку видань ВАК, в яких друкуються матеріали дисертаційних робіт у галузі технічних наук

У віснику публікуються статті за результатами досліджень з актуальних питань комп’ютерних наук та інформаційних технологій, виконаних професорсько-викладацьким складом Національного університету “Львівська політехніка” та провідними вченими України і зарубіжними вченими в галузі проектування архітектури і компонентів комп’ютерних систем, моделювання складних об’єктів, процесів і систем та розробки і використання новітніх інформаційних технологій.

Для науковців, викладачів вищих закладів освіти, інженерів, що спеціалізуються у сфері новітніх обчислювальних систем, мереж, комп’ютеризованих засобів розв’язання задач цифрового опрацювання сигналів і зображень, автоматизованого проектування та керування, а також докторантів, аспірантів та студентів старших курсів відповідних спеціальностей.

***Рекомендувала Вчена рада Національного університету “Львівська політехніка”
(Протокол № 51 засідання від 27.03.2012 р.)***

***Свідоцтво про державну реєстрацію друкованого засобу масової інформації
Серія КВ №13038-1922Р від 20.07.2007 р.***

Редакційна колегія:

д-р техн. наук, проф. Ю.М. Рашкевич (відп. редактор);
д-р техн. наук, проф. Р.А. Мельник (заст. відп. редактора);
д-р фіз-мат. наук, проф. Я.П. Драган;
д-р техн. наук, проф. Р.П. Базилевич;
д-р техн. наук, проф. М.О. Медиковський
д-р техн. наук, проф. В.В. Пасічник;
д-р техн. наук, проф. В.В. Різник;
д-р техн. наук, проф. Д.В. Федасюк;
д-р техн. наук, проф. Р.О. Ткаченко;
д-р техн. наук, проф. М.В. Лобур;
д-р техн. наук, проф. Д.Д. Пелешко (відп. секретар)

Адреса редколегії:

***Національний університет “Львівська політехніка”
вул. С. Бандери, 12, Львів–13, 79013,
корп. № 5, кімн. 515.
akm805@ukr.net***

АРХІТЕКТУРА ТА КОМПОНЕНТИ КОМП'ЮТЕРНИХ СИСТЕМ

УДК 681.3, 621.3

І. Пастернак

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ПАРАМЕТРИЗОВАНІ МЕРЕЖНІ ІНТЕРФЕЙСИ

© Пастернак І., 2012

Запропоновано варіант клієнт-серверної взаємодії на основі віддалених об'єктів у глобальній мережі, а також запропоновано використання та класифікацію шаблонів під час побудови та параметризації мережних інтерфейсів.

Ключові слова: віддалений об'єкт, шаблон, клієнт-серверна взаємодія, параметризований мережний інтерфейс.

The variant of client-server interaction based on distant objects in the global network. Also proposed use and classification of patterns in the construction and parameterization of network interfaces.

Key words: remote object, pattern, client-server interaction parameter network interface.

Вступ

Використання віддалених об'єктів в мережі за взаємодії клієнт-сервер дуже складно, особливо якщо їх потрібно використовувати повторно. Необхідно підібрати відповідні віддалені об'єкти, віднести їх до різних класів, зберігаючи ступінь деталізації, визначити параметризацію інтерфейсів і ієрархію наслідування і встановити відносини між параметрами. Параметризований інтерфейс повинен, з одного боку, відповідати задачі, яка розв'язується, а з іншого – бути загальним, щоб вдалося врахувати всі вимоги, які можуть виникнути в майбутньому. Розробники скажуть вам, що забезпечити «правильний», тобто достатньою мірою гнучкий та придатний для повторного використання параметризований мережний інтерфейс, з першого разу важко, якщо взагалі можливо. Перед тим як вважати мету досягнутою, вони зазвичай намагаються випробувати знайдене рішення на декількох задачах, і кожного разу модифікують його завдяки параметризації мережного інтерфейсу.

І все ж таки розробникам вдається створити параметризований мережний інтерфейс системи. Водночас користувачі вражені кількістю можливих варіантів і часто повертаються до звиклих не об'єктно-орієнтованих методик. Проходить багато часу перед тим, як стає зрозуміло, що краще використовувати об'єктно-орієнтований параметризований мережний інтерфейс через можливість зміни параметрів, які в результаті можуть змінити вигляд цього інтерфейсу [1, 3]. Завдяки цьому розробнику зрозуміло, що не потрібно розв'язувати кожну нову задачу з самого початку. Замість цього він намагається повторно скористатися тими рішеннями, котрі виявилися вдалими в минулому. Відшукавши одне рішення відразу, він використовуватиме його знову і знову. В багатьох об'єктно-орієнтованих системах ви зустрінете шаблони, які повторюються і які складаються з класів і взаємодіючих об'єктів. Саме враховуючи їх, можна параметризувати мережний інтерфейс.

З використанням шаблонів розробники програмного забезпечення вирішують конкретні задачі, внаслідок чого об'єктно-орієнтований вигляд параметризованого мережного інтерфейсу стає більш

гнучким, ефективним, і його можна застосовувати повторно. Розробник програмного забезпечення, який знайомий з шаблонами, може зразу використовувати їх для розв'язання нової задачі, не намагаючись кожного разу винайти щось нове і знову параметризувати мережний інтерфейс.

Огляд літературних джерел

За рахунок параметризації мережного інтерфейсу за клієнт-серверної взаємодії клієнт з сервером можуть обмінюватися віддаленими об'єктами в глобальній мережі. Розглянемо кроки, необхідні, аби дозволити клієнту отримати доступ до об'єкта на віддаленому сервері:

- Створити TCP або HTTP-з'єднання між клієнтом і сервером через параметризований мережний інтерфейс.
- Вибрати повідомлення, які відправлені від клієнта до сервера та відформатувати їх.
- Реєструвати типи, які будуть доступні у віддаленому режимі.
- Створити віддалений об'єкт та активувати його з сервера або клієнта.

Ви не повинні розуміти, як реалізувати параметризований мережний інтерфейс, чи знати, як працюють протоколи TCP, HTTP, або знати порти – треба просто вказати тип з'єднання і який порт використовувати, що можна задати автоматично. Якщо вибрано протокол http, для з'єднання необхідно використовувати Simple Object Access Protocol (SOAP) в бінарному форматі, який використовує TCP. За замовчуванням SOAP використовується тоді, коли вибраний протокол HTTP і TCP тоді реалізується бінарне форматування. Хоча це і не так ефективно, як у двійковому форматі, комбінація SOAP і HTTP стала стандартом де-факто для передавання даних через брандмауери, який використовується для реалізації параметризованого мережного інтерфейсу. Двійковий формат рекомендується у випадках, коли брандмауер не є проблемою. Віддалені об'єкти створюються з класів. Вони знаходяться на віддаленому хості, щоб реалізувати канал для з'єднання клієнта з сервером та передачу віддаленого об'єкта, до яких вони надають доступ [2].

Додаток може підтримувати кілька режимів ввімкнення і кілька віддалених об'єктів. Клієнт вказує на об'єкт, який він хоче отримати, маючи доступ до сервера і використовувати або клієнт, або сервер активації режиму. А також клієнт вибирає метод реєстрації і передає декілька його параметрів, які визначають адресу сервера і тип (клас), який буде доступний. Сервер реєструє типи з'єднання і порти, які він зробить доступними для клієнтів. Сьогодні є чимало прикладів коду, метою якого є презентація прототипів, які можна використовувати для реалізації широкого спектра віддалених додатків. Наведено приклади сервера, який отримує повідомлення і відправляє їх клієнту за запитом. Коли клієнт намагається викликати метод віддаленого об'єкта, його виклик проходить крізь кілька шарів на боці клієнта. Перший з них являє собою проксі-абстрактний клас, який має такий самий параметризований мережний інтерфейс, як віддалений об'єкт, який він представляє. Це підтверджує, що кількість і тип аргументів у виклику вірні, упаковує запит в повідомленні і передає його клієнту через канал. Канал відповідає за транспортування запиту на віддалений об'єкт. Канал складається з приймача, що серіалізує запит в потік і транспортує клієнту, який фактично передає запит в порт на сервері.

Сервер, з іншого боку, вказує, які об'єкти він хоче зробити доступними для видалених клієнтів і активізує режим, необхідний для доступу до них. Це робиться за допомогою механізму, відомого як тип реєстрації. Процес реєстрації відбувається на сервері і клієнті. У своїй найпростішій формі реєстрація на хост полягає у створенні екземпляра об'єкта каналу та його реєстрації передаванням як параметра статичного методу відповідного класу. Цей приклад реєструє канал HTTP на порт 3200. Є кілька правил, щоб реєструвати канал для взаємодії клієнта з сервером:

- І сервер, і клієнт можуть зареєструвати кілька каналів, однак клієнт повинен зареєструвати канал, який відповідає одному хосту.
- Кілька каналів не можуть використовувати той самий порт.
- За замовчуванням, HTTP і TCP протоколам присвоюються імена HTTP і TCP відповідно. За спроби зареєструвати кілька HTTP або TCP-канали з іменем за замовчуванням ви отримаєте виняток. Способом вирішення цієї проблеми є створення каналів за допомогою конструктора, який приймає параметр «ім'я» каналу.

Альтернативою є вкладений канал, протокол якого дозволяє йому бути зазначеним у файлах конфігурації, пов'язаний з клієнтом і хост-вузлом. Коли об'єкт стану віддаленого об'єкта за значенням отримує копію об'єкта у власному домені додатку, він може працювати з об'єктом на місцевому рівні і не потребує проксі-сервера. Такий підхід не є ефективнішим, ніж стан віддаленого об'єкта за посиланням, де всі запити є на віддаленому об'єкті. Тим не менш виклик віддалених об'єктів, які призначені для запуску на клієнтських комп'ютерах, реалізувати простіше, ніж на сервері, і це може зменшити витрати часу клієнтських запитів.

Постановка задачі

Реалізувати клієнт-серверну взаємодію за допомогою параметризованого мережного інтерфейсу на основі шаблонів.

Основні результати досліджень

Для ефективної клієнт-серверної взаємодії в глобальній мережі використовують параметризований мережний інтерфейс, тобто мережний інтерфейс, який параметризується і наслідується за допомогою класів з використанням шаблонів (рис. 1).



Рис. 1. Схематичне зображення клієнт-серверної взаємодії з використанням параметризованого мережного інтерфейсу

Коли клієнт надсилає запит на віддалений об'єкт на сервері, доступ до нього він отримує через мережний інтерфейс. Мережний інтерфейс, своєю чергою, можна параметризувати, щоб спростити взаємодію клієнта з сервером через віддалені об'єкти. А саме змінюючи один параметр з використанням параметризованого мережного інтерфейсу можна вирішувати різні поставлені завдання, які потрібно застосовувати під час з'єднання клієнта з сервером в глобальній мережі. Для ефективної параметризації і наслідування мережного інтерфейсу використовують шаблони. Клас взаємодії і форматування інстанціоє шаблони, які переводить у певний формат.

Програмуючи параметризований мережний інтерфейс, важливо знати, що шаблони відрізняються від класів за ступенем деталізації і рівнем абстракції і повинні бути якимось чином організовані у параметризований мережний інтерфейс. Будемо класифікувати шаблони за двома критеріями (див. таблицю). Першим критерієм буде призначення шаблону. У зв'язку з цим виділяють породжувальні шаблони, структурні шаблони і шаблони поведінки. Перші пов'язані з процесом створення об'єктів. Другі мають відношення до композиції об'єктів та класів. Шаблони поведінки характеризують те, як класи чи об'єкти взаємодіють між собою.

Класифікація шаблонів

Мета\Рівень	Породжувальні шаблони	Структурні шаблони	Шаблони поведінки
Клас	Фабричний метод	Адаптер (класа)	Інтерпретатор
			Шаблонний метод
Віддалений об'єкт	Абстрактна фабрика	Адаптер (об'єкта)	Ітератор
	Одинак	Декоратор	Команда
	Прототип	Заступник	Спостерігач
	Будівник	Компонувальник	Відвідувач
		Міст	Посередник
		Пристосованець	Стан
		Фасад	Стратегія
			Хранитель
			Ланцюжок зобов'язань

Другий критерій – рівень – говорить про те, для чого застосовуються шаблони: до віддалених об'єктів чи до класів. Шаблони рівня класів описують відношення між класами та підкласами. Такі відношення виражаються за допомогою наслідування, тому вони статичні, тобто зафіксовані на етапі компіляції. Шаблони рівня об'єктів описують відношення між об'єктами, котрі можуть змінюватися під час виконання і тому динамічніші. Майже усі шаблони якоюсь мірою використовують наслідування [5]. Тому до категорії «шаблони класів» віднесено тільки ті, які сфокусовані лише на відношеннях між класами. Зверніть увагу: більшість шаблонів діють на рівні віддалених об'єктів.

Породжувальні шаблони класів частково делегують відповідальність за створення віддалених об'єктів своїм підкласам, своєю чергою, породжувальні шаблони віддалених об'єктів передають відповідальність іншому віддаленому об'єкту. Структурні шаблони класів використовують наслідування для вкладеності класів, тоді як структурні шаблони віддалених об'єктів описують способи складання віддалених об'єктів з частин. Поведінкові шаблони класів використовують наслідування для опису алгоритмів і потоку управління, а поведінкові шаблони віддалених об'єктів як об'єкти, які належать деякій групі, разом функціонують і виконують задачу, котра жодному окремому віддаленому об'єкту не під силу.

Взаємодію клієнта з сервером через параметризований мережний інтерфейс на основі шаблонів можна реалізувати через:

- віддалений об'єкт ;
- клієнт активованих об'єктів;
- сервер активованих об'єктів.

Стан віддаленого об'єкта за посиланням використовується, коли клієнт робить запит на об'єкт, який працює на віддаленому сервері. Віддалені об'єкти отримують стан і повинні наслідувати клас. Активізують режим віддаленого об'єкта за одним з двох варіантів: клієнт активованих об'єктів і сервер активованих об'єктів, також зазвичай називають відомі віддалені об'єкти. Це залежить від клієнта: він обирає, який з варіантів використовувати. Якщо клієнт вибирає сервер активованих об'єктів, то підтверджує використання активованого сервером одного запиту або активованого сервером одного з віддалених об'єктів. Вибір активації режиму впливає на загальний вигляд мережного інтерфейсу, продуктивність і масштабованість віддаленого додатку. Він визначає, коли віддалені об'єкти створюються, скільки цих об'єктів створено, як керувати їхнім життєвим циклом і чи підтримуватиме стан віддалених об'єктів інформацію, яку вони містять. На рис. 2 показано взаємодію клієнта з сервером через параметризований мережний інтерфейс за допомогою віддаленого об'єкта.

Коли клієнт намагається спілкуватися з віддаленим об'єктом через параметризований мережний інтерфейс, його посиланням на віддалений об'єкт насправді займається посередник, відомий як сервер активованих об'єктів. Є два типи активації режиму віддалених об'єктів: сервери активованих об'єктів, коли клієнт зв'язується з віддаленими об'єктами напряму, і реальний клієнт активованих об'єктів, коли клієнт приймає запит і пересилає його на віддалений об'єкт [4]. Сервер активованих об'єктів активізує

параметризований мережний інтерфейс для клієнта, який збігається з класом. Активізовані об'єкти переконуються, що всі клієнтські запити відповідають заданому цільовому методу, тобто тип і кількість параметрів збігаються. Розробник відповідає за забезпечення метаданими, які визначають клас віддаленого об'єкта, доступні під час компіляції та виконання.

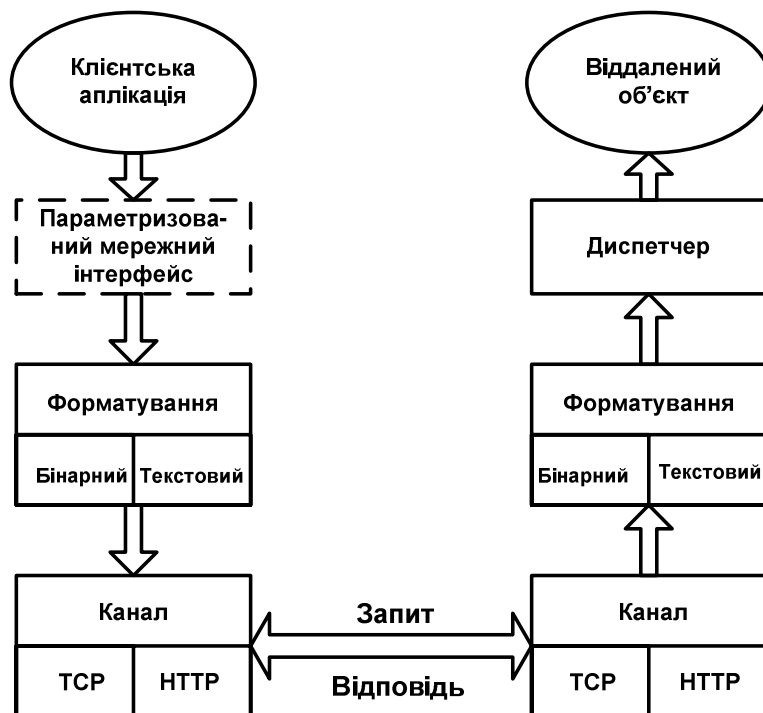


Рис. 2. Блок-схема взаємодії клієнта з сервером через параметризований мережний інтерфейс за допомогою віддаленого об'єкта

Найпростіший спосіб полягає в наданні клієнту копії віддаленого об'єкта на сервері, що містить клас. Після того сервер активованих об'єктів упакує запит у повідомлення як об'єкт класу, який реалізує параметризований мережний інтерфейс. Повідомлення об'єкта класу передається як параметр для виклику реальних методів, які передають його каналом мережного з'єднання. Там форматується об'єкт, серіалізуються повідомлення і передає його в об'єкт каналом мережного з'єднання, що фізично відправляє повідомлення на віддалений об'єкт.

Використання і поведінка клієнта активізованого об'єкта нагадують, що об'єкт створений локально. Обидва вони можуть бути створені за допомогою нового оператора, як можна використати параметризовані конструктори на додаток за замовчуванням і як зберегти інформацію про стан властивостей або полів. Як показано на рис. 3, вони відрізняються тим, що сервер активованого об'єкта працює на хості в окремому домені програми і називається проксі-сервером.

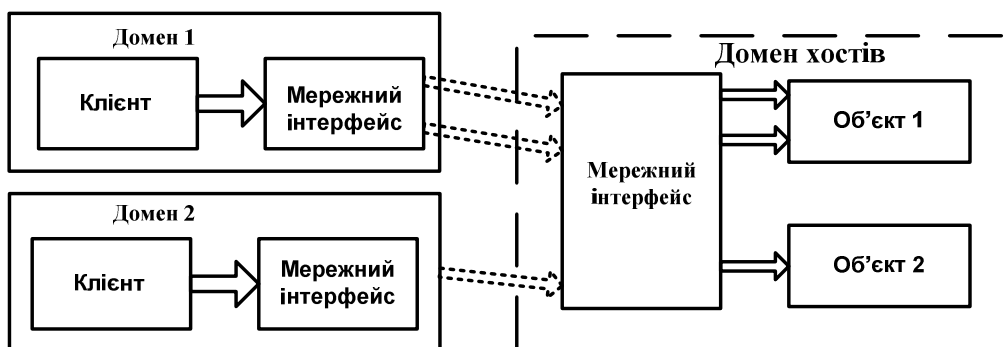


Рис. 3. Спрощена схема вигляду клієнта активованих об'єктів, коли клієнт зберігає контроль над об'єктом

Той факт, що віддалений об'єкт знаходиться в іншому домені, означає, що він просто існує і може бути знищений, навіть якщо клієнт використовує його, як і раніше. Домен обробляє цю потенційну проблему, призначивши оренди кожного віддаленого об'єкта, який може бути використаний, щоб зберегти його активним. Сервер активізованих об'єктів може бути реалізований як з одного виклику віддаленого об'єкта, так і один віддалений об'єкт створюється для кожного запиту з використанням параметризованого мережного інтерфейсу. У цьому випадку якнайкраще підходить для спільного використання одного ресурсу або спільних операцій між декількома користувачами. Одиначний режим використовує виклик, коли клієнти повинні виконати за відносно короткий час декілька операцій на сервері, який не вимагає збереження інформації про стан від одного виклику до іншого. Такий підхід є найбільш масштабованим рішенням і має додаткову перевагу, працює добре в середовищі, яке використовує балансування навантаження на прямі виклики на кілька серверів. Єдиний віддалений об'єкт використовується для обробки всіх викликів в основі вигляду параметризованого мережного інтерфейсу (рис. 4).

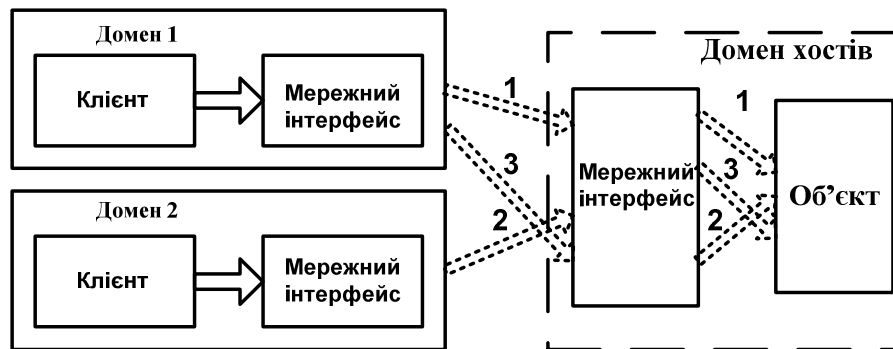


Рис. 4. Спрощена схема вигляду сервера, активованого єдиним об'єктом:
один об'єкт обробляє всі запити

Сервер створює віддалений об'єкт, коли перший клієнт намагається отримати до нього доступ, а не коли він намагається створити його. Тому що віддалений об'єкт створюється тільки один раз зусиллями інших клієнтів; замість цього вони всі дали посилання на той самий об'єкт через параметризований мережний інтерфейс. Кожного разу, коли клієнт викликає віддалений об'єкт, виділяється один новий потік з пулу потоків. З цієї причини за це відповідає розробник. Це також обмежує масштабованість, тому що, як правило, лише кінцеве число доступних потоків.

У режимі активації запитів сервер створює новий віддалений об'єкт кожного разу, коли зроблено запит на нього. Після запиту він буде оброблений, віддалений об'єкт буде вимкнено. Кілька запитів обробляють, коли перший запит завершено, і віддалений об'єкт, створений для нього, буде видалений на момент, коли клієнт, який зробив цей запит, посилається на нульове значення (рис. 5).

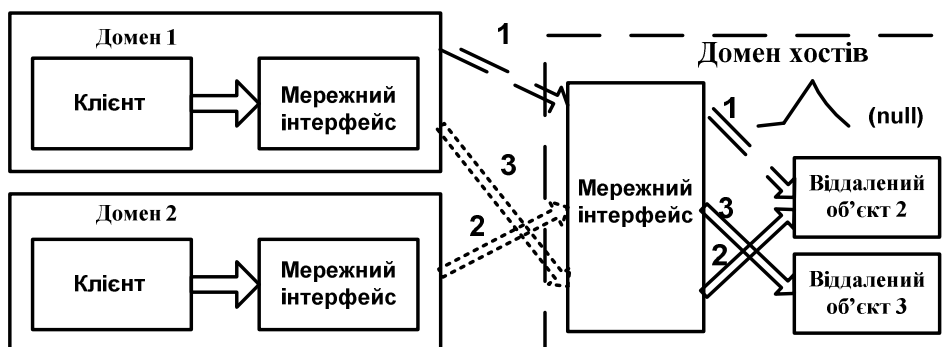


Рис. 5. Спрощена схема вигляду сервера, активованого одним запитом:
один об'єкт створюється для кожного запиту

Другий виклик виходить від іншого клієнта і в результаті створення другого об'єкта сервера. Нарешті, перший клієнт робить свій другий запит і тоді буде створено третій віддалений об'єкт. Перевага активізації запиту в тому, що ресурси є доступні, як тільки запит завершений [6]. Це відрізняється від клієнт-активізованих запитів, коли клієнт проводить на ресурси, поки він не закінчив використання віддаленого об'єкта через параметризований мережний інтерфейс. Недолік одного запиту в тому, що за своєю природою не підтримують інформацію про стан між запитами на віддалений об'єкт. Якщо ви хочете скористатися одним масштабованим запитом, але вимагається, щоб інформація про попередні запити збереглася, можна створити сервер для підтримки своєї інформації про стан у файл або базу даних.

Висновки

Було реалізовано клієнт-серверну взаємодію з використанням параметризованого мережного інтерфейсу. В основу цієї реалізації було покладено параметризацію і наслідування шаблонів за рахунок класів. Також було проаналізовано переваги використання шаблонів: вони дають змогу поєднувати досвід експертів і зробити його доступним розробникам; імена шаблонів складають словник, котрий допомагає розробникам краще розуміти один одного; якщо в документації системи вказано, які шаблони в ній використано, то це дає змогу читачу швидше зрозуміти систему; шаблони спрощують реструктуризацію системи незалежно від того, чи використовувались паттерни під час його проектування.

1. Гамм Э., Хелм Р., Джонсон Р., Влссидес Д. *Приёмы объектно-ориентированного проектирования. Паттерны проектирования.* – М., 2007. – 366 с. 2. Влссидес Дж. *Применение шаблонов проектирования.* – М., 2003. – 130 с. 3. Беркович В., Чудин А. *Практика применения паттернов проектирования.* – 2006 г. [Электронный ресурс]. – Режим доступа <http://rsdn.ru/>. 4. ANSI/NISO Z39.50-1995. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification.* Z39.50 Maintenance Agency Official Text for Z39.50-1995, July 1995. 5. Dave Raggett, Arnaud Le Hors, Ian Jacobs, editors. *HTML 4.01 Specification.* World Wide Web Consortium, 1999. 6. Федоров А. *Платформы и средства создания Web-сервисов // Компьютер-Пресс.* – 2002. – № 6. – С. 38–53. 6. Эммерих В. *Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI: Пер. с англ.* – М.: Мир, 2002. – 510 с.

АНАЛОГОВА СТРУКТУРНО-ФУНКЦІОНАЛЬНА НЕЙРОННА СХЕМА ВИЗНАЧЕННЯ МАКСИМАЛЬНИХ СИГНАЛІВ

© Тимошук П., 2012

Наведено модель неперервного часу аналогової *K*-winners-take-all (KWTA)-нейронної схеми, яка дає змогу визначати *K* найбільших серед *N* невідомих вхідних даних, які можна розрізнити, де $1 \leq K < N$. Модель описується одним рівнянням стану з розривною правою частиною і вихідним рівнянням. Подано відповідну структурно-функціональну схему у вигляді *N* жорсткообмежувальних нейронів прямого поширення і одного жорсткообмежувального нейрона зворотного зв'язку, який використовується для визначення динамічного зсуву вхідних дій. Модель поєднує у собі такі властивості, як висока точність і швидкодія, низькі обчислювальна складність і складність апаратної реалізації і незалежність від початкових умов. Наведено приклади моделювання, які демонструють функціонування моделі.

Ключові слова: модель неперервного часу, рівняння стану, структурно-функціональна схема, жорсткообмежувальний нейрон, аналогова *K*-winners-take-all-нейронна схема.

A continuous-time model of analogue *K*-winners-take-all (KWTA) neural circuit which is capable to extraction the *K* largest from any finite value *N* unknown distinct inputs, where $1 \leq K < N$, is presented. The model is described by one state equation with discontinuous right-hand side and output equation. A corresponding functional block diagram of the model is given as *N* feedforward and one feedback hardlimiting neurons, which is used to determine the dynamic shift of inputs. The model combines such properties as high accuracy and convergence speed, low computational and hardware implementation complexity, and independency on initial conditions. Simulation examples demonstrating the model performance are provided.

Keywords: continuous-time model, state equation, functional block-diagram, hardlimiting neuron, analogue *K*-winners-take-all neural circuit

1. Вступ

Відомо, що *K*-winners-take all (KWTA)-нейронні мережі реалізують вибір *K* найбільших серед *N* вхідних дій, де $1 \leq K < N$. Коли *K* дорівнює одиниці, KWTA-мережа є winner-takes all (WTA)-мережею, що знаходить максимум з множини *N* вхідних даних [1–3].

KWTA-нейронні мережі мають численні застосування, зокрема, вибір *K* найбільших елементів з більшої їх кількості є фундаментальною операцією під час обробки даних і сигналів, прийняття рішень, розпізнавання образів, конкуруючого навчання і сортування [4–6]. KWTA-мережі використовуються у телекомунікаціях [7] і системах технічного бачення [8], для розв'язання задач класифікації [9], фільтрування [10], декодування [11], обробки зображень [12], кластеризації [13]. KWTA-режим застосовується у машинному навчанні, навігації мобільних роботів, видобуванні ознак [14, 15]. KWTA-мережі можуть застосовуватись як базові операційні вузли під час комп'ютеризованої медичної діагностики, для перегляду і отримання інформації, під час видобування й аналізу даних, для фінансового прогнозу, як навчальні засоби, для аналізу оглядів і анкет у різних сферах та ін. [16].

Аналогові KWTA-нейронні мережі (або KWTA-нейронні мережі неперервного часу) порівняно з аналогами дискретного часу забезпечують стабільне функціонування у ширшому діапазоні

зміни параметрів і ширших межах зміни швидкодії мереж [17]. Для розв'язання КВТА-задачі було запропоновано багато різних аналогових нейронних мереж [2–5, 18–24]. У цій статті узагальнюється модель аналогової КВТА-нейронної схеми, запропонованої в [22]. Як і попередник, представлена модель містить одну змінну стану і використовує розривні жорсткообмежувальні активаційні функції. Різницеву функцію моделі отримують в результаті спрощення різницевої функції попередньої моделі. Результати комп'ютерного моделювання свідчать про те, що швидкість збіжності траєкторій змінної стану моделі до КВТА-режиму є близькою до такої швидкості КВТА-нейронних мереж типу Хопфілда, тоді як обчислювальна складність і складність апаратної реалізації моделі є меншою, ніж у цих мереж. Складність апаратної реалізації моделі є близькою до такої складності однієї з найпростіших КВТА-моделей неперервного часу, тоді як швидкість збіжності до КВТА-режиму моделі є меншою, ніж у цього аналога.

2. Модель аналогової нейронної схеми ідентифікації найбільших сигналів

2.1. Математична модель схеми

Нехай задано вхідний вектор $\mathbf{a} = (a_{n_1}, a_{n_2}, \dots, a_{n_N})^T$, $1 < N < \infty$ з невідомими елементами, що мають скінченні значення. Припускається, що вхідні дії локалізуються у відомому діапазоні $[a_{\min}, a_{\max}]$, де елементи a_{\min} і a_{\max} представляють мінімальне і максимальне можливі значення вхідних дій відповідно, $a_{\max} - a_{\min} = A$. Прийmemo, що вхідні дії є такими, що їх можна розрізнити, і впорядкованими у спадному порядку за величиною, задовольняючи нерівності

$$\infty > a_{n_1} > a_{n_2} > \dots > a_{n_N} > -\infty, \quad (1)$$

де n_1, n_2, \dots, n_N – номери першого, другого і так далі до N-го найбільшого елемента вхідного вектора включно. Спроекуємо модель аналогової нейронної схеми, що ідентифікує K найбільших серед цих елементів, які називаються переможцями, де $1 \leq K < N$ – позитивне ціле.

Покладемо, що проєктована модель повинна обробляти вхідний вектор \mathbf{a} так, щоб після скінченного часу збіжності отримати вихідний вектор $\mathbf{b} = (b_{n_1}, b_{n_2}, \dots, b_{n_N})^T$, який задовольняє нерівності

$$b_{n_i} > 0, i \in 1, 2, \dots, K; b_{n_j} < 0, j \in K+1, K+2, \dots, N. \quad (2)$$

Нерівності (2) визначають КВТА-властивість, тобто той факт, що оскільки вихідні дані від b_{n_i} до b_{n_K} є позитивними компонентами \mathbf{b} , то вхідні дані від a_{n_1} до a_{n_K} є K найбільшими компонентами вектора \mathbf{a} . У цій статті представлено модель, яка визначає K найбільших елементів вхідного вектора \mathbf{a} , яка називається моделлю неперервного часу аналогової КВТА-нейронної схеми.

Попередня обробка вхідного вектора \mathbf{a} за допомогою віднімання від усіх його компонентів значення a_{\min} дає такі елементи:

$$\infty > c_{n_1} > c_{n_2} > \dots > c_{n_N} > 0, \quad (3)$$

де $c_{n_k} = a_{n_k} - a_{\min}$, $k = 1, 2, \dots, N$. Подамо вихідні дані проєктованої моделі у вигляді:

$$\begin{aligned} b_{n_i} &= c_{n_i} - x > 0, i \in 1, 2, \dots, K; \\ b_{n_j} &= c_{n_j} - x < 0, j \in K+1, K+2, \dots, N, \end{aligned} \quad (4)$$

де x – скалярний динамічний зсув вхідних даних [23].

Для проєктування моделі аналогової КВТА-нейронної схеми сконструюємо процедуру знаходження значення x , яке задовольняє умову (4). Для цього припустимо, що існує певний момент часу t^* , коли змінна x у встановленому режимі набуває значення $x = x^*$, що задовольняє (4). Для зупинки обчислювального процесу у момент t^* сформулюємо умову, яка контролюватиме кількість позитивних вихідних даних у кожний момент часу протягом обчислювального процесу. З цією метою спростимо різницеву функцію, наведену в [22], до такої форми:

$$E(x) = K - \sum_{k=1}^N S_k(x), \quad (5)$$

де

$$S_k(x) = \begin{cases} 1, & \text{if } c_{n_k} - x > 0; \\ 0, & \text{if } c_{n_k} - x \leq 0 \end{cases} \quad (6)$$

– ступінчата функція, а сума $\sum_{k=1}^N S_k(x)$ визначає число позитивних вихідних даних. Як можна побачити, $E(x)=0$, якщо кількість позитивних вихідних даних дорівнює K . Тому рівність $E(x)=0$ можна використати для ідентифікації необхідної кількості найбільших вхідних даних K .

Візьмемо до уваги, що $x \in [0, A]$. Спроекуємо траєкторію неперервного часу $x(t)$, яка повинна проходити через увесь поміжок $[0, A]$ і досягати значення $x(t^*)$, задовольняючи рівність $E(x)=0$ з будь-якого початкового значення $x_0 \in [0, A]$. Припустимо, що траєкторія $x(t)$ є розв'язком відповідного диференційного рівняння. Нехай $x(t)$ змінюватиметься за експоненціальним законом. Використаємо для цього рівняння стану, представлене в [22], узагальнивши його до такого вигляду:

$$\dot{x} = -\alpha \begin{cases} x, & \text{if } E(x) > 0; \\ 0, & \text{if } E(x) = 0; \\ x - A, & \text{if } E(x) < 0, \end{cases} \quad (7)$$

де α – постійний параметр (або коефіцієнт загасання), який можна використати для контролю швидкості збіжності траєкторій змінної станів до КВТА-режиму. Тоді математичну модель аналогової КВТА-нейронної схеми можна подати у вигляді рівняння стану (7) і вихідного рівняння

$$b_{n_k} = c_{n_k} - x, \quad k = 1, 2, \dots, N. \quad (8)$$

Якщо $E(x) > 0$, то, згідно з (5), $\sum_{k=1}^N S_k(x) < K$ і значення динамічного зсуву $x(t)$ повинно бути зменшене. І навпаки, якщо $E(x) < 0$, то величина $x(t)$ має бути збільшена. У встановленому режимі, коли $\sum_{k=1}^N S_k(x) = K$, $x(t)$ змінюватись не повинно. Для ідентифікації K найбільших вхідних даних рівняння стану (7) повинно формувати відповідний зсув x , що поступово наближається і зрештою потрапляє у діапазон між K -м і $(K+1)$ -м максимальними значеннями вхідних дій, тобто $c_{K+1} \leq x(t^*) < c_K$. Після потраплення зсуву x у проміжок між c_{K+1} і c_K вихідні дані (8) мають точно забезпечувати КВТА-режим.

Зазначимо, що модель (7), (8) можна використати для обробки змінних у часі вхідних дій $a_{n_k}(t)$, $k = 1, 2, \dots, N$, якщо протягом перехідного поцесу модуль зміни швидкості таких дій є значно меншим, ніж у змінної стану x . Інакшими словами, у цьому випадку для кожного $t < t^*$ має задовольнятися умова

$$|da_{n_k}/dt| \ll |dx/dt|, \quad (9)$$

де $k = 1, 2, \dots, N$. Як можна побачити з (7), для того, щоб задовольнялась умова (9), значення параметра α повинно бути вибране достатньо великим.

2.2. Структурно-функціональна схема

Аналогову структурно-функціональну КВТА-нейронну схему, побудовану на основі моделі, що описується рівнянням стану (7) і вихідним рівнянням (8), показано на рис. 1. Схема містить вхідні дані $a_1 \dots a_N$, суматори \sum , інвертуючий інтегратор I з коефіцієнтом підсилення α , зовнішні джерела постійних сигналів K, x_0, A, a_{\min} , блоки S_1, \dots, S_N ступінчатих функцій $S_k(x)$, $k = 1, 2, \dots, N$,

вихідні дані $b_1 \dots b_N$, і функції змінної структури $S_{N+1}(x) = \begin{cases} 0, & \text{if } E(x) \geq 0; \\ x - A, & \text{if } E(x) < 0 \end{cases}$ та

$$S_{N+2}(x) = \begin{cases} x, & \text{if } E(x) > 0; \\ 0, & \text{if } E(x) \leq 0. \end{cases}$$

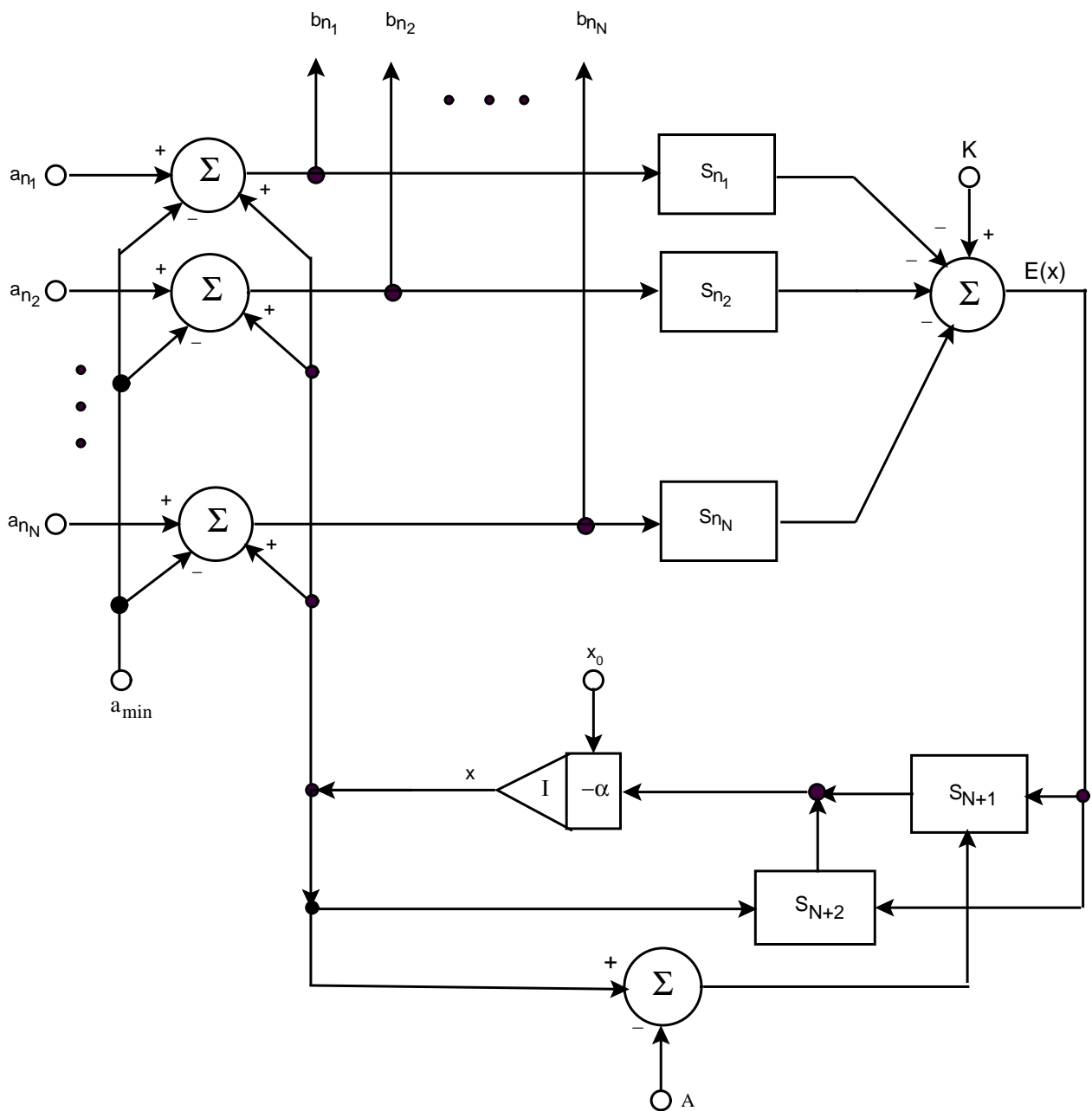


Рис. 1. Архітектура KWTA-схеми, яка описується моделлю (7), (8)

Зазначимо, що виходи блоків S_1, \dots, S_N можуть використовуватись як виходи схеми. Однак, у цьому випадку буде визначатись тільки число переможців K серед N вхідних дій. Інформація про впорядкування вхідних даних за величиною, яка може використовуватись надалі для розв'язання інших задач, отримуватись не буде [20].

Представлена архітектура може бути реалізована у сучасному апаратному забезпеченні, використовуючи такі традиційні компоненти електронних схем, як аналогові суматори, перемикачі, інтегратор і джерела постійних напруги або струму. Як можна побачити, з погляду складності апаратної реалізації схема містить $N+2$ суматорів, $N+2$ перемикачів, один інтегратор і чотири джерела постійних сигналів (або три джерела постійних сигналів, якщо $x_0 = 0$). Для порівняння, одна з KWTA-нейронних мереж неперервного часу типу Хопфілда, запропонована у [2], може бути реалізована в аналоговому апаратному забезпеченні, використовуючи три перемножувачі, $N+4$ суматори, $N+1$ блоків сигмоїдних функцій, N інтеграторів і шість джерел постійних сигналів.

Неважко побачити, що з погляду апаратної реалізації запропонована архітектура є простішою, ніж у цієї мережі. Апаратна реалізація однієї з найпростіших КВТА-нейронних мереж, запропонованої в [21], потребує $N+1$ суматорів, N перемикачів, одного інтегратора і одного джерела постійних сигналів. Отже, складність апаратної реалізації представленої моделі є близькою до цього аналога.

З погляду обчислювальної складності описана модель на кожному обчислювальному циклі потребує послідовного виконання $N+5$ додавань/віднімань, двох логічних операцій, одного підсилення і однієї операції інтегрування. КВТА-мережа з [2] потребує для цього послідовного виконання $N+2$ перемножень, $N+3$ додавань/віднімань, однієї операції сигмоїдної функції і однієї операції інтегрування. Як відомо, перемноження вимагає значно більше машинного часу, ніж додавання/віднімання. Тому очевидно, що складність одного циклу обчислень запропонованої моделі є меншою, ніж у цієї мережі. Модель, представлена в [21], потребує на кожному обчислювальному циклі $N+2$ додавань/віднімань, однієї логічної операції, одного підсилення і однієї операції інтегрування. Отже, складність одного періоду обчислень запропонованої моделі є близькою до обчислювальної складності цього аналога.

Роздільна здатність описаної моделі є теоретично нескінченною і не залежить від значень її параметра. Іншими словами, якщо вхідні дані є такими, що їх можна розрізнити, тоді модель завжди здатна їх ідентифікувати відповідно до КВТА-властивості (2). Оскільки модель спроможна коректно обробляти довільні вхідні дані із скінченними значеннями, які можна розрізнити, її роздільна здатність є такою самою, як в інших нейронних мереж [2, 18, 21].

Оскільки представлена модель може функціонувати коректно за будь-яких початкових значень змінної стану $0 \leq x_0 \leq A$, то відповідна схема для повторної обробки вхідних даних не потребує періодичного скидання, відповідної аналогової керувальної схеми, а також витрачання додаткового часу обробки, що спрощує апаратне забезпечення і підвищує швидкість обробки сигналів [20].

3. Результати комп'ютерного моделювання

Для підтвердження теоретичних результатів, наведених у статті, розглянемо два приклади з відповідним комп'ютерним моделюванням, які демонструють обробку вхідних даних запропонованою моделлю неперервного часу аналогової КВТА-нейронної схеми. Для цього скористаємось відповідними програмними кодами мови MATLAB і персональним комп'ютером 1.81 GHz.

Приклад 1. Задамо для моделі (7), (8) 100 випадкових, однорідно розподілених вхідних даних $a \in [-10000, 10000]$ зростаючої розмірності $N=2, 3, \dots, 100$, випадкові, однорідно розподілені і заокруглені до найближчого цілого величини $K \in \{1, 2, \dots, N-1\}$ і випадкові початкові значення $x_0 \in [-10000, 10000]$ при $A=20000$. Використаємо розв'язувач нежорстких диференціальних рівнянь Адамса–Башфорта–Моултона змінного порядку ODE113, задавши допустимі відносну і абсолютну похибки величиною $1e-15$. Графіки максимального, середнього і мінімального часу збіжності траєкторій змінної стану моделі до КВТА-режиму при $\alpha=10^6$ наведено на рис. 2. Отримані результати свідчать про те, що час збіжності траєкторій змінної стану до КВТА-режиму майже не залежить від кількості вхідних даних N . Різниця між максимальним, середнім і мінімальним значеннями часу збіжності є незначною.

Порівняємо функціонування моделі (7), (8) і однієї з найшвидших моделей аналогових КВТА-нейронних мереж типу Хопфілда, представленої в [24]. Розглянемо перехідні процеси моделі КВТА-мережі (20) і КВТА-моделі (31), (32) з кусково-постійними активаційними функціями, показані на рис. 5–7 і рис. 8–11 відповідно в [24]. Порівняємо ці характеристики з перехідними процесами запропонованої моделі, наведеними на рис. 2. Неважко побачити, що усі

ці моделі демонструють близький час збіжності траекторій змінної стану до КВТА-режиму. Однак моделі (20), (31) і (32), запропоновані в [24], є складнішими і містять обмеження на значення своїх параметрів.

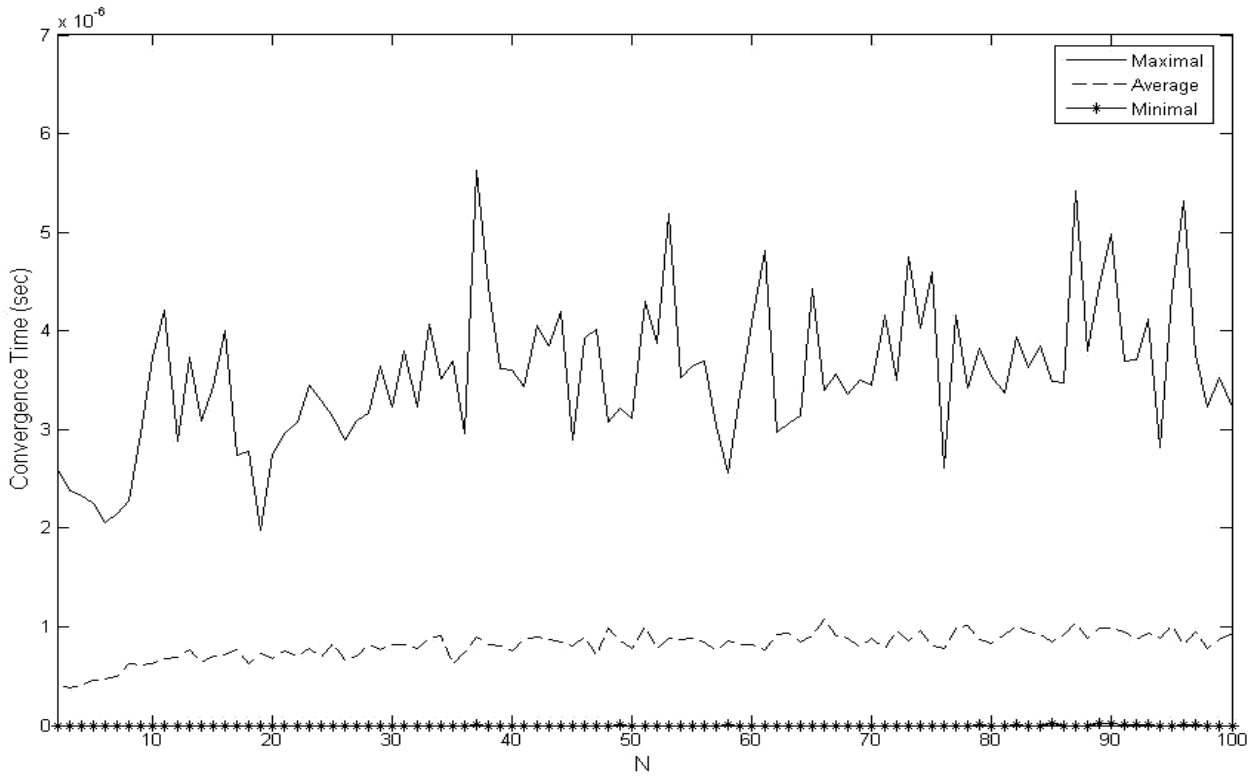


Рис. 2. Графіки максимального, середнього і мінімального часу збіжності траекторій змінної стану до КВТА-режиму моделі (7), (8) при 100 випадкових вхідних даних $a \in [-10000, 10000]$ зростаючої розмірності N , випадковому і заокругленому до найближчого цілого $K \in \{1, 2, \dots, N-1\}$, $\alpha = 10^6$ і випадкових початкових значеннях $x_0 \in [-10000, 10000]$ у приклад 1

Порівняємо ефективність моделі (7), (8) і однієї з найпростіших моделей аналогових КВТА-нейронних мереж, запропонованої в [21], яка містить ступінчасту активаційну функцію Хевісайда. Проаналізуємо перехідні процеси змінної стану, показані на рис. 2, і показані на рис. 6 в [21]. Як можна побачити, час збіжності траекторій змінної стану до КВТА-режиму у моделі (7), (8) є меншим на три порядки.

Приклад 2. Для аналізу ефективності моделі у випадку змінних у часі вхідних дій змодельюємо функціонування моделі у випадку таких чотирьох синусоїдальних сигналів неперервного часу: $a_{n_k}(t) = 10 \sin[2\pi(t + 0.2(k-1))]$ ($k=1, 2, 3, 4$), тобто $N = 4$ і $K=2$. Для зменшення часу обчислень реалізуємо ітераційну модель, використавши замість диференційного рівняння (7) різницеве рівняння з кроком дискретизації $\Delta t = 0.001$. Графіки перехідних процесів для чотирьох вхідних дій, змінної стану і чотирьох вихідних сигналів моделі наведено на рис. 3, де $\alpha = 20$, $A = 10$ і $x_0 = 0$. Результати моделювання свідчать про те, що КВТА-модель дає змогу визначати два найбільші вхідні сигнали серед змінних у часі дій. Зазначимо, що коректного функціонування досягають за величини параметра α на порядок меншої, ніж у випадку моделі, наведеної в [24].

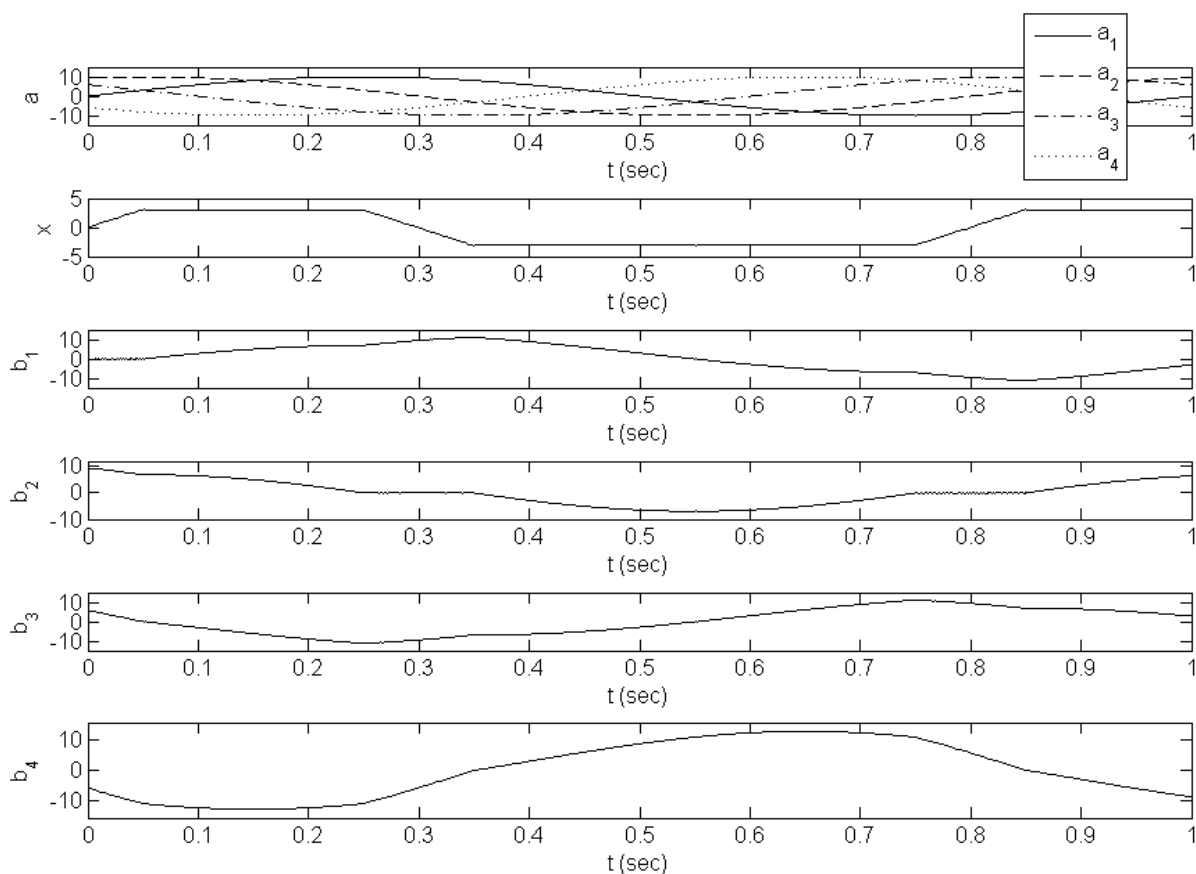


Рис. 3. Графіки входних дій, змінної стану і вихідних сигналів KWTA-моделі (7), (8) при $\alpha = 20$, $A = 10$ і $x_0 = 0$ – приклад 2

Отже, згідно з результатами комп'ютерного моделювання, час збіжності траєкторій змінної стану до KWTA-режиму у запропонованій моделі є близьким до такого часу у моделях аналогових KWTA-мереж типу Хопфілда. Однак, складність апаратної реалізації моделі є меншою, ніж у цих мереж. З іншого боку, швидкість збіжності траєкторій змінної стану у наведеній моделі до KWTA-режиму є вищою, ніж в одній з найпростіших KWTA-моделей неперервного часу. Результати моделювання підтверджують відповідність теоретичним результатам і показують, що запропонована модель спроможна ефективно ідентифікувати не тільки найпостійніший у часі входні дані, але й максимально змінні у часі сигнали.

4. Висновки

Наведено модель неперервного часу і відповідну аналогову структурно-функціональну K-winners-take-all-нейронну схему, спроектовану на основі методу динамічного зміщення входних сигналів. Модель спроможна визначати K максимальних серед N довільних невідомих входних дій зі скінченними значеннями, які можна розрізнити, де $1 \leq K < N$. Швидкість збіжності траєкторій змінної стану моделі до KWTA-режиму може керуватись шляхом зміни величини її єдиного параметра. Комп'ютерне моделювання показує, що швидкість збіжності траєкторій змінної стану моделі до KWTA-режиму є близькою до такої швидкості у KWTA-мереж неперервного часу типу Хопфілда. Складність апаратної реалізації запропонованої моделі є меншою, ніж у цих мереж і близькою до такої складності в одній з найпростіших аналогових KWTA-моделей. Згідно з комп'ютерним моделюванням, швидкість збіжності траєкторій змінної стану до KWTA-режиму у представленій моделі є меншою, ніж у цього аналога.

Функціонування відповідної аналогової нейронної схеми є незалежним від початкових умов. Тому така схема не потребує періодичного скидання, відповідного апаратного забезпечення і

витрачання додаткового часу обробки на цей режим, що є корисним для обробки сигналів реального часу. Отже, запропонована модель поєднує у собі такі властивості, як високі точність і швидкість збіжності траєкторій змінної стану до KWTA-режиму, низькі обчислювальна складність і складність апаратної реалізації, а також незалежність від початкових умов.

1. Lippmann, R. P., Gold, B., Malpass, M. L.: *A comparison of Hamming and Hopfield neural nets for pattern classification*. Technical report TR-769, MIT Lincoln Laboratory (1987).
2. Majani, E., Erlanson, R., Abu-Mostafa, Y.: *On the K-winners-take-all network*. In: Touretzky, D. S. (ed.) *Advances in Neural Information Processing Systems*, vol. 1, pp. 634–642. Morgan Kaufmann Publishers Inc., San Francisco (1989).
3. Tymoshchuk, P., Kaszkurewicz, E.: *A Winner-take-all circuit based on second order Hopfield neural networks as building blocks*. In: *IEEE Int. Joint Conf. on Neural Networks*, vol. 2, pp. 891–896. IEEE Press (2003).
4. Atkins, M.: *Sorting by Hopfield nets*. In: *IEEE Int. Joint Conf. on Neural Networks*, pp. 65–68. IEEE Press (1982).
5. Urahama, K., Nagao, T.: *K-winners-take-all circuit with $O(N)$ complexity*. *IEEE Trans. on Neural Networks* 6, 776–778 (1995).
6. Kwon, T. M., Zervakis M.: *KWTA networks and their applications*. *Multidimensional Syst. and Signal Processing* 6, 333–346 (1995).
7. Binh, L.N., Chong, H.C.: *A neural-network contention controller for packet switching networks*. *IEEE Trans. on Neural Networks* 6, 1402–1410 (1995).
8. Itti, L., Koch, C., Niebur, E.: *A model of saliency-based visual attention for rapid scene analysis*. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20, 1254 – 1259 (1998).
9. Kwon, T.M., Zervakis, M. E.: *A parallel sorting network without comparators: A neural network approach*. In: *IEEE Int. Joint Conf. on Neural Networks*, pp. 701–706. IEEE Press (1992).
10. Cilingiroglu, U., Dake, T. L. E.: *Rank-order filter design with a sampled-analog multiple-winners-take-all core*. *IEEE Journal on Solid-State Circuits* 37, 978–984 (2002).
11. Erlanson, R., Abu-Mostafa, Y.: *Analog neural networks as decoders*. In: Touretzky, D. S. (ed.) *Advances in Neural Information Processing Systems*, vol. 1, pp. 585–588. Morgan Kaufmann Publishers Inc., San Francisco (1991).
12. Fish, A., Akselrod, D., Yadid-Pecht, O.: *High precision image centroid computation via an adaptive k-winner-take-all circuit in conjunction with a dynamic element matching algorithm for star tracking applications*. *Analog Integrated Circuits and Signal Processing* 39, 251–266 (2004).
13. Jain B. J., Wysotzki, F.: *Central clustering of attributed graphs*. *Machine Learning* 56, 169–207 (2004).
14. Liu S., Wang, J.: *A simplified dual neural network for quadratic programming with its KWTA application*. *IEEE Trans. on Neural Networks* 17(6), 1500–1510 (2006).
15. DeSouza, G. N., Zak, A. C.: *Vision for mobile robot navigation: a survey*, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24, 237–267 (2002).
16. Graupe, D.: *Principles of Artificial Neural Networks (2nd Edition)*. World Sci. Publisher, Singapore (2007).
17. Cichocki, A., Unbehauen, R.: *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, Chichester (1993).
18. Wolfe, W. J., Mathis, D., Anderson, C., Rothman, J., Gottler, M., Brady, G., Walker, R., Duane, G., Alaghband G.: *K-winner networks*. *IEEE Trans. on Neural Networks* 2, 310–315 (1991).
19. Ferreira, L. V., Kaszkurewicz, E., Bhaya, A.: *Synthesis of a k-winners-take-all neural network using linear programming with bounded variables*. In: *IEEE Int. Joint Conf. on Neural Networks*, pp. 2360–2365. IEEE Press (2003).
20. Calvert, B. D., Marinov, C. A.: *Another K-winners-take-all analog neural network*. *IEEE Trans. on Neural Networks* 4(11), 829–838 (2000).
21. Wang, J.: *Analysis and design of a k-winners-take-all model with a single state variable and the Heaviside step activation function*. *IEEE Trans. on Neural Networks* 9, 1496–1506 (2010).
22. Tymoshchuk, P. V.: *A dynamic K-winners take all analog neural circuit*. In: *IVth IEEE Int. Conf. "Perspective technologies and methods in MEMS design"*, pp. 13–18. IEEE Press, L'viv (2008).
23. Tymoshchuk, P. V.: *A discrete-time dynamic K-winners-take-all neural circuit*. *Neurocomputing* 72, 3191–3202 (2009).
24. Liu, Q., Wang, J.: *Two k-winners-take-all networks with discontinuous activation functions*. *Neural Networks* 21, 406–413 (2008).

МОДЕЛЬ КОНВЕРТАЦІЇ АБСТРАКТНИХ УНІТЕРМІВ ДО ТИПОВИХ ГРАФІЧНИХ УНІТЕРМІВ-АВТОМАТІВ

© Овсяк В., Козелко М., 2012

Засобами алгебри алгоритмів описано модель побудови системи типових графічних унітермів, призначених для конвертації абстрактного унітерма.

Ключові слова: система, модель, графічний унітерм, абстрактний графічний унітерм, алгоритм.

Means algebra of algorithms described created a model of the system default image unitermiv designed to convert abstract uniterm.

Key words: system, model uniterm graphic, abstract graphic uniterm, algorithm.

Вступ і формулювання задачі

Інструментальні засоби інформаційних технологій і систем реалізуються графічними інтерфейсами. Можливі різні типи інструментальних засобів, наприклад, Windows чи Web проєктів. Сам процес проєктування інструментальних засобів є творчим і складним. Особливо на початкових стадіях проєктування є невідомими як загальний вигляд інструментальних засобів, так і його складові (графічні унітерми [1–3]) – кнопки, елементи альтернативного і безальтернативного вибору, текстові поля та інші. Для спрощення проєктування інструментальних засобів доцільно вводити абстрактні графічні елементи – унітерми.

Використання абстрактних унітермів надає можливість відмовитись від строго типизованих елементів на етапі розроблення інтерфейсу. Такий підхід є універсальнішим, гнучкішим, передбачає повторне використання спроектованих інтерфейсів для різних типів проєктів – таких, як Windows-програма [4, 5] чи інтернет-сторінка [6], а також і в інших проєктах. Однак при компіляції готового проєкту усі абстрактні унітерми мають бути конвертовані до типових графічних унітермів. У зв'язку з цим існує проблема коректного конвертування абстрактних унітермів до типових графічних унітермів із заданням всіх властивостей та методів їх опрацювання, а також дочірних елементів абстрактного унітерма типовому графічному унітерму.

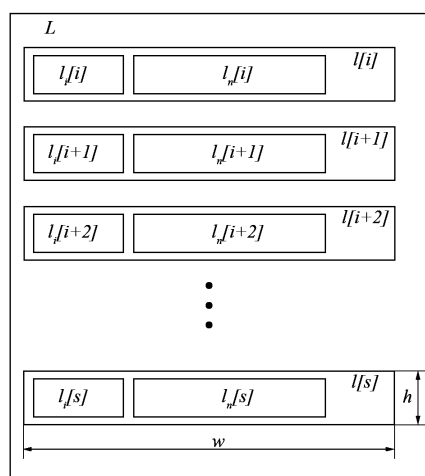


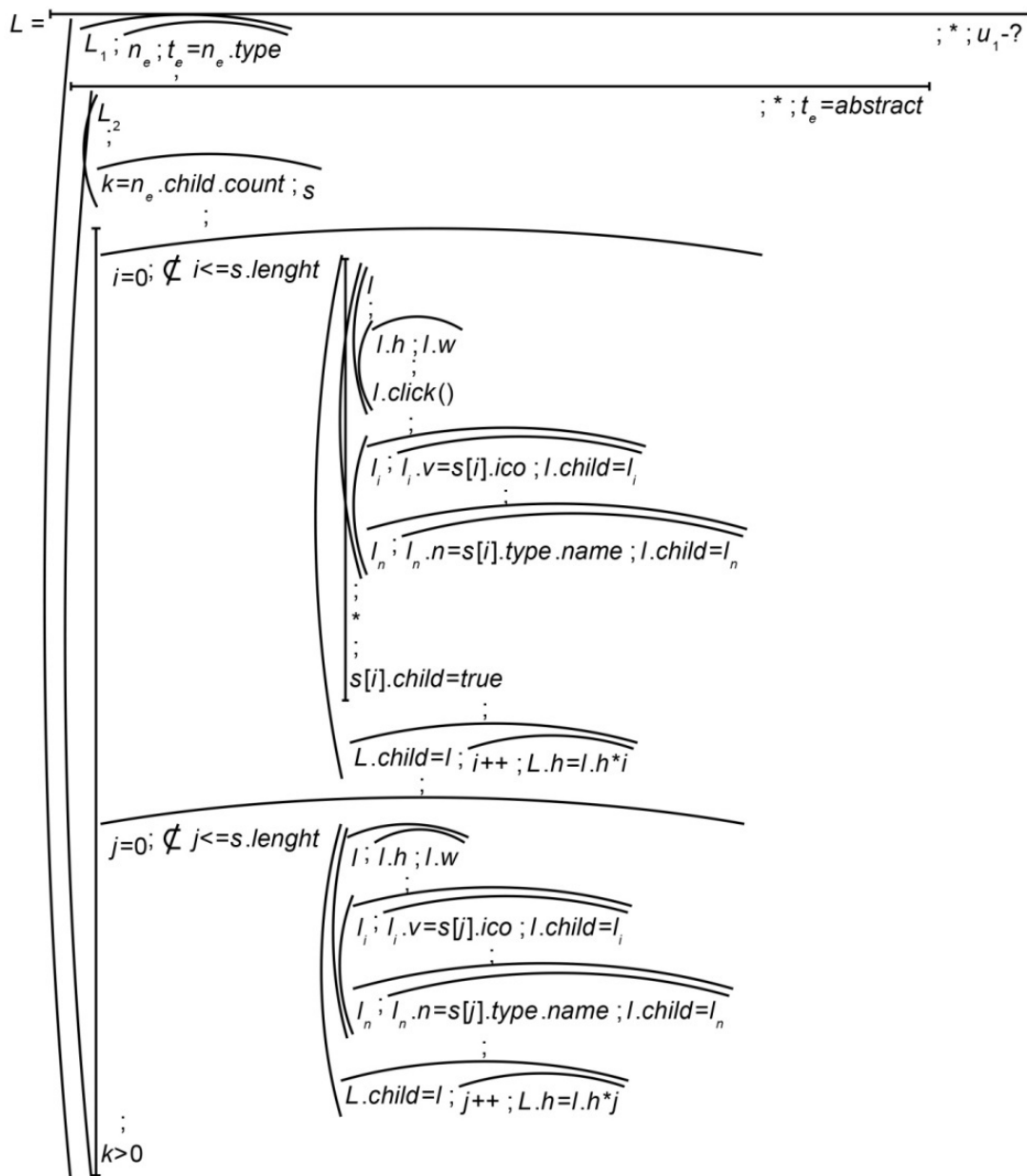
Схема моделі системи типових графічних унітермів, призначених для конвертування абстрактного унітерма

Схематична модель системи типових графічних унітермів

Створену схематичну модель наведено на рисунку. Система типів (L) утворюється із s графічних унітермів контейнерного типу, тобто таких, які можуть містити інші графічні унітерми. Для кожного i -го графічного унітерма задається поле l , яке утворене назвою (l_n) і графічним позначенням (l_i). Для кожного поля задається його висота ($l.h$) і ширина ($l.w$). Кожному полю l задається подія вибору його користувачем для подальшого конвертування абстрактного унітерма до вибраного графічного типу.

Модель побудови системи типових графічних унітермів

Система типових графічних унітермів L формується у разі вибору користувачем унітерма з розміщених на унітермі-формі (виконання умови u_1 , див. наведену нижче формулу алгоритму для побудови якої застосовано модифіковану алгебру алгоритмів [7–10]). Якщо умова u_1 виконується, то за функціональним унітермом L_1 визначається, котрий унітерм вибрав користувач (n_e), а також встановлюється тип вибраного унітерма ($t_e = n_e.type$). Зважаючи на те, що конвертувати можливо лише унітерм абстрактного типу, є доцільним формувати систему L лише при виборі користувачем унітерма абстрактного типу. У зв'язку з цим перевіряють умови $t_e = abstract$ на виявлення абстрактного унітерму. Якщо умова $t_e = abstract$ виконується, то за функціональним унітермом L_2 визначається кількість дочірніх унітермів вибраного абстрактного унітерма ($k = n_e.count$), а також перелік усіх доступних графічних унітермів для конвертації абстрактного. Залежно від кількості дочірніх унітермів k формується система графічних унітермів L . Якщо $k > 0$, тобто обраний абстрактний унітерм містить дочірні унітерми, то формується система L , яка містить графічні унітерми лише контейнерного типу. Для відображення у списку назв типів графічних унітермів, у циклі, поки виконується умова $i \leq s.length$ ($length$ – типова системна властивість [4, 5, 6], яка використовується для обчислення кількості унітермів s системи), генерується поле відображення типу графічного унітерма для конвертації абстрактного. Щоб реалізувати відбір до системи L



графічних унітермів лише контейнерного типу, перевіряють умову $s[i].child=true$, тобто перевіряється чи може графічний унітерм містити дочірні унітерми. Якщо умова $s[i].child=true$ виконується, то генерується поле відображення типу графічного унітерма l із заданими розмірами висоти $l.h$ і ширини $l.w$. Кожному полю l задається подія вибору його користувачем для подальшого конвертування абстрактного унітерма до вибраного графічного типу. Поле відображення типу графічного унітерма l містить унітерм l_n назви типу графічного унітерма і унітерм графічного позначення цього унітерма l_i . Згенерованому унітерму l_i задається графічне позначення відповідного графічного унітерма $l_i.v=s[i].ico$ і розміщення його на полі $l.child=l_i$. Унітерму відображення типу графічного унітерма l_n задається назва типу $l_n.n=s[i].type.name$ і проводиться розміщення його на полі $l.child=l_n$. Успішно згенероване поле l додається до списку $L.child=l$. Враховуючи кількість визначених контейнерних унітермів, формують висоту списку $L.h=l.h*i$. Якщо ж $k=0$, тоді формується система L із всіх можливих типів графічних унітермів. Залежно від кількості доступних для конвертації типів графічних унітермів, у циклі, поки виконується умова $j \leq s.length$, генерується поле відображення типу графічного унітерма l . Полю l задається висота $l.h$ і ширина $l.w$, а також подія вибору поля користувачем $l.click()$. На полі l розміщуються унітерми відображення назви типу графічного унітерма для конвертації $l.child=l_n$ із заданим значенням $l_n.n=l[j].type.name$, а також розміщується графічне позначення відповідного типу $l.child=l_i$ з наступним значенням $l_i.v=s[j].ico$. Успішно згенероване поле відображення типу графічного унітерма l додається до списку $L.child=l$. У цьому випадку висота списку залежатиме від кількості доступних для конвертації обраного абстрактного унітерма типів графічних унітермів $L.h=l.h*j$.

Фрагмент програмної реалізації моделі

Реалізація моделі конвертації абстрактних до типових графічних унітермів автоматів проілюстровано фрагментом програмного коду мови програмування C# у вигляді методу `private void Convert_Click(object sender, RoutedEventArgs e)`. Даний програмний код реалізує конвертацію абстрактного унітерма до унітерма типу кнопки, або типу текстового поля у залежності від вибору користувача. Програмно визначається тип унітерма обраного користувачем для подальшої конвертації абстрактного до типолого графічного унітерма. На основі обраного типу генерується екземпляр елемента цього типу із заданням всіх властивостей притаманних абстрактному унітерму: розміри, розміщення, колір і т.д. Задаються всі функціональні особливоти абстрактного новому згенерованому унітерму. Також відбувається перевірка наявності у абстрактного унітерма дочірних унітермів, для розміщення їх на новому графічному унітермі.

```
private void Convert_Click(object sender, RoutedEventArgs e)
{
    removRes();
    if (contr.GetType().Name == "Abstract")
    {
        FrameworkElement fr = new FrameworkElement();
        SortedList sl = new SortedList();

        Abstract ab = (Abstract)contr;
        if (typStr == "Button")

            //Кнопка
            {

                //Створення об'єкта класу типового графічного
                унітерма
                childbutton = new Button();

                //Передача параметрів абстрактного унітерма
                childbutton.Name = ab.Name;
                childbutton.Content = ab.Name;
            }
    }
}
```

```

        childbutton.Height = ab.Height;
        childbutton.Width = ab.Width;
        Canvas.SetLeft(childbutton, Canvas.GetLeft(ab));
        Canvas.SetTop(childbutton, Canvas.GetTop(ab));

        //Задання назви методу опрацювання події
        childbutton.Click += new
RoutedEventHandler(childbutton_Click);
        oblist.Add(childbutton);
        fr = childbutton;

        //Приписування усіх властивостей
        EventL.Add(fr.Name, sl);
    }
    if (typStr == "TextBox")
        //Текстове поле
    {
        childTextBox = new TextBox();
        childTextBox.Name = ab.Name;
        childTextBox.Height = ab.Height;
        childTextBox.Width = ab.Width;
        Canvas.SetLeft(childTextBox, Canvas.GetLeft(ab));
        Canvas.SetTop(childTextBox, Canvas.GetTop(ab));
        childTextBox.GotFocus += new
RoutedEventHandler(childTextBox_GotFocus);
        oblist.Add(childTextBox);
        fr = childTextBox;
        EventL.Add(fr.Name, sl);
    }
}

//Перевірка і врахування дочірніх унітермів
switch (ab.Parent.GetType().Name)
{
    case "Abstract":
        Abstract a = (Abstract)ab.Parent;
        a.Children.Add(fr);
        a.Children.Remove(ab);
        oblist.Remove(ab);
        break;
    case "Canvas":
        Canvas c = (Canvas)ab.Parent;
        c.Children.Add(fr);
        c.Children.Remove(ab);
        oblist.Remove(ab);
        break;
    case "StackPanel":
        StackPanel s = (StackPanel)ab.Parent;
        s.Children.Add(fr);
        s.Children.Remove(ab);
        oblist.Remove(ab);
        break;
    default:
        break;
} }

```

Висновки

1. Використання створеної моделі забезпечує коректне і безвтратне перетворення абстрактних унітермів до типових графічних унітермів.
2. Нові графічні унітерми забезпечуються усіма функціональними особливостями, притаманними абстрактному унітерму, які задані коритувачем на етапі розроблення графічного інтерфейсу.
3. Модель перетворення абстрактного до типових графічних унітермів програмно реалізовано і апробовано.

1. Овсяк О.В. *Модель інформаційної технології опрацювання формул алгоритмів* / О. Овсяк // *Вісник Нац. ун-ту "Львівська політехніка": Комп'ютерні науки та інформаційні технології*. – 2011. – № 710. – С. 224–234. 2. Овсяк В., Овсяк О., Василюк А. *Елементи мови предметних унітермів* // *Збірник наукових праць „Квалілогія книги”*. – 2009. – № 2(16). – С. 4–9. 3. Овсяк О.В. *Інформаційна технологія побудови моделей інтерфейсів інформаційних технологій і систем* / О.В. Овсяк // *Збірник наукових праць „Комп'ютерні технології друкарства”*. – 2011. – № 26. – С. 105–114. 4. Шилд Г. *C# 4.0: Полное руководство*. – М.: ООО И.Д. “Вильямс”, 2011. – 1056 с. 5. Троэлсен Э. *Язык программирования C# 2010 и платформа .NET 4.0*. – М.: ООО И.Д. “Вильямс”, 2010. – 1392 с. 6. Мак-Дональд М. *WPF: Windows Presentation Foundation в .NET 3.5 с примерами на C# 2008 для профессионалов*. – М.: ООО И.Д. “Вильямс”, 2008. – 928 с. 7. Ovsyak A. *The extended algebra of algorithms width multiconditional elimination* / V. Ovsyak, A. Ovsyak // *Вісник Нац. ун-ту "Львівська політехніка": Комп'ютерні науки та інформаційні технології*. – 2010. – № 672. – С. 291–300. 8. Овсяк О. *Розширена алгебра алгоритмів і модель зв'язку між класичною і розширеною операцією елімінування* / О. Овсяк // *Збірник наукових праць „Комп'ютерні технології друкарства”*. – 2010. – № 23. – С.45–53. 9. Owsiak A. *Rozszerzenie algebry algorytmów* / W. Owsiak, A. Owsiak // *Pomiary, automatyka, kontrola*. – 2010. – № 2. – S. 184–188. 10. Овсяк О.В. *Розширення алгебри алгоритмів аксіомами операцій циклів* / О.В. Овсяк // *Вісник Нац. ун-ту "Львівська політехніка": Комп'ютерні системи проектування. Теорія і практика*. – 2010. – № 685. – С. 12–20. 11. Ovsyak A. *The extended algebra of algorithms with additional cycle elimination axioms* / A. Ovsyak, V. Ovsyak // *Conference "Intelligent Information and Engineering Systems" (INFOS 2011), September 19–23, 2011, Polańczyk, Poland*. – P. 23–34.

ВИКОРИСТАННЯ MVC ДЛЯ СТВОРЕННЯ СОЦІАЛЬНОЇ МЕРЕЖІ ВЕТЕРИНАРНИХ КЛІНІК

© Пиц М., Демида Б., 2012

Досліджується популярна у Веб-програмуванні концепція MVC (Model View Controller) в аспекті застосування її для побудови соціальної мережі ветеринарних клінік. Соціальна мережа ветеринарних клінік характеризується великим масивом та постійним поповненням інформації про нових клієнтів та їхніх домашніх улюбленців, а також різноманітними медичними даними від ветеринарів, такими як: результати аналізів тварини та її історія хвороби. Щоб реалізувати таку систему, недостатньо створити декілька статичних сторінок на PHP, тут варто використати MVC архітектуру, яка дасть змогу оживити мережу і дасть їй можливість рости.

Ключові слова: MVC, веб, соціальна мережа, ветеринарна клініка, домашні улюбленці, PHP, історія хвороби, повідомлення, лікар, клієнт.

This article implements the research of MVC (Model View Controller) concept , which is very popular in WEB - programming and its usage to develop a social network of veterinary clinics. Social network is characterized by a vast, always expanding array of information about new clients, their pets and various medical data, added by attending doctors, like: pet tests and medical history. Implementing few static pages on PHP is not enough to create such a system , an MVC architecture should be used , it will bring this system to life and allow it to grow.

Key words: MVC, web, social network, veterinary clinic, pets, PHP, medical history, message, doctor, client.

Аналіз предметної області

Зазвичай веб-сторінку створюють за такою схемою: базовий HTML каркас, CSS під'єднаний в тезі <head>, а необхідний PHP-код розміщений всередині тега <?php ?>, який, своєю чергою, розташований в потрібних HTML-тегах. Додаємо JavaScript-складові, щоб оживити сторінку. При цьому ми отримуємо традиційну сторінку – основу більшості сайтів. Доцільнішим є інший підхід, за яким можна реалізувати гнучкішу архітектуру. З цією метою ми пропонуємо застосувати концепцію MVC (Model View Controller). Це дасть змогу створити веб-орієнтоване рішення рівня соціальної мережі. Так ми створюємо сервіс, в якому кожен користувач отримує власну, заповнену унікальним контентом сторінку, просто заповнивши відповідну форму під час реєстрації свого профілю.

Постановка задачі та формулювання мети проекту

Головною метою проекту є використання сучасних веб-технологій для створення соціальної мережі [4] ветеринарних клінік. Тепер існує декілька програмних рішень, які допомагають у менеджменті ветеринарних клінік, але більшість із них є програмами для локального користування і не в змозі надати сучасній ветклініці більшості тих можливостей, які можна реалізувати тільки за допомогою веб-програмування і мережі інтернет.

У цій статті розглянуто процес створення зручного для користувачів веб-продукту на основі концепції MVC, який функціонально зорієнтований для управління ветклінікою.

Завданням роботи є створення соціальної мережі, де зможуть зареєструватись як лікарі, що матимуть можливість створити власну онлайн-клініку і набрати туди персонал (інших лікарів), так і клієнти, які матимуть змогу реєструвати свій власний профіль, в якому буде можливість реєстрації своїх домашніх улюбленців та відобразитиметься уся необхідна інформація про них, а саме: результати аналізів тварини, їх історія хвороби тощо. Лікарі та клієнти матимуть змогу спілкуватись між собою за допомогою різноманітних комунікацій чи повідомлень.

Актуальність: запропонована соціальна мережа вирішить такі проблеми:

- Комунікації між клінікою і клієнтами.
- Управління складом клініки.
- Взаємодії працівників клініки (співробітники матимуть змогу спілкуватись між собою будь-де за умови наявності інтернету, зможуть отримати інформацію про тварин на лікуванні у своїх співробітників у будь-який час доби).
- Заочні консультації (лікар зможе консультувати свого клієнта онлайн і за потреби обговорювати з клієнтом час візиту до клініки для огляду тварини).

У зв'язку з цими проблемами визначимо сукупність функцій, більшість з яких буде реалізовано в онлайн-режимі:

Управління складом, зокрема: Продаж медикаментів; Визначення вартості медикаментів, витрачених під час медичних процедур та їхньої кількості; Можливість перегляду наявних і вилучених зі складу медикаментів;

Функція «історія» – в кожній тварини повинна бути своя історія хвороби, яку зможе переглядати лікуючий лікар, а також можливість зберігати дані про аналізи тварин.

Якщо привнести концепцію соціальних мереж до створюваної системи, то слід передбачити і втілення таких функцій:

- Створення системи з 4-х профілів:
 - **лікар**
 - **клієнт**
 - **тварина** (створюється клієнтом зі свого профілю і так прив'язується до нього)
 - **клініка** (створюється лікарем, що вперше створює свою клініку і не є зареєстрований як співробітник іншої клініки), саме в ній і тільки її власник і співробітники зможуть працювати із функціями роботи зі складом та менеджменту клініки.
- Обмін повідомленнями в реальному часі.

Архітектура MVC

Відповідно до стандарту MVC[1, 2, 3, 4], наслідком декомпозиції архітектурних рішень є три складові (рис. 1), а саме :

- **Model** – Модель надає знання: дані та методи роботи з цими даними, реагує на запити, змінюючи свій стан. Не містить інформації, як ці знання можна візуалізувати.
- **View** – Відповідає за відображення інформації (візуалізацію). Часто як View використовують форму (вікно) з графічними елементами.
- **Controller** – Забезпечує зв'язок між користувачем і системою: контролює введення даних користувачем і використовує model і view для реалізації необхідної реакції.

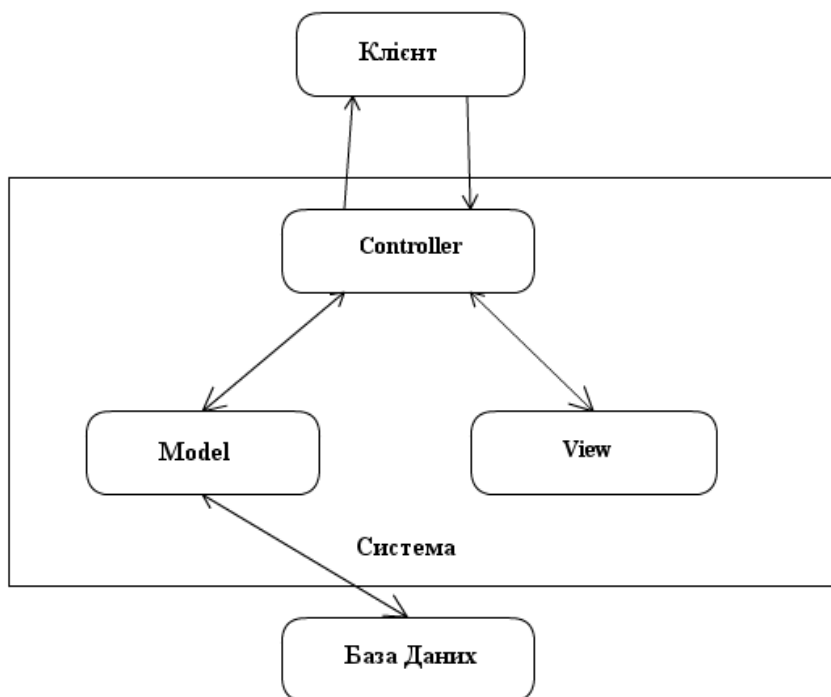


Рис. 1. Діаграма MVC

Вибір мови програмування

У [5] було описано новітні дослідження ефективності мов програмування, в яких брали участь 3 команди висококваліфікованих програмістів, кожна з яких програмувала обраною мовою, а саме : Java , Perl і PHP . Командам дали завдання створити багатофункціональний веб-сервіс. Упродовж доби кожна команда подала готовий проект. При цьому найнадійнішим рішенням виявилась програмна розробка PHP, хоча Java-проект забезпечував найвищу швидкодію . Для реалізації соціальної мережі нами обрано мову PHP.

Реалізація MVC концепції на PHP

Створюючи перший варіант програмних рішень для побудови соціальної мережі, призначеної для застосування у ветеринарії, ми застосували поширену програмну платформу PHP[1–3, 5]. Розглянемо головні аспекти побудови архітектурних рішень власного проекту.

Розпочнімо зі створення єдиної точки входу на домені. Йдеться про те, щоб всі запити до веб-рішення скеровувалися в один файл “index.php”, який буде залежно від URL, вказаного користувачем, визначати, який View і Model використовувати в певній ситуації. Цей файл і буде покладено в основу контролера(**Controller**). Для веб-сервера Apache цього можна досягти так.

Створюємо файл .htaccess в директорії файла index.php (тобто в кореневому каталозі нашого домену), в якому формуємо елемент рушія RewriteEngine із певними параметрами RewriteCond та RewriteRule.

```

RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php

```

Розглянемо, як працює така схема коду. Спершу перевіряємо, чи існує запитований файл, використовуючи директиву RewriteCond, і якщо ні, то перескеровуємо запит на index.php. Така перевірка на наявність файла необхідна, тому що інакше index.php намагатиметься опрацювати всі запити до сайта, зокрема запити на зображення.

Розглянемо спрощену схему дерева каталогів системи (рис. 2).

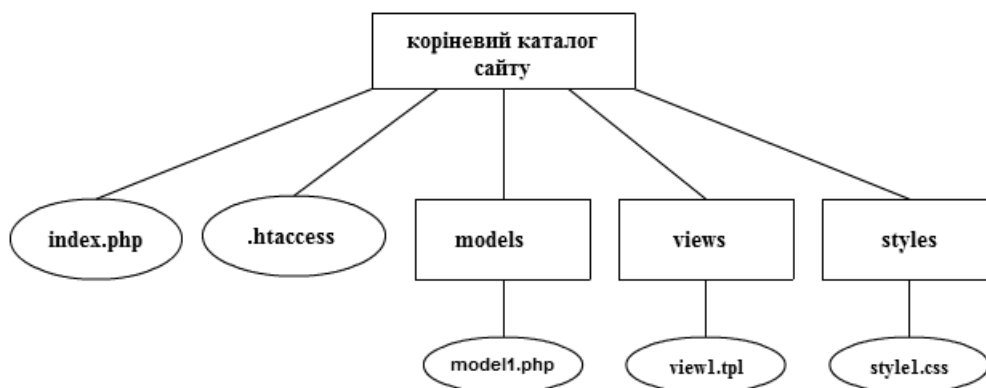


Рис. 2. Дерево каталогів MVC системи

View файли – це файли з розширенням .tpl (принципової різниці між tpl і php файлами немає, це розширення скоріше візуально вказує, що цей файл є темплейтом), в яких міститься код, що відповідає за зовнішній вигляд сторінки – туди підключаються необхідні CSS файли (стилі), а також різноманітні виводи значень певних змінних на сторінку, наприклад, у випадку соціальної мережі, це: виведення імені користувача, даних про нього, його фотографії, різноманітних кнопок, якими він зможе скористатись для запуску певних функцій чи переходу на інші сторінки.

Змінні, використані в View файлах, отримують значення з файла **Model**. Як відомо, модель відповідає за взаємодію з базою даних – у нашому випадку це буде PHP файл, в якому ми опишемо класи, що працюватимуть з базою. Вони міститимуть функції виведення, запису, оновлення та видалення інформації з БД. У цьому ж файлі можна буде визначити права авторизованого користувача залежно від логіна та пароля, які він ввів в процесі входу на сайт – ці права будуть представлені в певній змінній масиву \$_Session.

Далі в файлі Controller (index.php) за допомогою регулярного виразу визначимо, який View файл під'єднати залежно від URL, по якому користувач переходить, і його прав, визначених у Model.

На рис. 3 зображено, як елементи системи взаємодітимуть між собою, а саме:

1. Користувач переходить за посиланням.
2. .htaccess отримує запит і перевіряє, чи існує на сервері файл, який шукає користувач; якщо такого файла немає, запит перескерується в index.php .
3. В index.php під'єднується model файл.
4. За допомогою регулярного виразу model аналізує URL отриманого запиту, виконує відповідні запити до бази даних і присвоює змінним конкретні значення, наприклад, якщо в URI запиту було вказано client/1, то з таблиці клієнтів у БД будуть отримані всі дані клієнта із id "1".
5. В index.php під'єднуємо файл View, який відповідає сегменту client/, згаданому в пункті 4 URI; у цьому випадку це темплейт сторінки клієнта – він використає змінні, визначені в Model файлі для виведення на сторінку інформації про клієнта.

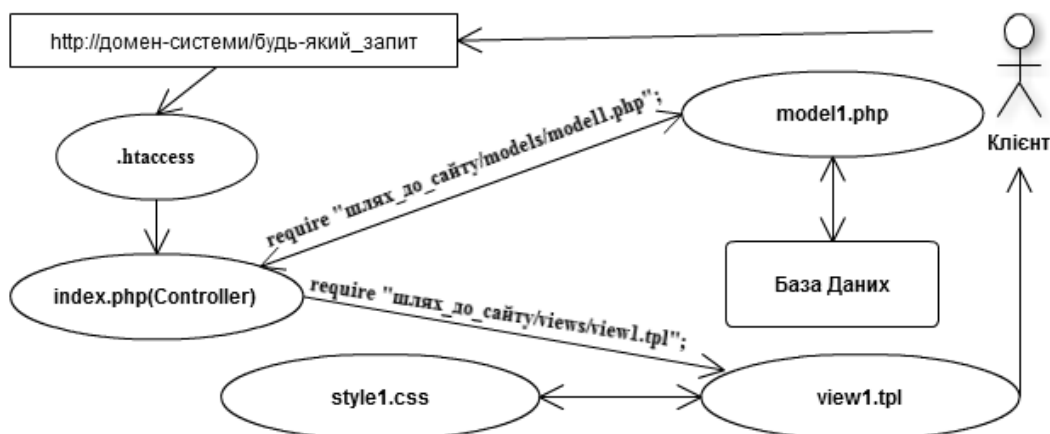


Рис. 3. Діаграма зв'язків системи

На рис. 4 зображено практичне використання описаного вище алгоритму на прикладі входу авторизованого клієнта на власну сторінку у соціальній мережі, припустимо, що її URL :
http://домен_системи/client/1

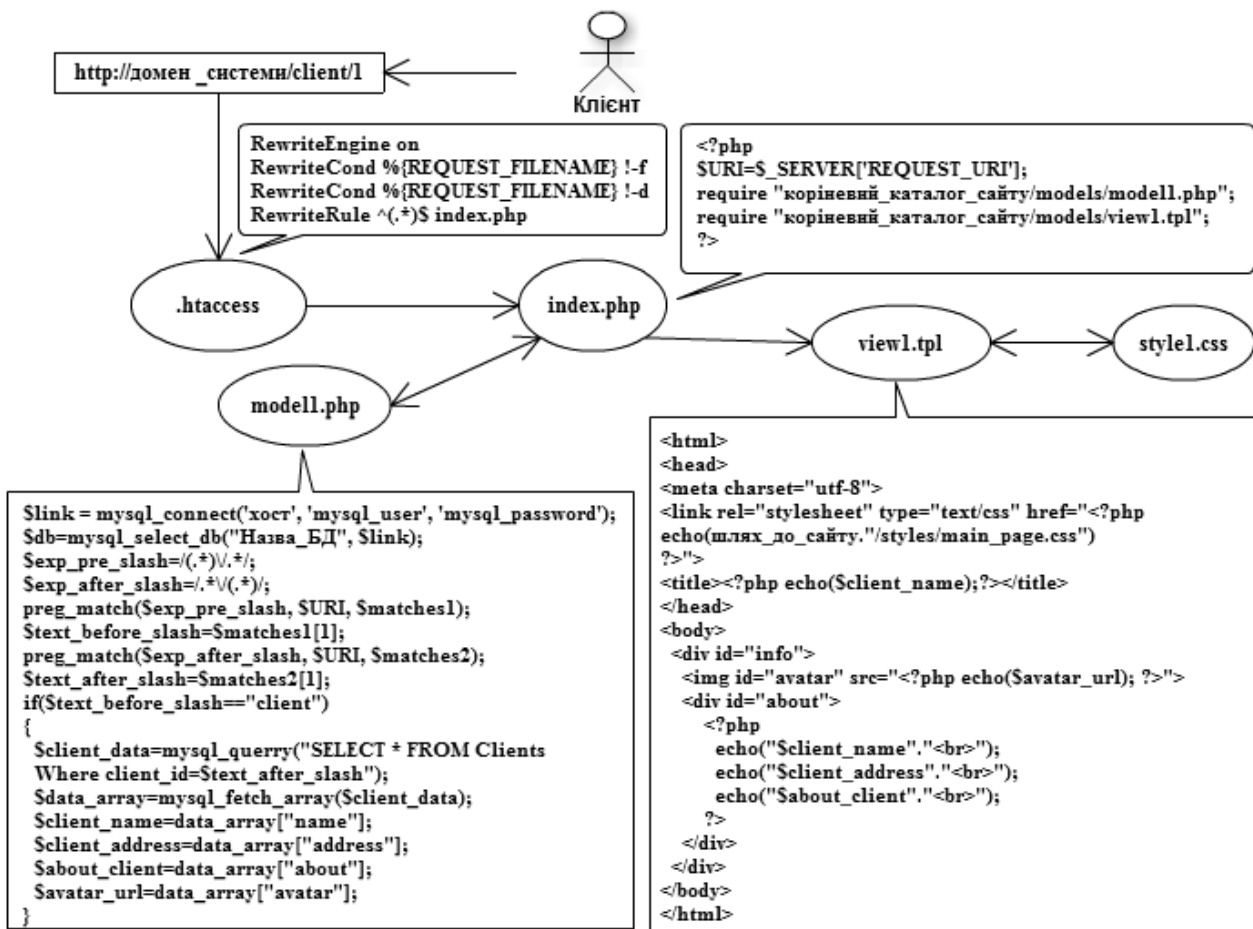


Рис. 4. Приклад реалізації входу клієнта на власну сторінку в соцмережі

Висновки

Соціальна мережа, яку ми пропонуємо для конкретного застосування у ветеринарії, є однією із спроб застосувати новітні веб-технології і досвід провідних розробників [4, 5] для проблеми взаємодії між ветеринарними клініками та їх клієнтами. Це дасть змогу численним ветеринарним клінікам, вагомість яких зростає в сучасних умовах, покращити якість надання допомоги власникам тварин.

Запропоновані рішення сприятимуть покращанню обліку пацієнтів у режимі онлайн, налагодженню співпраці між українськими вет-клініками та забезпечать здійснення управлінських функцій під час ділових стосунків, зокрема на рівні логістичних задач управління складами. Запропонований опис є спрощенням фактичної реалізації соціальної мережі.

1. <http://scripts.by/vvedenie-v-mvc-dlya-php.html> – Основи MVC для PHP. 2. <http://habrahabr.ru/post/31270/> – Витяг із матеріалів спеціалізованого веб-сайта із проблематики MVC для PHP. 3. <http://sitear.ru/material/mvc-php> - Визначення, аналіз переваг MVC в PHP на основі фахових рекомендацій спеціалізованого веб-сайта. 4. Крис Б., Джулиен С. *Агенты влияния в интернете. Как использовать социальные медиа для продвижения бизнеса: пер. с англ.* – СПб.: Питер, 2012. – 304 с. 5. *Идеальная разработка программного обеспечения: Рецепты лучших программистов / Пер. с англ.* – СПб.: Питер, 2012. – 592 с.

ЗАСТОСУВАННЯ СХОВИЩ ТА ПРОСТОРИ ДАНИХ У СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ

© Шаховська Н., Виклюк Я., 2012

Описано методи побудови інтелектуальних систем прийняття рішень. Для опрацювання різнотипних даних обрано простір даних.

Ключові слова: сховище даних, простір даних, системи прийняття рішень, якість даних.

This article is described construction methods of intellectual systems of decision-making. To process data different is selected dataspace.

Key words: datawarehouse, dataspase, decision support systems, data quality.

Вступ

Введемо ряд означень.

Інформаційний ресурс (ІР) – документи і масиви документів в інформаційних системах: бібліотеках, архівах, фондах, банках даних, інших класах інформаційних систем, організовані для багаторазового використання та задоволення потреб користувача.

Структура даних ІР (СДІР) – опис складних інформаційних об’єктів засобами простіших типів даних. Характеризується множиною допустимих значень; множиною допустимих операцій; характером організованості.

Інформаційний продукт (ІП) – документований інформаційний ресурс, підготовлений відповідно до потреб користувачів і поданий у формі товару. Інформаційними продуктами є програмні продукти, текстові файли, веб-сторінки, електронні таблиці, xml-файли, бази даних, сховища даних та ін.

Каталог ІП – метадані про інформаційні продукти. Описує місцезнаходження інформаційного продукту, його структуру даних, методи доступу до інформаційних ресурсів тощо.

1. Постановка задачі

Важливою науковою проблемою, яка виникає у слабкоструктурованому середовищі, є розроблення і удосконалення методів опрацювання різнотипних даних з метою підтримки прийняття рішень. Опрацювання інформаційних ресурсів, що використовують різні моделі даних, схеми керування тощо вимагає розроблення уніфікованого методу доступу до них для того, щоб надати можливість користувачу вибирати адекватний інструментарій для вивчення та використання різних засобів опрацювання даних. Необхідність у цьому виникає в організацій, робота яких полягає в опрацюванні великої кількості різнотипних, взаємозалежних джерел даних, для яких не всі семантичні взаємозв’язки відомі та вказані. У деяких випадках семантичні зв’язки невідомі через невизначену кількість початкових джерел або через брак людей, кваліфікованих у визначенні таких зв’язків. У інших випадках не всі семантичні зв’язки необхідні для класифікації послуг користувачам. Тому в користувачів немає єдиної технології, за якою вони можуть створювати запити відносно цільових задач.

Для прийняття адекватних рішень необхідно, щоб дані, які надходять із різних джерел, задовольняли такі вимоги:

- були повними, несуперечливими та надходили вчасно;
- були інформативними, оскільки повинні застосовуватися для прийняття рішень;
- були однакової структури, щоби можна було завантажити їх у єдине сховище даних та проаналізувати;
- зберігалися в однакових моделях даних та були незалежними від платформи розроблення, щоби можна було використовувати їх в інших засобах.

Однак сьогодні немає жодної методики опрацювання даних, яка б задовольняла всі наведені вимоги до опрацювання даних.

У статті розглянуто відмінності між засобами зберігання та опрацювання даних та визначено їх місце в системах прийняття рішень.

2. Аніліз літературних джерел

2.1. Методи опрацювання даних з різних джерел

Є такі методи опрацювання даних з джерел з різними структурами даних.

1. *Теорія* інтеграції [1] є підмножиною теорії баз даних і формалізує основні поняття за допомогою логіки першого порядку.

Система інтеграції даних формально визначається як трійка $\langle G, S, M \rangle$, де G є глобальною схемою (або схемою посередника), S – множина схем різнорідних джерел даних, M – відображення між запитом джерела і запитом глобальної схеми. G і S подаються виразами мови, алфавіт якої складається зі символів, що встановлюють відображення між символами, притаманних обоим схемам (тут теорія інтеграції переплетена з роботами щодо інтеграції даних через створення метамови). Відображення M складається з тверджень між запитами на G і запитам на S . У випадку, коли користувачі створюють запити за системою інтеграції даних, вони встановлюють зв'язки між елементами в глобальній схемі та схемі джерела.

База даних через схеми визначається як множина наборів – по одному для кожного відношення (у реляційної бази даних). База даних відповідного джерела схеми S складатиметься з множин кортежів для кожного з різнорідних джерел даних. Це єдине джерело бази даних насправді може бути множиною непов'язаних баз даних. База даних, схема якої відповідає віртуальній схемі посередника G , називається глобальною базою даних. Глобальна база даних має задовольняти відображення M стосовно вихідної бази даних. Чинність цього відображення залежить від характеру зв'язку між G і S . Є два способи моделювання цієї відповідності: глобальний (Global As View, GAV), локальний (Local As View, LAV).

У GAV підмножина кортежів, що відображається посередником, є набагато меншою, ніж множина кортежів джерел даних. У LAV кількість кортежів глобальної схеми є набагато більшою, ніж кількість кортежів у джерелах даних. Тому у LAV системах часто доводиться зустрічатися з неповними відповідями.

Опрацювання запитів у системах інтеграції даних зазвичай відображається за допомогою об'єднання. У GAV-системах розробник посередника пише код, щоб перевизначити запис. Кожен елемент в запиті користувача відповідає правилу заміни так само, як кожен елемент глобальної схеми відповідає запиту до джерела. Опрацювання запитів просто розширює підцілі за запитом користувача відповідно до правила, зазначеного у посередника, тобто в результаті запити стають еквівалентні. Найефективнішим з алгоритмів перезапису запитів для GAV є Tsimmis [2].

У LAV-системах процес переписування запитів є радикальніший, оскільки немає посередника, який може встановити відповідність з глобальною схемою. Системі інтеграції необхідно виконати пошук по всьому простору можливих питань для того, щоб знайти ті, які відповідають запиту користувача. У результаті перезапису є ймовірність отримати нееквівалентний запит, але такий, що найбільше відповідає запиту користувача, внаслідок чого і виникає невизначеність у відповіді на запит. Станом на 2009 алгоритм MiniCon є найкращим серед алгоритмів перезапису запитів для LAV [3].

Загалом складність перезапису запитів є NP-повною. Тому для простору даних, що складається зі сотень джерел, алгоритми перезапису можуть використовуватися лише для пошуку даних у визначеній наперед невеликій підмножині джерел.

Іншою назвою «теорії інтеграції» є семантична інтеграція (визначення в ISO 15926). Зазвичай під семантичною інтеграцією розуміють підхід GAV. Для реалізації пропонується використання онтологій та посередника.

2. *Побудова канонічних систем.* У роботах [4, 5] передбачено, що проектування інформаційної системи (ІС) для вирішення завдань над множинними неоднорідними джерелами інформації є композиційним, основна ідея якого полягає в тому, щоб побудувати композицію специфікацій існуючих, релевантних задачі компонентів (інформаційних, програмних, процесних) так, щоб вона уточнювала абстрактну специфікацію.

З метою проектування специфікації компонентів та ІС приводяться до однорідного подання у канонічній інформаційній моделі. Принциповою складовою процесу композиційного проектування є формальне доведення факту уточнення специфікації ІС композицією специфікацій компонентів. Уточнення системи у системі А означає, що користувач може використовувати систему В замість системи А, не помічаючи факту заміни А на В.

У контексті простору даних ідеї Л.С. Калініченка доцільно застосувати для побудови єдиної схеми джерел даних, які використовуються для відповіді на запит користувача. Це означатиме, що користувач, обираючи в побудованій схемі необхідні йому атрибути, звертатиметься до джерела даних як до власної системи.

Для доведення процесу уточнення в Лабораторії композиційних методів проектування інформаційних систем ІПУ РАН (Росія) розроблено:

- формальну семантику канонічної інформаційної моделі (мови СИНТЕЗ) в нотації абстрактна машина (Abstract Machine Notation, AMN). Як ядро канонічної інформаційної моделі, призначеної для уніфікованого подання специфікацій під час композиційного проектування систем, використовується гібридна об'єктна мова. AMN є формальною мовою специфікацій, що ґрунтується на логіці предикатів першого порядку і теорії множин і призначена для побудови математичних моделей ІС;

- інструментальні засоби, які автоматично відображають специфікації канонічної моделі в AMN.

Ці методи успішно використовуються для опрацювання неоднорідних джерел, що створюються за метаописами мови СИНТЕЗ. Проте у реальних предметних областях вже існують джерела, які не відповідають цим метаописам. Тому теорія Л.С. Калініченка потребує уточнення для розроблення канонічних моделей для джерел з невідомими структурами даних. Проте ідея побудови відображення джерел даних у канонічній формі використано в роботі для побудови сховища консолідованих даних як результату інтеграції даних з джерел.

3. *Онтології.* Цей метод є розширенням методу пошуку на основі метаданих [6, 7]. Насамперед будується онтологічна модель предметної області, задаються онтологічні специфікації понять предметної області і зв'язків між ними. Онтологічні специфікації використовуються для пошуку класів і типів інформаційних джерел, релевантних класам і типам посередника. Елемент специфікації джерела вважається онтологічно релевантним елементу специфікації посередника того самого виду (клас, тип, атрибут, функція, параметр), якщо між відповідними для них онтологічними поняттями встановлена позитивна асоціація або асоціація узагальнення/спеціалізації.

Кожний клас («концепт») може бути співвіднесений з іншим подібним концептом завдяки доповненню тегів метаданих, що вказують на властивості, загальні риси, розбіжності тощо. Розширення моделей тегами дає змогу створювати такі структури, яких раніше не могло бути [7]. У семантичній моделі будь-яка інформаційна одиниця подається графом, що спрощує її модернізацію; наприклад, злиття двох моделей зводиться до об'єднання їхніх графів. Інформаційну одиницю може бути подано ідентифікатором Uniform Resource Identifier (URI), за допомогою якого можна встановити зв'язки між двома або більше інформаційними одиницями.

Є декілька проектів систем, що використовують погляд на світ з боку семантики, закладеної в метаданих, і застосовують онтології. До таких проектів відносяться SIMS, HERMES, InfoSleuth, TSIMMIS, Information Manifold [8]. Ці проекти надають доступ до гетерогенних і розподілених інформаційних ресурсів.

У Росії розроблено систему інтеграції даних на основі онтологічного підходу для нафтової галузі [9]. Інтегруються реляційні бази даних за допомогою RDF-RIF-описів. Для інтеграції джерел інших типів (XML) вимагається їх ручне описання в RDF. Перевагою підходу є використання дескриптивної логіки (правила, сформовані в RIF), що дає можливість виводити нові знання. Для порівняння двох баз даних (атрибутів) використовуються міри семантичної близькості. Визначаються на основі нестрогого порядку розміщення в онтології.

Проте метод А.Ф.Тузовського [9] вимагає роботи винятково зі структурованими джерелами і ґрунтується насамперед, на модифікованому підході Калініченка. Окрім того, єдиною мовою формування запитів є SPARQL, що вимагає від користувача деякої підготовки.

Отже, на рівні сховища даних доцільно використовувати традиційні методи інтеграції, а на рівні простору даних – семантичну інтеграцію або розширення традиційної з попереднім визначенням структури даних джерела та методів доступу до даних.

3. Означення сховища та простору даних

Означення 1. Сховище даних – це агрегований інформаційний ресурс, що містить консолідовану інформацію з усієї проблемної області та використовується для підтримки прийняття рішень.

Означення 2. Консолідована інформація – це одержані з декількох джерел та системно інтегровані різнотипні інформаційні ресурси, які в сукупності наділені ознаками повноти, цілісності, несуперечності та становлять адекватну інформаційну модель проблемної області з метою аналізу її опрацювання та ефективного використання в процесах підтримки прийняття рішень.

Означення 3. Простір даних DS – це множина усіх інформаційних продуктів предметної області

$$DS = \langle DB, DW, Wb, Nd, Gr \rangle, \quad (1)$$

де **DB, DW, Wb, Nd, Gr** – інформаційні продукти, що подають множини баз даних, сховищ даних, веб-сторінок, текстових файлів, електронних таблиць, графічних даних відповідно.

Говорячи про інформаційний продукт, матимемо на увазі його вміст (інформаційний ресурс), а також множину відомостей про нього (розміщення, схема доступу, швидкість оновлення інформації тощо). Також нас цікавитимуть операції, які виконуються над IP залежно від його СДІР. Хоча інформаційні продукти, що входять в ПД, мають різні структури даних, методи доступу, проте вони усі *виконують однакову роль*: надають дані для простору даних через фіксацію свого стану та забезпечують виконання притаманних для них операцій, причому ці операції та їх результати є визначені для усього простору даних.

Основним завданням простору даних є надання можливості користувачеві працювати з джерелами даних, не знаючи їхніх СДІР, розміщення, методів доступу тощо.

4. Завдання, що вирішуються за допомогою сховищ даних

Спектр застосування технології сховищ даних достатньо широкий. Задачі, що розв'язуються за допомогою сховищ даних, як правило, належать до класу задач керівного аналізу і стратегічного планування. Нижче наведено приклади типових питань, на які можна відповідати за допомогою сховищ даних: фінансовий аналіз, аналіз продажу, аналіз прибутковості, аналіз каналів продажу, аналіз клієнтської бази, маркетинг, аналіз якості обслуговування клієнтів, аналіз складських запасів, аналіз постачальників, аналіз персоналу.

Прикладом предметної області, для якої доцільно будувати сховище даних, є університет. Цей об'єкт характеризується наявністю ієрархічної структури, великими обсягами інформації, необхідністю розв'язання задач комплексного аналізу. Для інформатизації ВНЗ зазвичай необхідно інтегрувати дані, оскільки на момент розроблення єдиного сховища даних вже є ряд інформаційних продуктів, які повинні обмінюватися між собою інформацією, а також надавати частину інформації у корпоративне сховище даних з метою її подальшого аналітичного опрацювання. До аналітичних

задач, які необхідно розв'язати аналітикам університету, належать: пошук залежностей між отриманими оцінками студентів за предметами та результатами вступу; пошук дисциплін, у яких показники «Успішність», «Якість» або дуже високі, або дуже низькі; пошук залежностей між результатами наукової діяльності студентів та їх практичними здобутками у вигляді проходження практик, участі в олімпіадах, конкурсах робіт тощо.

Існуючі програмні продукти виконують поставлені перед ним задачі і не завжди дають змогу розв'язати нові. Наприклад, для автоматизації та супроводу навчального процесу «Львівської політехніки» розроблено такі інформаційні системи:

- «Абітурієнт» – облік вступників на перші курси бакалаврату та магістратури, формування наказів на зарахування тощо;
- «Навчальні плани» – розроблення та облік навчальних планів за спеціальностями;
- «Деканат» – облік студентів, облік індивідуальних навчальних планів, облік та аналіз успішності студентів;
- «Розклад» – облік аудиторного фонду, формування розкладу занять та екзаменів;
- «Випускник-працевлаштування» – аналіз якості підготовки випускників «Львівської політехніки», облік практик та дипломних робіт студентів.

Перелічені системи зберігаються на одному сервері під керуванням СКБД SQL Server. Як метод інтеграції використовується інтеграція на рівні сховища даних, технологія тиражування – копіювання визначеної частини даних з однієї системи в іншу за певним розкладом. Схему взаємодії основних БД університету зображено на рис. 1.

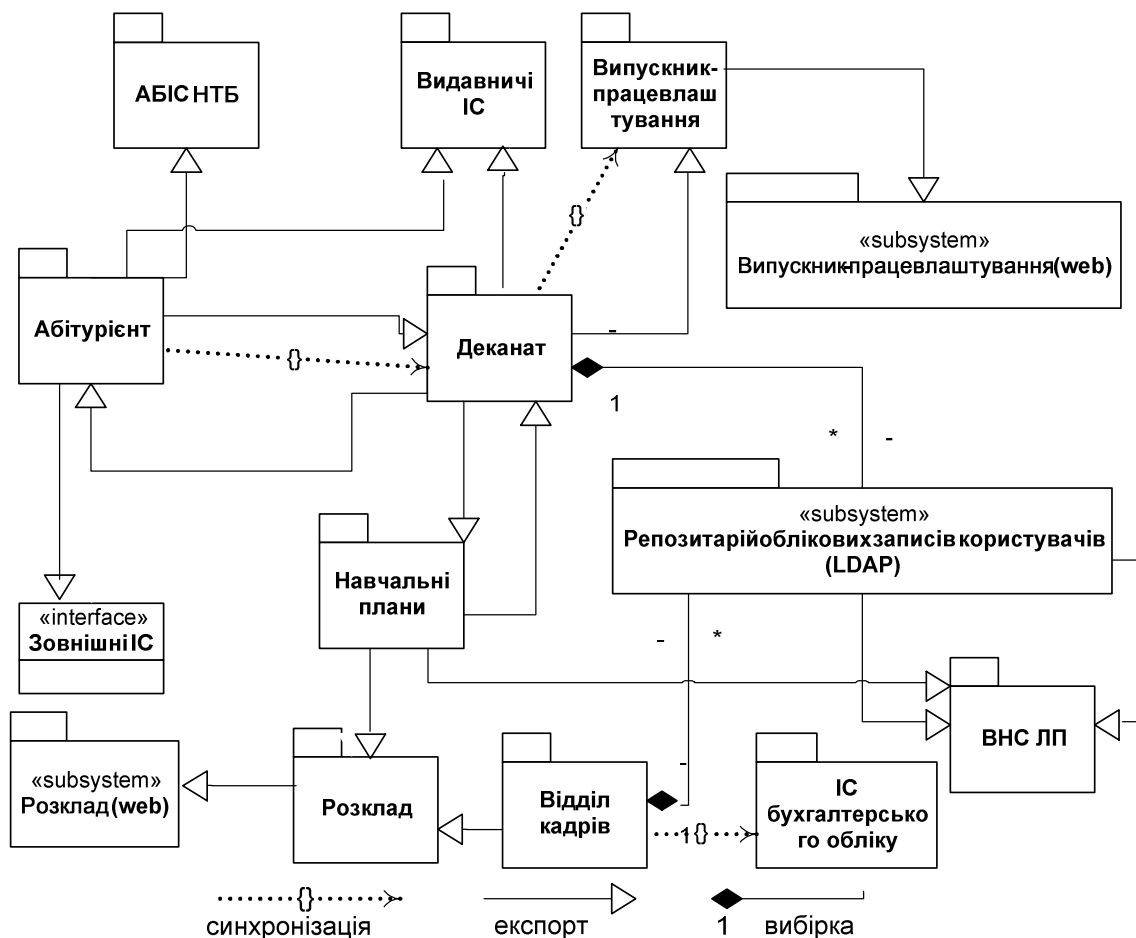


Рис. 1. Схema взаємодії основних БД ВНЗ

Зазвичай для тиражування використовуються три методи обміну даними між інформаційними продуктами (рис. 2):

- синхронізація – порівняння даних ;
- експорт – копіювання даних таблиці за певний період;
- вибірка – копіювання частин таблиці за певний період за параметрами користувача.

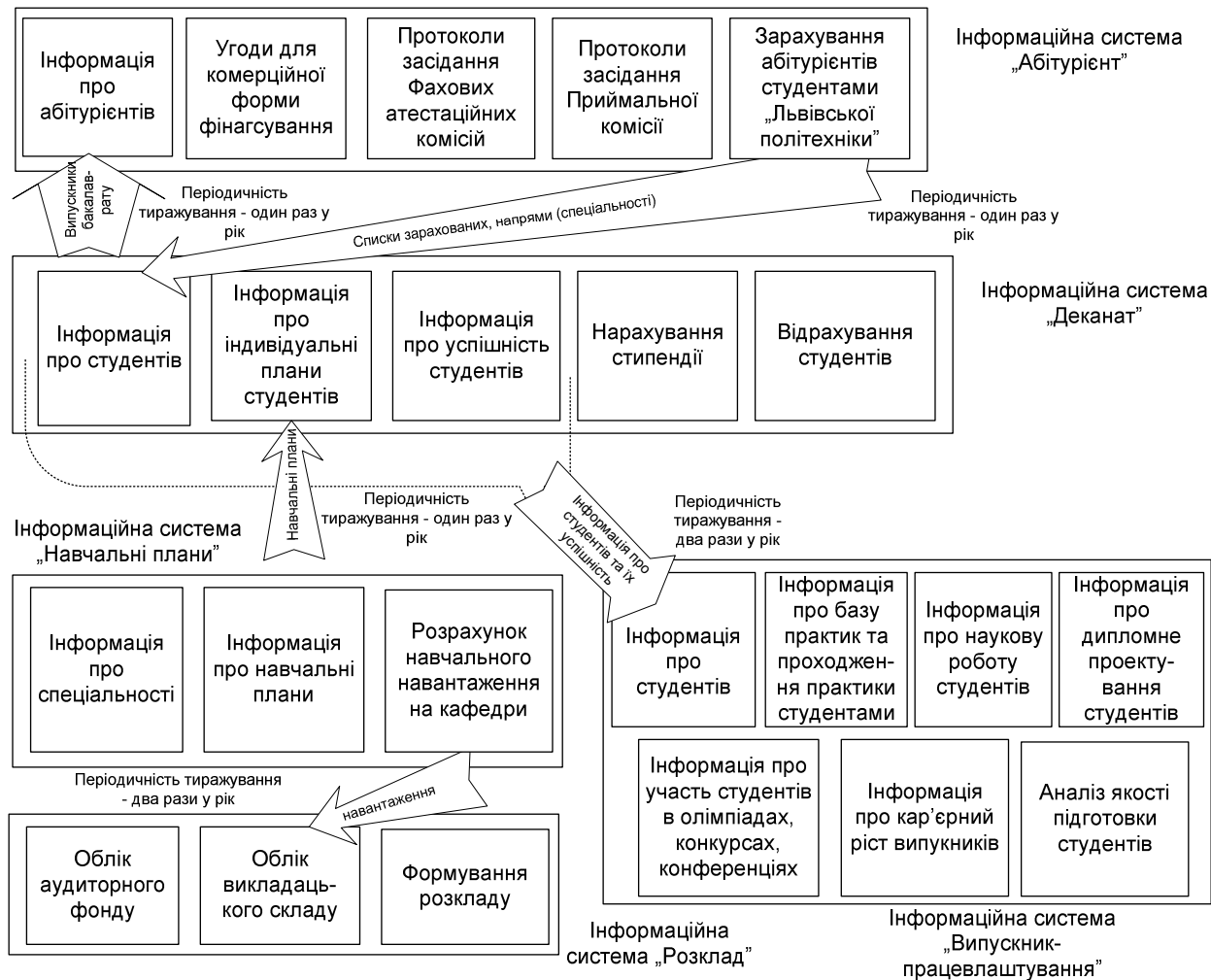


Рис. 2. Тиражування даних у інформаційних системах ВНЗ за допомогою переглядів

Методи обміну даними, показані на рис. 3, реалізуються такими засобами СУД MS SQL Server: DB Snapshot, перегляд, збережена процедура, розподілена транзакція, репліка.

Оскільки, як вже зазначалося, інформаційні продукти університету розроблялися ітераційно та на момент проектування сховища даних вже певний час функціонували, то для розроблення централізованого сховища даних використано підхід Інмона. Згідно з вимогами Інмона, сховище даних університету має виглядати так (рис. 3):

а. вітрини даних інститутів, які містять локальні копії систем «Деканат», «Випусник-працевлаштування» з первинними схемами даних (висока нормалізація) – інститути вносять детальну інформацію про свої потоки даних та використовують вітрини для оперативного аналізу даних;

б. центральне сховище університету, що містить інформацію з систем «Навчальні плани», «Розклад», «Абітурієнт», а також денормалізовану агреговану інформацію вітрин інститутів – інформація про навчальні плани та розклад тиражуються у вітрини даних, а агрегована інформація про успішність студентів консолідується з вітрин у центральне сховище.

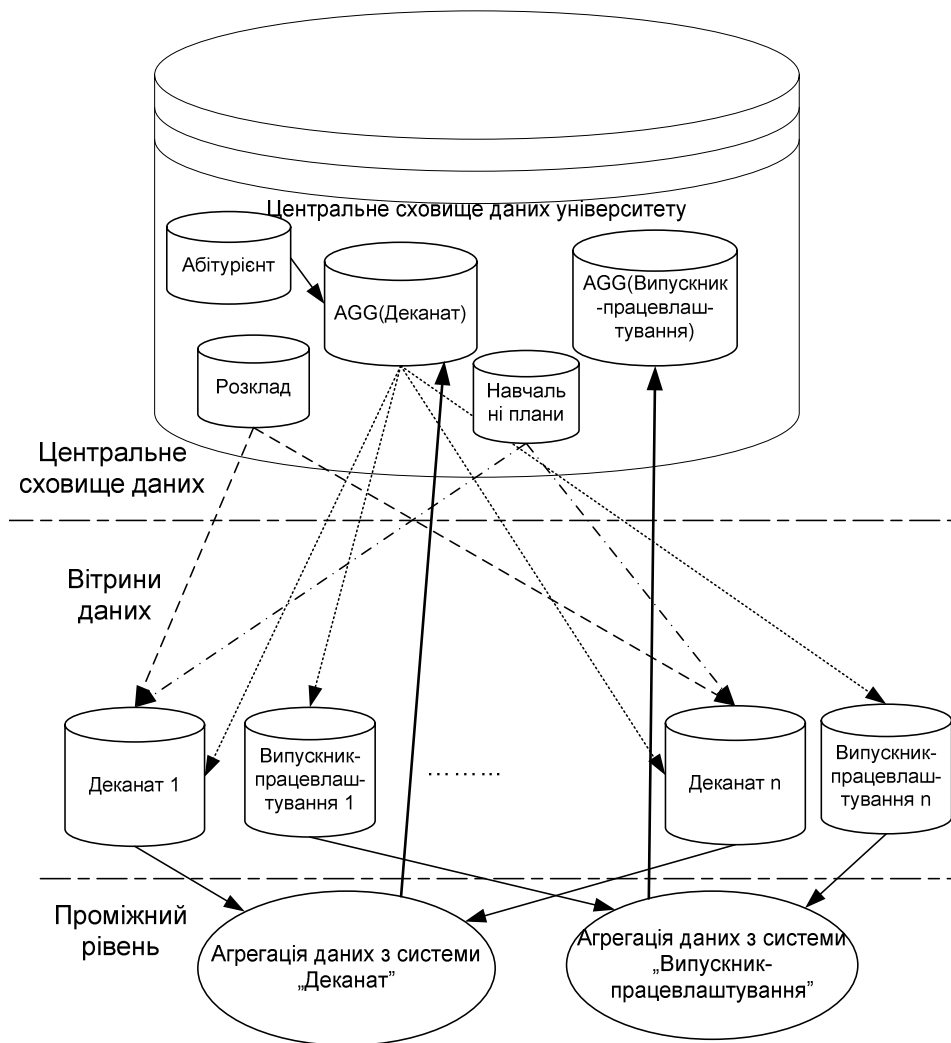


Рис. 3. Схема корпоративної фабрики сховища даних університету

5. Предметні області, для яких доцільно розробляти простори даних

Простори даних доцільно використовувати у різних галузях. Особливостями цих галузей є:

- ієрархія об'єктів;
- використання різних засобів опрацювання та аналізу даних;
- опрацювання даних з джерел, які наперед не були пристосовані для інтеграції;
- необхідність опрацювання поточних даних та даних, що надходять зі запізненням.

Спроекуємо простір даних для керування науковими даними.

Університети і дослідницькі інститути у всьому світі активно планують і реалізують архіви своєї наукової продукції. Крім того, Інтернет подає нові можливості для своєчасного поширення наукової інформації. Правильний вибір програмного забезпечення, яке б максимально повно задовольнило потреби навчальних і наукових організацій, за мінімуму прямих і непрямих витрат – один з аспектів розв'язання цієї задачі.

Простір даних обліку та супроводу наукових досліджень об'єднує інформацію про (рис. 4):

- інженерні та технічні розрахунки експериментів чи досліджень, подані у вигляді таблиць Excel, текстових файлів з розділювачами тощо;
- наукові статті та тези, описані у внутрішніх базах даних кафедр чи наукових підрозділів;
- програми, що використовуються для ведення розрахунків та експериментів;
- наукові звіти кафедр та наукових підрозділів про роботу працівників;
- літературні джерела (бібліотечні фонди), подані у вигляді внутрішніх баз даних чи загальноуніверситетського сховища даних, url-посилання, мультимедійна та графічна інформація, збережена на файл-сервері університету чи подана у глобальній мережі.

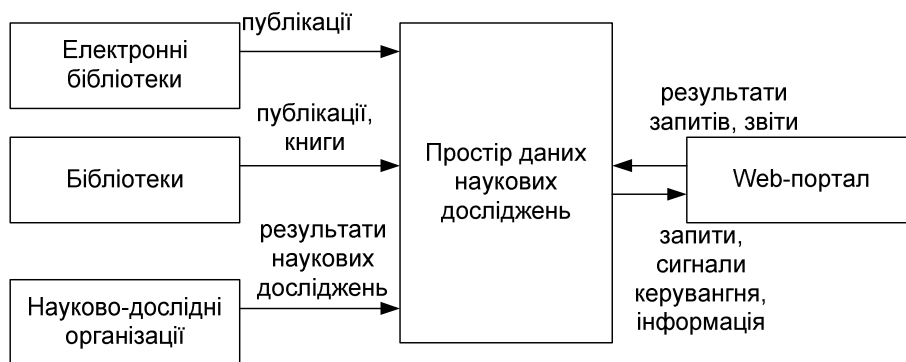


Рис. 4. Схема функціонування простору даних наукових досліджень

Простір даних стосовно аналізу наукових досліджень повинен забезпечувати:

1. Облік документів за результатами наукових досліджень – збереження структурованих даних про документи, авторів і зміст документів. Задача обліку полягає у розборі текстових даних і перетворенні їх на структуровану інформацію – опис документів, а також класифікація документів за галузями наукових досліджень. Також необхідним є накопичення такої інформації в базі знань.

2. Створення онтологій предметних областей – визначення сутностей предметної області і зв'язків між ними. До виконання цієї роботи залучаються експерти в предметних галузях. Результати оформляються у вигляді бази знань на основі онтологій.

3. Визначення змісту наукових досліджень. Розбір змістовного наповнення документа і його прив'язка до певної предметної області. Визначення термінів і понять, використаних в документі, і прив'язка до онтологій предметної області дослідження.

4. Пошук інформації в базі знань. Можливі два варіанти розвитку подій:

- a. пошук документів за запитом користувача;
- b. пошук документів, які пов'язані з документом або схожі на заданий документ.

5. Кількісний аналіз – визначення кількості дослідників (досліджень) за галузями і тематиками. Такий аналіз необхідний для визначення перспективності досліджень в певній галузі чи темі.

6. Інтеграція з іншими системами – використання електронних бібліотек і надання власного програмного інтерфейсу (API) для доступу до системи.

Простір даних наукових досліджень призначений для застосування, насамперед, у галузі науки та освіти, а також в промисловості для пошуку нових технологій і слідкування за їх розвитком з метою майбутнього використання. Впровадження пропонується в університетах та науково-дослідних інститутах, профільних міністерствах, Вищій атестаційній комісії. Також можливе незалежне застосування для інтеграції даних з різних організацій та установ.

Розроблювана програма має містити такі модулі (рис. 5):

- 1) база даних і знань;
- 2) підсистема інтеграції і збирання інформації;
- 3) підсистема аналізу;
- 4) Web-портал для керування системою і відображення результатів роботи системи.

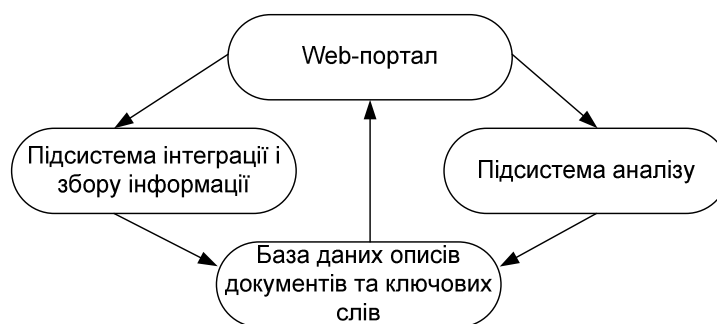


Рис. 5. Діаграма залежностей компонентів

База даних і знань призначена для накопичення структурованих даних і метаданих про наукові публікації. Вона є центральною частиною програми, оскільки надає інформацію іншим модулям. Також в базі даних зберігаються метадані про документи – тобто перетворені в зрозумілий для програми формат з метою подальшого аналізу.

Оскільки система передбачає інтеграцію з іншими системами і джерелами документів про наукові дослідження, вона має мати функціональність, яка б давала змогу зчитувати файли в різних форматах і надавати свої дані для інших систем. За це відповідає Підсистема інтеграції і збору інформації. Через цю підсистему наповнюється база даних. Результатом роботи модуля є структурована інформація про документ. Підсистема аналізу аналізує документи на основі метаданих та реалізує кілька різних стратегій порівняння документів.

Першим етапом реалізації простору даних наукових досліджень є проектування бази даних (БД). Схему БД подано на рис. 6.

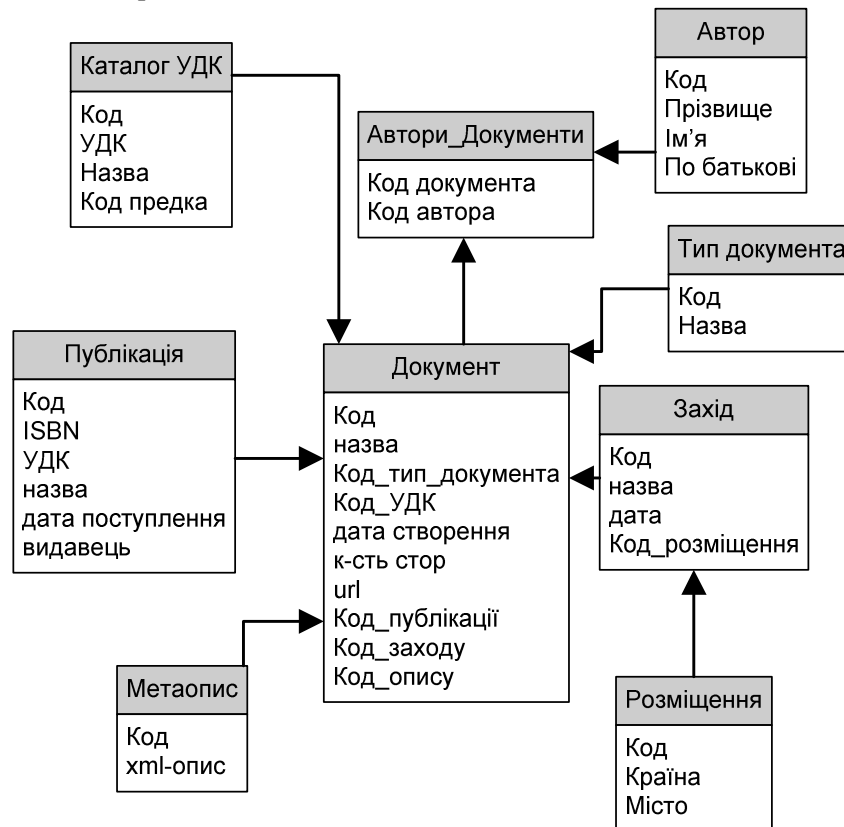


Рис. 6. Схеми БД обліку наукових досліджень

База даних реалізована в СКБД SQL Server 2005, що дає змогу використовувати великий арсенал готових рішень для аналізу даних і текстів.

Додавання документа до системи відбувається через його розбір. Оскільки документи відрізняються як за форматом, так і за змістом, то необхідно передбачити можливість додавання розширення можливостей для розбору.

6. Засоби реалізації сховищ та просторів даних

Для реалізації сховищ та просторів даних використовують системи управління базами даних, засоби обміну даними та інтеграції. Джерела даних, такі як електронні таблиці, мультимедійна інформація тощо, можуть мати свої власні засоби зберігання та опрацювання, і тоді завданням засобів інтеграції є розпізнавання цих інформаційних ресурсів та організація доступу до них. Коли йдеться про сховища даних, то структура джерел є відомою наперед, і основним завданням є очищення та завантаження самих даних.

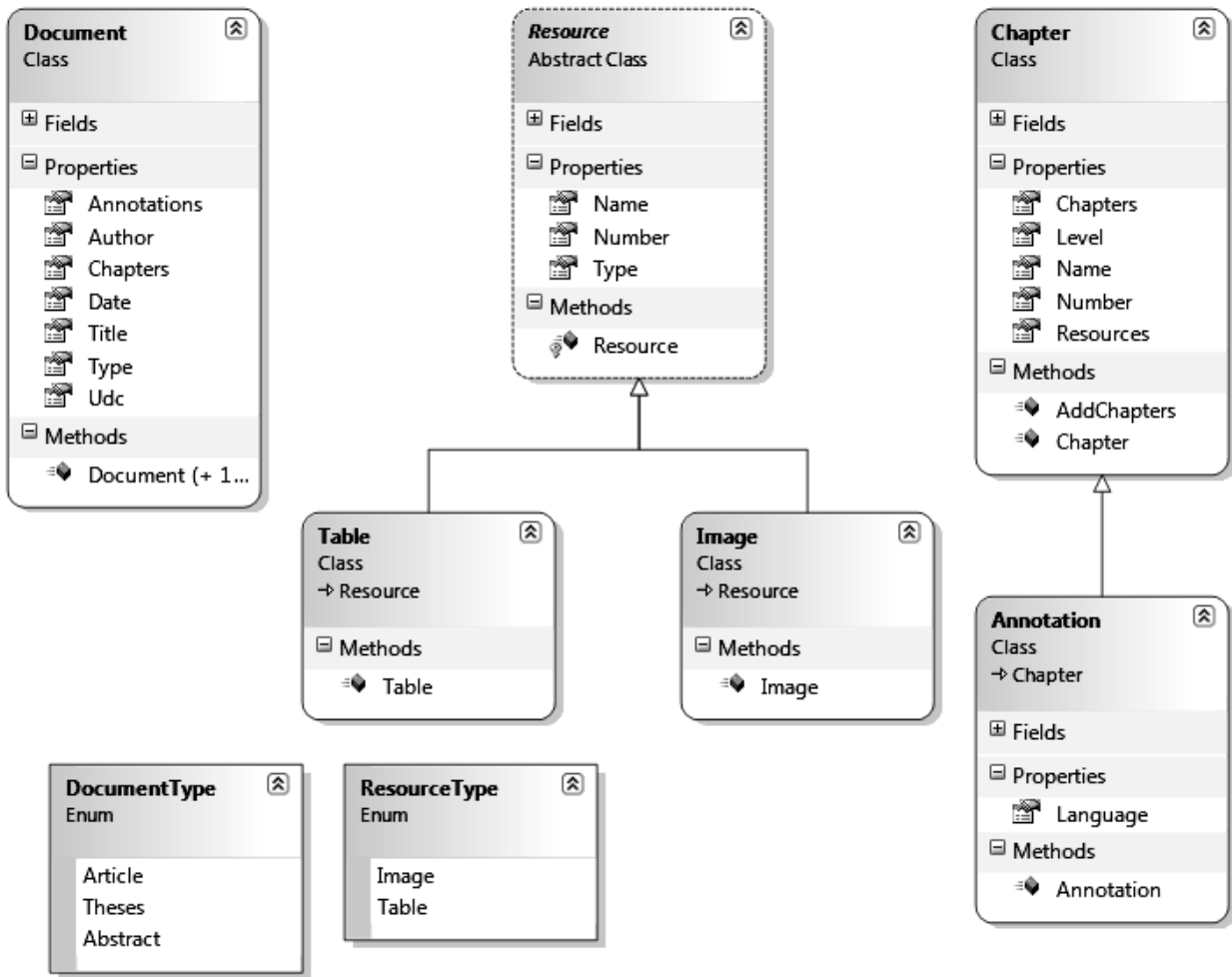


Рис. 7. Діаграма класів агрегату Document

Для просторів даних необхідно забезпечити можливість працювати з програмним продуктом, якого теоретично може і не бути на робочій станції користувача. Якщо не передбачити такої можливості, то необхідно передбачити розроблення сховища даних такої структури, щоб у нього можна було завантажити дані з джерел даних для забезпечення відповіді на запит користувача.

6.1. Технології та засоби реалізації сховищ даних

Порівняльну характеристику засобів реалізації сховищ даних подано в таблиці.

Порівняльна характеристика засобів реалізації сховищ даних

Засіб	Переваги	Недоліки
1	2	3
Oracle Warehouse Builder Oracle Data Integration Oracle Optimized Warehouse Hyperion	СУБД рівня корпорації; можна використовувати як компоненту орієнтованої на дані архітектури в середовищі SOA або BI. Містить: переміщення даних, синхронізацію, перевірку якості даних, керування даними, сервіси перевірки актуальності даних	Для інтеграції необхідно попередньо описати джерела даних та в ручному режимі налагодити процедури перевірки якості даних
Database Application Server	Платформа для створення і розгортання розрахованих на багато користувачів мережних програм для Web, клієнтами яких можуть бути як стандартні браузеры, так і Java-застосування і аплеты)	ПЗ проміжного рівня

1	2	3
SQL Server 2008	Містить засоби інтеграції Integration Services, аналізу Analysis Services, формування звітів Reporting Services, набір інструментів керування сховищами даних Management Studio, набір інструментів розроблення застосунків інтелектуального опрацювання даних Business Intelligence Development Studio	Як і для Oracle, працює з наперед відомими джерелами
Biz Talk	Сервер інтеграції дає змогу аналізувати текстові дані та записувати у сховище даних; функціонує як на внутрішньокорпоративному, так частково і на міжкорпоративному рівнях	Функціонує за принципом сповіщень, що робить якість інтеграції сильно залежною від користувача
Informix MetaCube	Один з продуктів (INFORMIX-MetaCube for Excel) дає змогу здійснювати багатомірний аналіз дуже великих даних безпосередньо з Excel-таблиць. INFORMIX-MetaCube Agents дає змогу опрацювати запити у фоновому режимі.	Розрахований на розроблення сховищ даних, але засоби інтеграції поступаються аналогічним в SQL Server чи Oracle
Visual Warehouse (IBM)	До сімейства програмних продуктів IBM входить проміжний сервер DataJoiner, який дає змогу інтегрувати дані з реляційних БД та завантажувати дані з текстових файлів	Обмежений перелік підтримуваних операційних систем – для версії 5.2 від квітня 2009 р.
Netezza	Архітектура з масовим паралелізмом без поділу ресурсів (на рівні зберігання) і симетрична багатопроцесорна архітектура (на рівні хоста), прискорення опрацювання даних під час передавання до сховища	Закрита технологія, масштабованість до сотень терабайтів, недостатньо розроблені засоби інтеграції
Teradata	За рахунок технології масового паралелізму сховище Teradata можна масштабувати до декількох петабайтів	Закрита архітектура, не підтримує SaaS

Засоби інтеграції можна розділити на два умовні класи: інтеграція застосунків та інтеграція веб-застосунків. Засоби інтеграції застосунків реалізуються за допомогою проміжного шару, спеціалізованих засобів та серверів інтеграції. Засоби інтеграції застосунків розроблено в Oracle, SQL Server. Узагальнену схему інтеграції даних подано на рис. 8.



Рис. 8. Узагальнена схема інтеграції даних

Для інтеграції геоінформації та баз даних використовують спеціалізовані засоби, які перетворюють векторні дані на спеціальний формат.

Простір даних вимагає набагато більше технологічних та методичних рішень, оскільки у ньому опрацюється інформація з різними структурами даних, наперед невизначеними, а також використовуються різні засоби опрацювання та зберігання даних.

6.2. Технології та засоби реалізації просторів даних

Якщо розглядати технології, які дають змогу реалізувати можливості просторів даних, то насамперед необхідно зупинитися на *грід* та *хмарних обчисленнях* (англ. Cloud Computing).

Сервіс-орієнтована грід-технологія надає нові можливості, яких не було в мережах, організованих за схемою реєг-to-реєг або клієнт-сервер. Функціонування і взаємодія сервісів близькі до технології мультиагентних систем, а роль інтелектуальних агентів виконують грід-сервіси. При цьому в такому поданні вони мають низку переваг перед web-сервісами. Серед них:

- можливість реалізації функціональності пошуку даних, не обмеженої набором процедур, реалізованих на сервері сховища даних;

- можливість здійснювати аналіз як в глобальних, так і в корпоративних мережах;

- можливість продовжувати роботу сервісів-агентів пошуку та збирання даних і після виконання конкретного запиту;

- вбудована можливість передавання прав доступу до даних від користувача до усієї послідовності грід-сервісів за допомогою цифрового сертифікату.

Сервіс-орієнтований підхід до слабкоз'язаних масивів даних як до простору даних дає змогу вже сьогодні створювати сервіси нового рівня, що оперують не тільки БД або метаданими, але і працюють безпосередньо з Web-даними та іншими слабкоструктурованими ресурсами. При цьому відчувається потреба застосування доопрацьованих технологій СКБД для просторів даних для того, щоби з нових позицій вирішувати проблему інтеграції федеральних гетерогенних інформаційних ресурсів.

Хмарні обчислення— технологія опрацювання даних, за якою програмне забезпечення надається користувачеві як Інтернет-сервіс. Користувач має доступ до власних даних, але не може керувати і не повинен піклуватися про інфраструктуру, операційну систему і власне програмне забезпечення, з яким він працює.

Сьогодні є такі *реалізації хмарних обчислень*, пов'язані з просторами даних.

a. Windows Azure – хмарна операційна система компанії Microsoft, призначена для розроблення й запуску веб-застосунків, які виконуються на сервері постачальника, а не на комп'ютері користувача. Входить до складу платформи Microsoft Azure. Використовує реляційні структури для зберігання даних. Також передбачається використання SharePoint в хмарі.

b. Google App Engine – це платформа, яка дає змогу використовувати одну інфраструктуру для створення й хостинга своїх застосунків. Використовує нереляційне розподілене сховище даних.

Порівняємо обидві технології.

У хмарі, як і в грід, акумулюються потужності, щоб одержати економічніше, масштабованіше рішення і позбавити користувача необхідності працювати винятково на власному апаратному та програмному забезпеченні. Проте за десять років з моменту появи ідеї грід принципово змінилися обсяги опрацьованих даних, з'явилися віртуалізаційні рішення.

Архітектури грід і хмар відрізняються, оскільки вони створювалися за різних передумов. На першу вплинуло прагнення ефективніше використовувати дорогі розподілені обчислювальні ресурси, зробити їх динамічними й однорідними. Тому архітектура побудована на інтеграції вже існуючих ресурсів, враховуючи апаратне та програмне забезпечення, операційні системи, локальні засоби, що забезпечують керування й безпеку. У результаті створюється «віртуальна організація», ресурси якої, переведені в логічну форму, можуть використовуватись членами тільки цієї організації. Існування цієї організації підтримується п'ятьма рівнями протоколів, інструментами й сервісами, побудованими поперех них (рис. 9, а). Нижнім є інфраструктурний рівень, що поєднує

комп'ютери, системи зберігання, мережі, репозиторії кодів. Вище нього розташований рівень зв'язності, на якому визначені комунікаційні протоколи й протоколи аутентифікації. Ресурсний рівень забезпечує надання ресурсів, можливості керування ними, поділ між окремими користувачами й оплату. Колективний рівень доповнює ресурсний, даючи змогу оперувати наборами ресурсів. Рівень додатків слугує для підтримки роботи додатків.

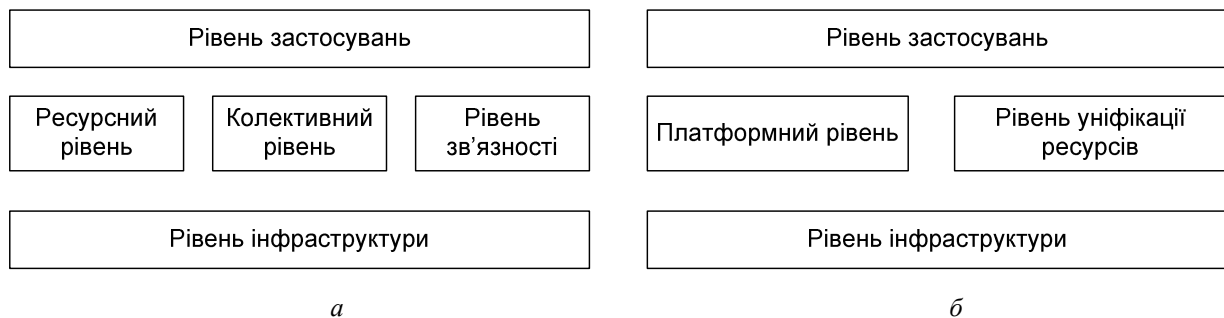


Рис. 9. Порівняння архітектури грід та хмарного обчислення

Архітектура хмар відкрита для доступу через Мережу, а не тільки в рамках грід. Звертаються до пулів обчислювальних ресурсів і систем зберігання даних за стандартними протоколами, наприклад таким, як WSDL і SOAP, або за допомогою технологій Web 2.0 (REST, RSS, AJAX), а також через існуючі технології грід. Протоколи хмар можна розділити на чотири рівні (рис. 9 б). Інфраструктурний рівень містить «сирі» комп'ютерні ресурси (сервери, системи зберігання, мережі). Рівень уніфікації ресурсів містить ті самі ресурси, але в абстрагованому виді – їх можна подати користувачам верхнього рівня як віртуалізовані сервери, кластери серверів, файлові системи й СУБД. Рівень платформ додає набір спеціалізованих інструментів, що зв'яже ПЗ й сервіси поверх універсальних ресурсів, утворюючи середовище для розроблення й впровадження додатків.

Висновки

Розглянуто аналіз таких засобів побудови систем прийняття рішень, як сховища та простори даних, а також загальний аналіз галузі комп'ютингу.

Наукова новизна статті полягає у встановленні формальних відмінностей між такими об'єктами, як бази даних, сховища даних та простори даних.

Практичне значення статті полягає у визначенні основних задач компонент систем прийняття рішень та зв'язку між ними.

1. Lenzerini M. *Data Integration: A Theoretical Perspective* // *PODS 2002*. pp. 233–246. [Електронний ресурс]. [Режим доступу] - <http://www.dis.uniroma1.it/~lenzerin/homepage/talks/TutorialPODS02.pdf>.
2. Papakonstantinou Y. *Object Exchange Across Heterogeneous Information Sources* / Papakonstantinou Y., Garcia-Molina H., Widom J. // *IEEE International Conference on Data Engineering, Taipei, Taiwan, 2005*. – P. 251–260.
3. Halevy A. *Answering queries using views: A survey* / Halevy A. // *The VLDB Journal*. – 2001. – № 10. – P. 270–294.
4. Kalinichenko L.A. *Extensible ontological modeling framework for subject mediation* / Kalinichenko L.A., Skvortsov N.A. // *Proc. of the Fourth Russian Conference on Digital Libraries (RCDL'2002)*. – Dubna: JINR, 2002. – V. 1. – P. 99–119.
5. Калиниченко Л.А. *Методы и средства интеграции неоднородных баз данных* / Л.А. Калиниченко. – М.: Наука, 1983. – 420 с.
6. *Основные концепции и подходы при создании контекстно-поисковых систем на основе реляционных баз данных*. – [Електронний ресурс]. – [Режим доступу] http://www.citforum.ru/database/articles/search_sys.shtml.
7. Рогушина Ю.В. *Формирование тезауруса предметной области как средства моделирования информационных потребностей пользователя при поиске в Интернете* / Ю.В. Рогушина, А.Я. Гладун // *Вестник компьютерных и информационных технологий*. – М., 2007. – № 1. – С. 26–33.
8. Селиверстова А.В. *Совершенствование механизма отбора информации для принятия управленческих решений: Автореф.* – Челябинск, 2002. – 20 с.
9. Тузовский А.Ф. *Системы управления знаниями (методы и технологии)* / Под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.

В. Висоцька, О. Окрушко
 Національний університет “Львівська політехніка”,
 кафедра інформаційних систем та мереж

ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗПОДІЛУ ДАЙДЖЕСТІВ МІЖ ПРАЦІВНИКАМИ ЕЛЕКТРОННИХ ЗАСОБІВ МАСОВОЇ ІНФОРМАЦІЇ

© Висоцька В., Окрушко О., 2012

Розглянуто питання розроблення уніфікованих методів опрацювання інформаційних ресурсів систем електронної контент-комерції. Розроблено формальну модель та узагальнену типову архітектуру систем електронної контент-комерції, а також методи проектування та реалізації системи електронної контент-комерції на прикладі Інтернет-журналу, який відображає результати теоретичних досліджень.

Ключові слова: контент, інформаційний ресурс, Інтернет-журнал, системи електронної контент-комерції.

This paper is devoted to the development of unified methods for processing information resources in the systems of electronic content commerce. A formal model and generalized typical architecture of systems of electronic content commerce are declared. Methods of designing and implementation of systems of electronic content commerce on the example of online Magazine, which reflects the results of theoretical research, are developed.

Key words: content, information resource, Internet Magazine, systems of electronic content-commerce.

Вступ. Постановка проблеми

Одною з найважливіших проблем в роботі інтернет-видань є надмірне завантаження персоналу опрацюванням великої кількості інформації щодня для постійного пошуку та аналізу актуальних тем для написання статей. Це призводить до необхідності використання частини ресурсів персоналу для розподілу дайджестів, тем, завдань між виконавцями; займає багато часу і ресурсів; не дозволяє розподілити ефективно контент (наприклад, дайджести) через суб'єктивізм та неуважність модераторів під час визначення якості роботи журналістів та редакторів. Тому постає питання про можливість автоматизації розподілу дайджестів між робітниками електронних засобів масової інформації із введенням статистики якості роботи виконавців [1, 2].

Аналіз останніх досліджень та публікацій

Контент (англ. content – зміст) – це інформаційне змістовне наповнення (наприклад, тексти, графіка, мультимедіа) інформаційного ресурсу; множина всіх значень і величин, якими оперує ІС; узагальнене поняття даних без наперед визначеної структури [2]. Комерційний контент – це об'єкт бізнес-процесів систем електронної контент-комерції (табл. 1), наприклад, інформаційний продукт або вміст Web-сайта таких електронних засобів масової інформації, як інтернет-газета, інтернет-журнал, інтернет-видання, інтернет-видавництво тощо [2].

Таблиця 1

Основні характерні риси систем електронної контент-комерції

Назва	Характеристика
Віртуальність	Відсутність особистого контакту між суб'єктами процесу купівлі/продажу.
Інтерактивність	Адекватне інформаційне забезпечення запиту користувача у інтерактивному режимі.
Глобальність	Відсутність часових, просторових, асортиментно-товарних, адміністративних меж.
Динамічність	Спроможність on-line торгівлі до моментальних змін й адаптації з появою нових умов.
Ефективність	Забезпечення попиту, прибутку, економічних вигод, соціального ефекту.

Інтернет-журнали, наприклад, Top Gear (topgear.com), Drive (Drive.ru), AutoDiary (auto-diary.ru) та Автоцентр (autocentre.ua) займають одні з перших місць рейтингів популярності та володіють перевагами використання [1, 2], зокрема: користувач має справу з розширеним контентним тематичним потоком (архіви, інші інформаційні ресурси) через множину гіперпосилань; через форуми/конференції відсутні межі між автором і читачем, а користувач має можливість брати участь у виробництві інформаційного продукту; відсутність цензури та придушення авторської думки; Інтернет-публікації є оперативними та не обмеженими терміном виходу номера; невелика собівартість та децентралізованість сприяє розвитку спеціалізованих видань та забезпечує свободу слова і самовираження. Незважаючи на велику кількість переваг, інтернет-журнали мають недоліки [2]: збереження анонімності автора при публікації матеріалу призводить до зловживань як автором (дезінформація, прихована реклама), так і псевдоавторами (плагіат); невисокий рівень грамотності в інтернеті через відсутність контролю з боку редактора.

При цьому важливим є забезпечення інваріантності середовища систем електронної контент-комерції до модифікації інформаційних ресурсів у таких змінах [1, 2]: способів подання, форматів та внутрішньої організації контенту; середовища зберігання контенту, фізичних одиниць зберігання, технічних засобів; вимог користувачів контенту, поява нових вимог та категорій користувачів; порядку розподілу контенту та способів доступу користувачів.

Отже, виникає проблема створення єдиного концептуального опису всього інформаційного ресурсу, який має на меті стабільне підтримання зовнішніх/внутрішніх позначень контенту відповідно до їх завдань, вимог та змін. Тому необхідно класифікувати інформаційні ресурси систем електронної контент-комерції для подальшого дослідження їх природних, технологічних та споживчих якостей з метою виявлення характерних та специфічних властивостей, а також закономірностей та особливостей їх формування та застосування. За основу класифікації взято основні властивості контенту в системах електронної контент-комерції як синтаксис (принципи формального подання контенту), структура (правила побудови та впорядкування контенту) та семантика (формування змісту/розуміння/функцій для визначення основних завдань та порядку застосування контенту). На їх основі обрано основні фактори класифікації [1, 2]: способи подання контенту у системах електронної контент-комерції; способи структурування інформаційного ресурсу; способи доступу до інформаційного ресурсу систем електронної контент-комерції; призначення інформаційного ресурсу систем електронної контент-комерції.

Формують інформаційний ресурс систем електронної контент-комерції різними шляхами: гомогенізацією, розподілом або інтеграцією (рис. 1) [2].

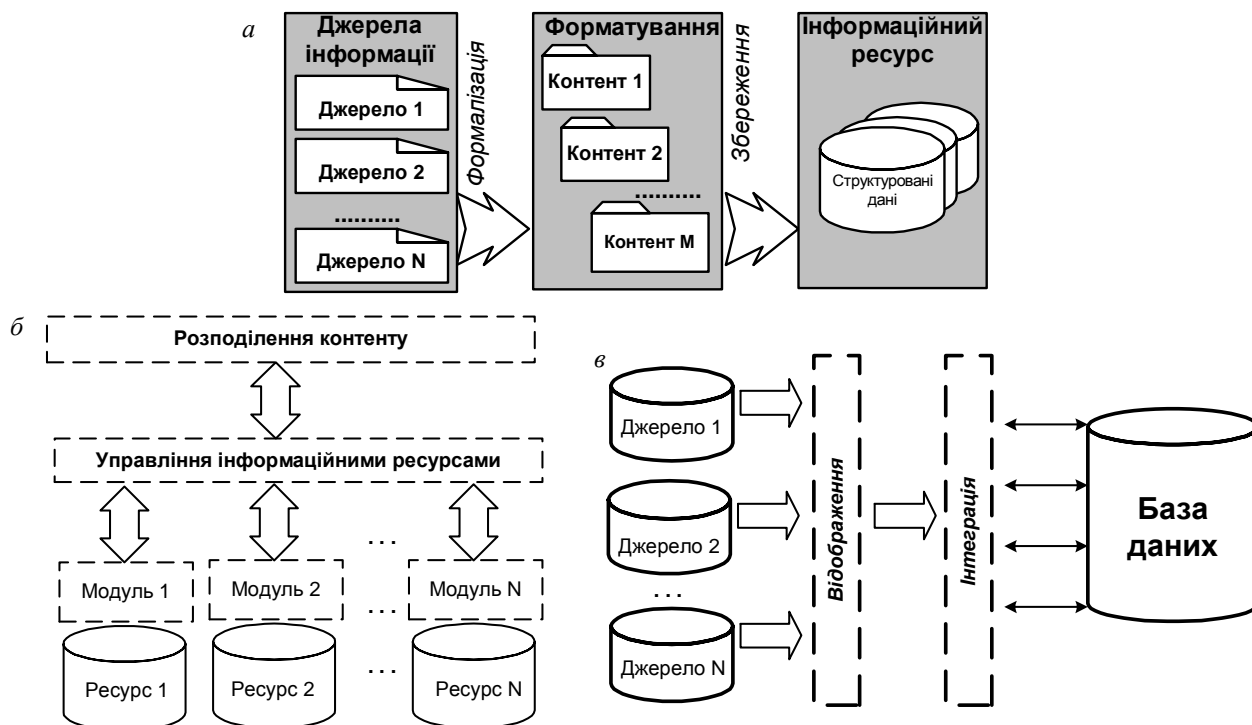


Рис. 1. Побудови інформаційного ресурсу шляхом:
а – гомогенізації; б – розподілу; в – інтеграції контенту

Корпорації EMC, IBM, Microsoft Alfresco, Open Text, Oracle і SAP розробили специфікації Content Management Interoperability Services (CMIS) на інтерфейс Web-сервісів, покликаних забезпечити взаємодію між системами електронної контент-комерції [2]:

- 1) підтримка інтеграції даних (забезпечення роботи нових застосувань із наявними репозиторіями даних і накопиченим в них контентом);
- 2) незалежність управління контентом різних репозиторіїв за допомогою Web-сервісів;
- 3) підтримка Web 2.0 (забезпечення загальних Web-сервісів та інтерфейсів для спрощення розроблення застосувань);
- 4) незалежність платформи (створення платформи, незалежної від мови контенту);
- 5) формування контенту (підтримка розроблення композитних застосувань і колажів, тобто нового контенту, складеного із контенту декількох джерел, який виглядає як єдине ціле).

Web 2.0 позначають ряд технологій та послуг інтернету та описують зміну сприйняття інтернет-користувачами [1, 2, 4]. Зміни полягають у посиленні комунікативності, співробітництва, безпечного використання контенту та загальному розвитку функціональності мережі [9, 10]. Термін не означає поновлення технічних специфікацій, а лише зміни у шляхах використання мережі розробниками програмного забезпечення та кінцевими користувачами (табл. 2) [9, 10].

Таблиця 2

Порівняння параметрів Web 1.0 та Web 2.0

Назва	Web 1.0	Web 2.0	Нові властивості
Актори	Розробник-користувач/автор контенту і читач	Користувач як співрозробник, читач як співавтор або товариство	Право на участь; скасування модерації
ПЗ	ПЗ створювалось для ПК, це товар; закриті вихідні коди; ліцензійний продаж; прив'язка ПЗ до обладнання; націленість на винахід; запланований реліз; для перегляду контенту є браузер.	ПЗ для Web; це сервіс/застосування; відкриті вихідні коди, open-source software; безкоштовне ПЗ; ПЗ над обладнанням; вічна бета; пошук застосування винайденому; альтернативні засоби сприйняття.	Web-платформа; зняття/розмиття бар'єрів/обмежень: доступність, універсальність, спрощення.
Контент	Поповнення БД; плата автору або наймання модераторів; таксономічна організація даних (ієрархія рубрик); засоби збереження даних – каталог, бібліотека, сховище; односторонні посилання; форма подання – персональні сторінки; статичний сайт; адресу у сторінки сайта; джерело – розум автора; меню навігації сайта для роботи з даними сайта; копірайт; для сприйняття контенту мандрують сайтом по посиланню чи закладці;	Поповнення баз даних – те, що має один, відразу стає доступне кожному; дані організують фолксономічно; засоби використання даних – API-інтерфейси; автоматичні двосторонні посилання; форма подання – блоги; динамічний сайт; адресу має мікро-елемент контенту; джерело – колективний розум; інтерфейс для роботи з даними по всій мережі; “вільна” ліцензія GNU FDL; для сприйняття контенту не потрібно відвідувати сайт – можливо читати RSS-стрічки.	Мережа як єдиний колективний розум, атомізація контенту, агрегація, синдикація.
Події	Замовлення та виготовлення ПЗ; публікація контенту авторами і сприйняття його читачами; звертання до третьої особи – посередника для задіяння його ресурсів; великі, нечисленні угоди;	Співпраця через відділ технічної підтримки ПЗ; взаємодія, додавання властивостей, цінності, створення спільного контенту кожним учасником; самообслуговування, яке засноване на партнерській архітектурі сервісу – сервіс лише посередник між користувачами, які використовують їхні ресурси; дрібні численні транзакції.	Співпраця; самодіяльність; масові одиничні взаємовідносини.
Цінність	Цінність в ПЗ; Інтернет цінний як джерело інформації;	Цінність в базах даних та сервісах роботи з ними; Інтернет – інструмент комунікацій.	Робота з базою даних; економія часу/уваги.

Життєвий цикл контенту (англ. Content lifecycle) – це складний процес, який проходить контент під час управління через різні етапи публікації [2]. Існуючі моделі життєвого циклу контенту не містять всіх етапів процесів опрацювання інформаційних ресурсів: формування,

управління та реалізація контенту (табл. 3) [4]. Кількість контентних потоків більша, ніж шляхів переміщення товарів на промислових підприємствах. Значна частина контентних потоків складається з легко формалізованих і автоматизованих процедур. Основна проблема – відсутність загального підходу до процесу моделювання, проектування та розроблення систем електронної контент-комерції (СЕКК) [2]. Відсутність загальної та детальної класифікації систем електронної контент-комерції, що приводить до проблеми визначення і формування загальних методів проектування та розроблення архітектури та алгоритмів функціонування цих систем. Це обґрунтовує мету, актуальність, доцільність та напрями дослідження. Наявні системи електронної комерції (СЕК) не підтримують всього життєвого циклу контентного потоку та не вирішують основних проблем опрацювання інформаційних ресурсів – формування та реалізації контенту (табл. 4).

Таблиця 3

Порівняння моделей життєвих циклів контенту

Автор моделі Content lifecycle	Формування	Управління	Реалізація
McKeever Susan	+/-	-	+/-
Bob Boiko	+/-	+/-	+/-
Gerry McGovern	+/-	-	+/-
JoAnn Hackos	+/-	-	+/-
Ann Rockley	+/-	+/-	+/-
Russell Nakano	+/-	-	+/-
The State government of Victoria	+/-	-	+/-
АІМ	+/-	+/-	+/-
СМР organization	+/-	+/-	-
Bob Doyle	+/-	+/-	+/-
Woods Randy	+/-	+	+
Halverson	+	+/-	+/-

Таблиця 4

Порівняння особливостей систем

Назва характеристики	СЕК	СЕКК
Нематеріальність товару	-	+
Постійна кількість товару	-	+
Ріст кількості різновиду товару	+/-	+
Відсутність складу	-	+
Збереження товару в базах даних	-	+
Ефективність просування товару за ключовими словами	+/-	+
Ефективність пошуку товару за ключовими словами	+/-	+
Автоматичне виявлення та ліквідація дублювання товару	-	+
Автоматичне визначення старіння товару за змістом	-	+
Автоматичне визначення актуальності товару	+/-	+
Автоматичний аналіз аудиторії	+/-	+
Автоматичне формування дайджестів	-	+
Автоматичний розподіл товару між учасниками	+/-	+
Автоматичний розподіл дайджестів між працівниками	-	+
Автоматичне формування товару	-	+
Автоматичне форматування товару	-	+
Вплив досвіду користувача на збільшення обсягу продажів	+/-	+

Автоматичний розподіл контенту передбачає декілька етапів: формування списку об'єктів розподілу (наприклад, статей, ПЗ, книг або дайджестів); визначення критеріїв (ознак) розподілу контенту з отриманого списку (процент унікальності контенту; кількість звернень до контенту; користувацька оцінка; час перегляду); оцінювання певних параметрів з метою використання в процесі розподілу. Наведені критерії не можуть вважатися однаковими за значенням та важливістю при аналізі їх загалом та обчисленні зведеної оцінки якості роботи. Крім цього, оцінювати кожну публікацію необхідно не поодиноці, а в комплексі з іншими публікаціями, які підлягають оціню-

ванню в цей момент. Одним з найважливіших критеріїв розподілу дайджестів між працівниками електронних засобів масової інформації є процент унікальності контенту в попередніх публікаціях кожного окремого журналіста [3]. Зауважимо, що проста перевірка на дублювання контенту загалом чи окремих його блоків з уже наявними в мережі зразками не завжди дозволяє виявити плагіат, оскільки існує метод рерайтингу (від англ. rewrite – «переписувати»). Хоча дублікат може виглядати стилістично гіршим за оригінал, проте процент унікальності дуже високий і звичайні сервіси з перевірки унікальності не знайдуть у цьому випадку плагіату. Проблема рерайтера полягає тільки в тому, що більшість контенту містить деякі терміни, до яких важко/неможливо підібрати синоніми. Відомими системами визначення унікальності контенту є Praide Unique Content Analyser 2, FIndCopy та Miratools. Існують такі види запозичення [8]: повне або часткове копіювання тексту з одного джерела; копіювання і компонування тексту з декількох джерел; копіювання тексту з іншого джерела і зміна послідовності слідування частин тексту.

Для приховування факту запозичень застосовують такі підходи.

1. Коригування родів, чисел та часів слів текстової інформації.
2. Незначна зміна запозиченого тексту.
3. Скорочення запозиченого тексту (видалення речень, абзаців, рисунків, формул тощо).
4. Заміна кирилических символів на аналогічні за накресленням латинські та навіпаки.
5. Здійснення ручної або автоматичної синонімізації тексту.

Це враховують, перевіряючи унікальність текстів та визначаючи якість роботи. Отримавши процент унікальності тексту в публікації працівника електронного видання, оцінюють якість цієї публікації та заносять отриману оцінку в таблицю рейтингів. Із збільшенням кількості публікації можна точніше оцінити якість та продуктивність кожного працівника електронного видання. Із збільшенням кількості критеріїв оцінювання можна охопити ширший спектр аспектів роботи працівника видання. Існує велика кількість методів вирішення проблеми комплексного оцінювання будь-чого. Одним з найвідоміших є метод аналізу ієрархій (MAI) – математичний інструмент системного підходу до складних проблем прийняття рішень [6]. MAI, який розробив американський математик Томас Сааті, дає змогу раціонально структурувати складну проблему прийняття рішень у вигляді ієрархії, порівняти їх і кількісно оцінити альтернативні варіанти рішення [5, 6].

Формулювання цілі статті

Вихідною інформацією процесу функціонування СЕKK є дані про призначення й умови роботи системи, які визначають основну мету моделювання СЕKK і дають змогу сформулювати вимоги до формальної моделі системи S та моделей управління контентом [2]. Формальна модель СЕKK $S = \langle X, C, V, H, Function, T, Y \rangle$ – це множини величин, що описують процес функціонування системи і утворюють такі підмножини: вхідні впливи на систему $x_i \in X$ ($i = \overline{1, n_X}$, $X = \{x_1, x_2, \dots, x_{n_X}\}$), впливи потоку контенту на систему $c_r \in C$ ($r = \overline{1, n_C}$, $C = \{c_1, c_2, \dots, c_{n_C}\}$), впливи зовнішнього середовища $v_l \in V$ ($l = \overline{1, n_V}$, $V = \{v_1, v_2, \dots, v_{n_V}\}$), внутрішні параметри системи $h_k \in H$ ($k = \overline{1, n_H}$, $H = \{h_1, h_2, \dots, h_{n_H}\}$), вихідні характеристики системи $y_j \in Y$ ($j = \overline{1, n_Y}$, $Y = \{y_1, y_2, \dots, y_{n_Y}\}$) та час транзакції опрацювання інформаційного ресурсу $t_i \in T$ ($i = \overline{1, n_T}$, $T = \{t_1, t_2, \dots, t_{n_T}\}$) [7].

Величини x_i , c_r , v_l , h_k , y_j є елементами непересічних підмножин і містять детерміновані і стохастичні складові [2, 4, 7]. Процес функціонування S описується функцією $y_j(t_i + \Delta t) = Function(x_i, c_r, v_l, h_k, t_i)$, де x_i – це запити відвідувачів/користувачів до СЕKK. Згідно з Google Analytics y_j – це кількість відвідувань за період часу Δt , середній час перебуття на сайті (хв:с), показник відмовлень (%), досягнута мета; динаміка (%), кількість всього перегляду сторінок, кількість перегляду сторінок за одне відвідування; нові відвідування (%); абсолютно унікальні відвідувачі; джерело трафіка у % (пошукові системи, прямий трафік або інші сайти). Впливи

величин c_r , v_l , h_k , на y_j як результат роботи СЕКК є невідомими та недослідженими [2]. Вивчення динаміки потоку комерційного контенту та побудова моделей опрацювання інформаційних ресурсів СЕКК є важливими та актуальними. Формальна модель СЕКК не розкриває механізмів управління контентом. Формальні моделі управління контентом призначені лише для визначення процесів старіння (актуальності) контентного потоку, а деякі із них (логістична, аналітична) – і для тематичного потоку. Вони не вирішують проблем формування, розподілу та реалізації контенту і вирішують не всі проблеми управління контентом, наприклад, подання множини контенту кінцевому користувачу згідно із його запитом, історією або інформаційним портфелем, автоматичне виявлення тематичних сюжетів, автоматичне формування дайджестів, інформаційних портретів, побудова таблиць взаємозв'язку понять, розрахунок рейтингів понять, збирання інформації з різних джерел та її форматування, виявлення ключових слів/понять та дублювання змісту, автоматична рубрикація, вибіркоче поширення контенту. Недолік моделей управління контентом – це відсутність зв'язків між вхідними даними, контентом та вихідними даними в СЕКК [2]. Під час розгляду динаміки тематичних потоків контенту виявлено обмеженість моделей (табл. 5), що відкриває шлях для подальших досліджень [4].

Таблиця 5

Складові систем електронної контент-комерції

Модель	Перевага
Бартона–Кеблера	Описує процес старіння контенту, втрати його актуальності, швидкості розвитку окремих тематик або всього контентного простору; має точний розв'язок у вигляді експоненти.
Просторово-векторна	Визначення значущого терму в потоці контенту та найактуальнішого контенту із множини наявних. Обов'язкове ранжування контенту, використання параметричних множників, що залежать від часу.
Лінійна	Визначення інтенсивності потоку в часі при лінійній динаміці управління тематичного контенту.
Експонентна	Описує процес старіння контенту, втрати його актуальності. Кореляція між окремим контентом несуттєва.
Логістична	Вивчення динаміки окремого тематичного потоку. Розмірність параметрів та їхній вимір не враховують.
Аналітична	Описує процес старіння контенту та втрати його актуальності з використанням словника ключових слів.

Із врахуванням вже описаних задач виникає потреба в розробленні інтелектуальної системи розподілу дайджестів між працівниками електронних засобів масової інформації, яка буде здатна оцінювати якість роботи та розподілу завдань. При цьому система повинна отримувати готовий список дайджестів і надавати до нього доступ усім працівникам окремо за чергою, яка формується відповідно до таблиці рейтингів працівників видання. Таблицю рейтингів формують за методом аналізу ієрархій, оцінюючи готові публікації за декількома критеріями одночасно.

Виклад основного матеріалу

Система електронної контент-комерції має таку структурну схему:

Вхідні дані → *Модуль формування контенту* → *Модуль управління контентом* →
Модуль реалізації контенту → *Інформаційний ресурс*,

а формальна модель – це шістька $S = \langle X, Formation, C, Management, Realization, Y \rangle$, де $X = \{x_1, x_2, \dots, x_{n_x}\}$ – множина вхідної інформації, *Formation* – функція формування контенту, $C = \{c_1, c_2, \dots, c_{n_c}\}$ – множина контенту, *Management* – функція управління контентом, *Realization* – функція реалізації контенту та $Y = \{y_1, y_2, \dots, y_{n_y}\}$ – множина вихідної інформації.

Формальна модель формування інформаційних ресурсів – це

$Formation = \langle X, C, Gathering, Formatting, KeyWords, Categorization, Backup, Dissemination \rangle$,

де X – множина вхідної інформації з Web-сайтів або від модераторів; C – множина контенту; *Gathering* – функція збирання інформації з джерел; *Formatting* – функція форматування

інформації, перетворення на множину контенту; *KeyWords* – функція виявлення ключових слів, понять; *Categorization* – функція автоматичної рубрикації; *Backup* – функція виявлення дублювання змісту контенту; *Dissemination* – функція вибіркового поширення контенту.

Моделі управління інформаційними ресурсами.

1. Генерація сторінок за запитом, структурна схема якої така:

Вхідні дані → *Модуль редагування* → *База даних* → *Модуль подання* → *Інформаційний ресурс*,

а формальна модель – $Management_Q = \langle X, C, Q, R, Edit, Y \rangle$, де X – множина вхідної інформації; C – множина контенту; Y – множина сформованих сторінок; Q – множина запитів; R – функція формування та подання сторінки; $Edit$ – функція редагування та модифікації контенту.

2. Генерація сторінок під час редагування, структурна схема якої така:

Вхідні дані → *Модуль редагування* → *База даних* → *Інформаційний ресурс*,

а формальна модель – $Management_E = \langle C, Edit, Y \rangle$, де C – множина контенту; Y – множина статичних сторінок; $Edit$ – функція редагування та модифікації контенту. Процес формування сторінок описується функцією $\bar{y}(t) = Edit(\bar{c}, Weight, t)$. У разі внесення змін до змісту сайту створюють набір статичних сторінок, t не враховується, тому є інтерактивність між відвідувачем і вмістом сайту.

3. Змішаний тип, структурна схема якої така:

Вхідні дані → *Модуль редагування* → *База даних інформаційних блоків* → *Модуль збору* → *Кеш* → *Модуль подання* → *Інформаційний ресурс*,

а формальна модель – $Management_M = \langle X, C, Q, R, Edit, Caching, Y \rangle$, де X – множина вхідної інформації; C – множина контенту; Y – множина сформованих сторінок; Q – множина запитів; R – функція формування та подання сторінки; $Edit$ – функція редагування та модифікації контенту, $Caching$ – функція формування кешу. Ця модель поєднує переваги перших двох та реалізується шляхом кешування – модуль подання генерує сторінку один раз, надалі її в декілька разів швидше завантажують з кешу, який оновлюють автоматично (після закінчення деякого терміну часу або при внесенні змін до певних розділів сайту) або вручну адміністратором. Інший підхід – збереження інформаційних блоків під час редагування сайту і збирання сторінки з блоків за запитом користувача.

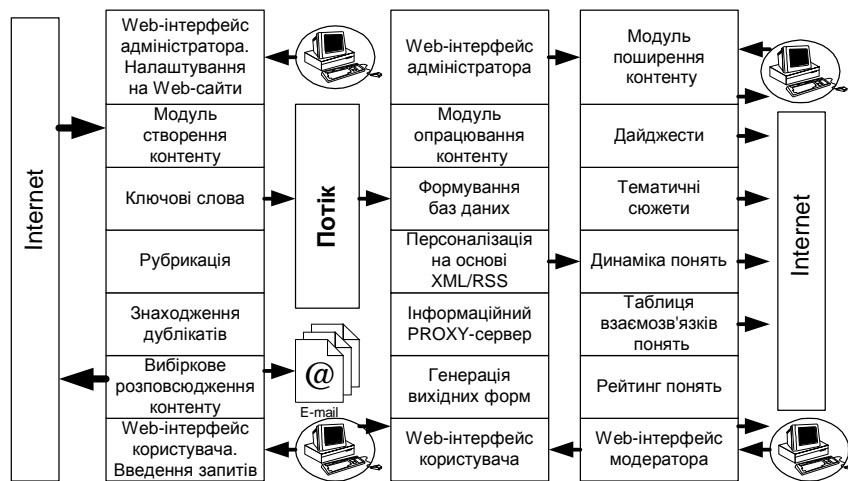
На рис. 2 подано типову схему СЕКК, що забезпечує ознайомлення, вибір категорії контенту, оформлення замовлення, здійснення взаєморозрахунків, відстеження виконання замовлення [4].

Аналіз лексико-граматичної та семантико-прагматичної побудови тексту використовують в модулі автоматичної рубрикації контенту та формування дайджестів. Виконання перерахованих в табл. 7 етапів призводить до формування тематично підібраних масивів контенту, в яких акумулюється інформація про висвітлення всіх аспектів досліджуваної проблеми, враховуючи різноманітність думок і поглядів.

Таблиця 7

Основні етапи роботи модуля рубрикації контенту

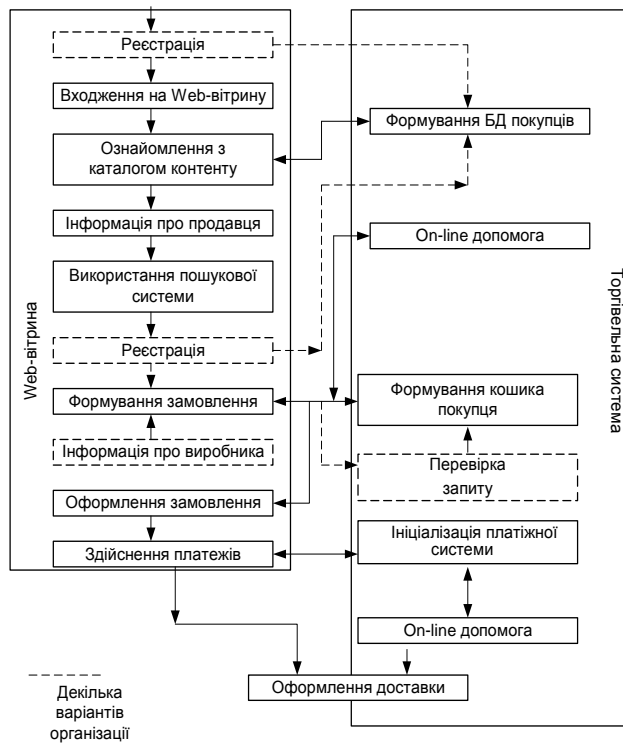
Назва	Призначення етапу
Підготовка	Визначення тематики, мети та об'єкту аналізу, його хронологічні та географічні рамки, принципи відбору.
Класифікація	Формування класифікатора відбору ключових цитат та інструкції для кодувальника.
Кодування	Кодування фрагментів текстової інформації.
Архівация	Збереження фрагментів текстової інформації в базі даних.
Аналіз	Автоматичне опрацювання фрагментів текстової інформації.



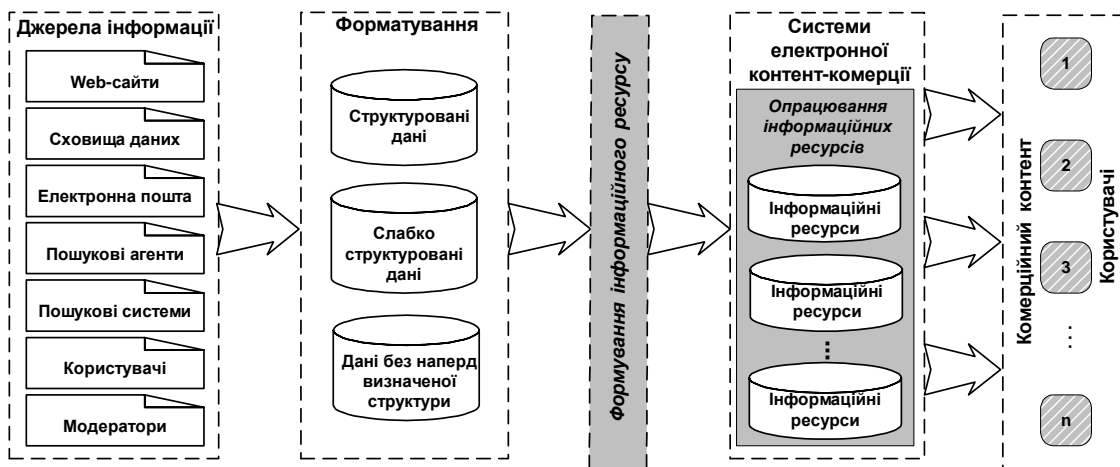
а



б



в



г

Рис. 2. Схема взаємодії модулів (а); архітектури (б); алгоритму роботи (в); функціонування CECC (г)

Основною формою подання і збереження інформації є природна мова, тому ефективність процесу рубрикації контенту залежить від розв'язання проблеми автоматичного опрацювання текстів, кінцева мета якого – розпізнавання їх змісту (табл. 8).

Таблиця 8

Основні етапи опрацювання текстів для розпізнавання змісту

Опрацювання	Призначення етапу
Формальне	Перетворення фрагментів тексту без звернення до аналізу його змісту. Морфологічні дані забезпечують доступ до змісту, опосередкованого через співвідношення одиниць змісту з одиницями виразу.
Змістове (семантичне)	Розпізнавання змісту окремих елементів і логіко-семантичних відношень між ними для подання семантики контенту.
Синтаксичне	Автоматично за наявності лексико-граматичних та граматичних даних до кожного слова синтаксично прив'язують словоформи у реченні.
Морфемне	Сегментування тексту, де виділення префіксів можливе без знання частин мови, а суфіксів – ні: потрібні різні їх набори та процедури відсікання суфіксів для іменників, дієслів, прикметників, прислівників.

Найпростіша ієрархія містить три рівні: мета, критерії та альтернативи. Числа на рис. 3 показують пріоритети елементів ієрархії з погляду мети, які обчислюються в МАІ на основі парних порівнянь елементів кожного рівня щодо пов'язаних з ними елементів вищого рівня. Пріоритети альтернатив обчислюються на завершальному етапі методу шляхом лінійного згортання локальних пріоритетів всіх елементів. У цьому випадку відомі пріоритети критеріїв: найбільшої ваги надають критерію *Унікальність*, оскільки він об'єктивно відображає якість роботи журналіста. Наступним за вагою є *Час читання*, який відображає міру зацікавленості користувачів в конкретному матеріалі. Критерії *Користувацька оцінка* та *Кількість звернень* мають меншу вагу, оскільки не можуть користуватися довірою через легкість фальсифікації їх показників. Так оцінюють всі альтернативи за кожним з критеріїв окремо. З розрахунків випливає, що друга альтернатива є найкращою за обраними критеріями оцінювання.

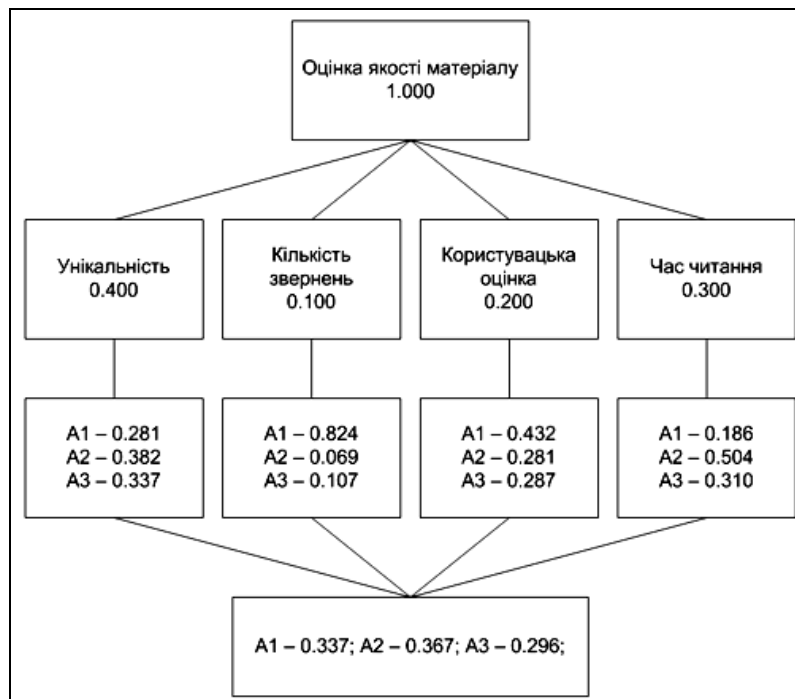


Рис. 3. Ієрархія у випадку трьох альтернатив розподілу дайджестів

Спроектуємо систему автоматичного розподілу дайджестів за допомогою методології IDEF, яку використовують для вирішення завдань моделювання складних систем. IDEF дає змогу відображати та аналізувати моделі діяльності широкого спектра складних систем у різних аспектах. Модель «To be» відображає функціонування системи загалом (рис. 4, а) та основні дані, змінювані під час роботи системи, правила її роботи, вихідні дані та необхідні для роботи ресурси. На рис. 4, б подано декомпозицію, яка поділяє роботу системи розподілу дайджестів на основні етапи.

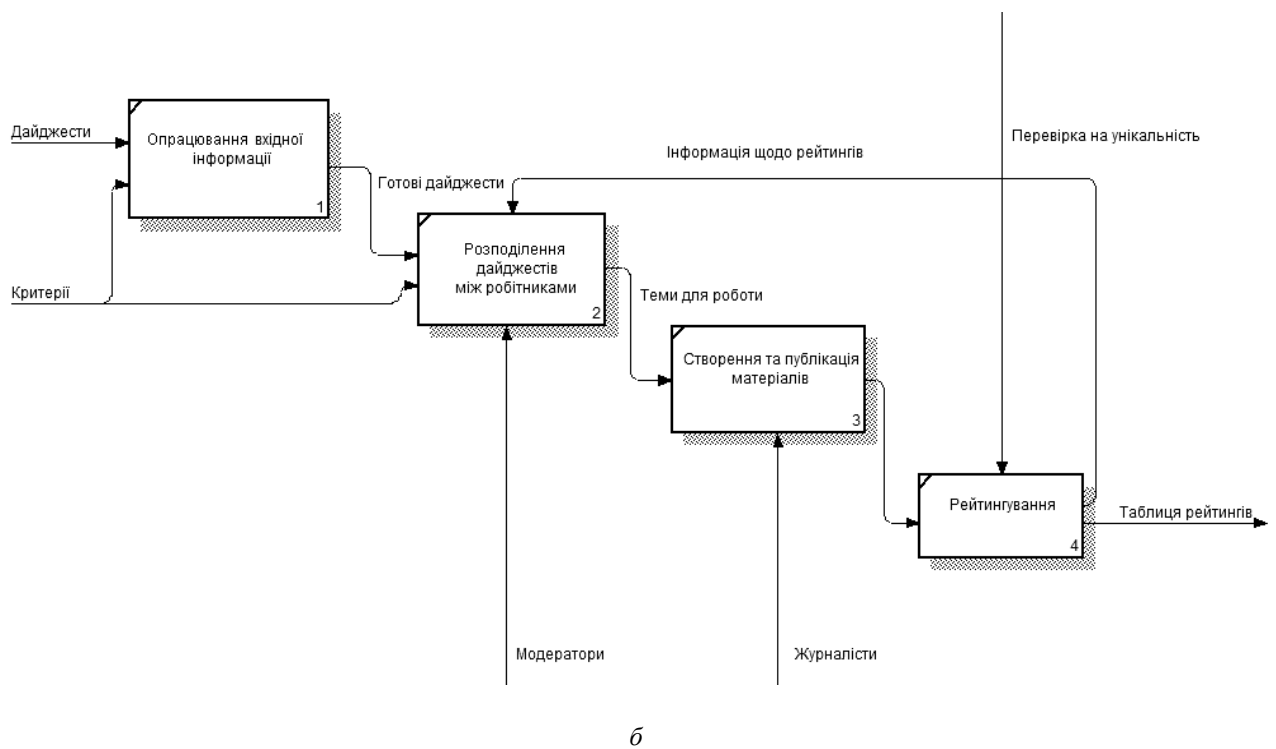


Рис. 4. Функціонування системи розподілу дайджестів загалом (а) та декомпозиція (б)

IDEF3 діаграма системи розподілу дайджестів (рис. 5) відображає послідовність дій з умовами, що виконуються під час роботи системи для досягнення поставленої мети. Вона складається з чотирьох основних робіт, двох розгалужень та чотирьох об'єктів посилань.

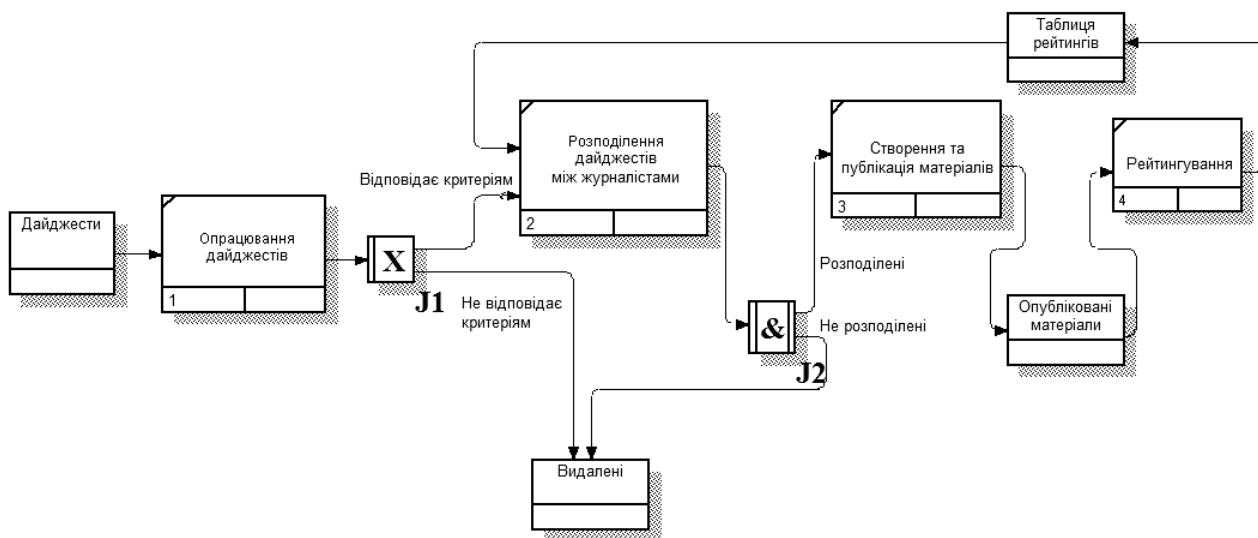


Рис. 5. Загальний вигляд IDEF3 діаграми системи розподілу дайджестів

На DFD діаграмі системи автоматичного розподілу дайджестів (рис. 6) відображено потоки даних та етапи перетворення інформації у системі. В системі використано 3 сховища даних (Джерело дайджестів, готові дайджести та таблиця рейтингів), 4 роботи (опрацювати дайджести, розподілити дайджести між працівниками, створити та опублікувати матеріали, визначити рейтинг за якістю публікації) та 2 зовнішні посилання (Модератори та журналісти). Початковим продуктом є джерело дайджестів, кінцевим – таблиця рейтингів, яку використовують в процесі розподілу дайджестів між працівниками.

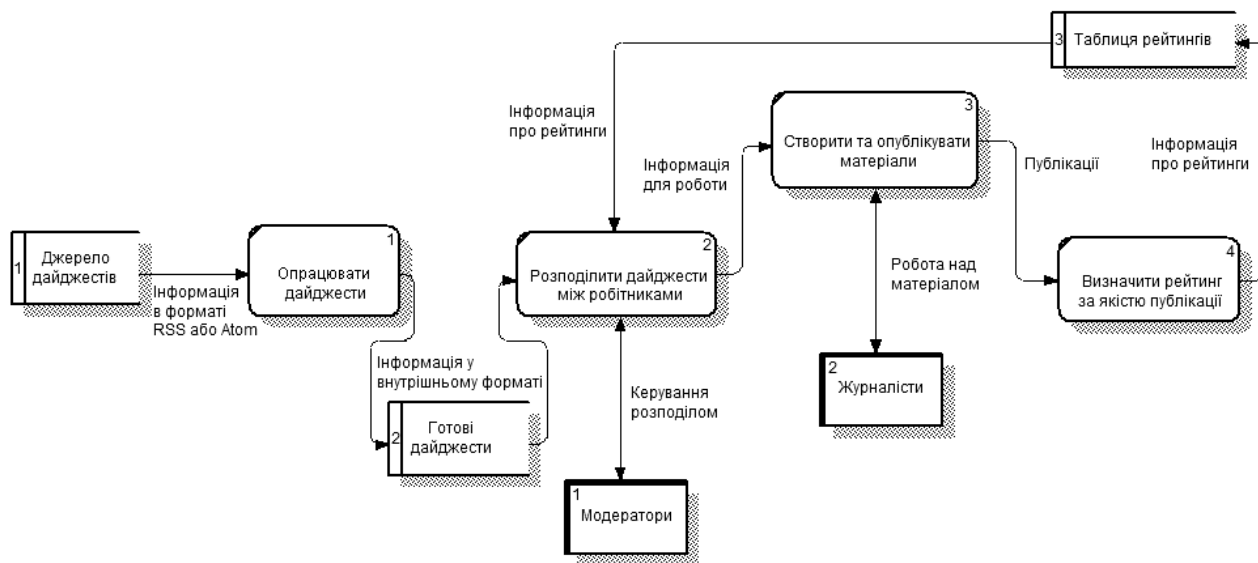


Рис. 6. Загальний вигляд DFD діаграми системи розподілу дайджестів

Висновки

Досліджено проблему керування розподілом робочої сили та робочого часу працівників засобів масової інформації. Подано структуру інтелектуальної системи розподілу дайджестів. Отже, за допомогою поданих вище діаграм стає можливим розроблення програмного продукту, який відповідав би поставленим вимогам. Такий програмний продукт можливо реалізувати у вигляді як самостійної системи, так і додаткового модуля для популярних систем керування вмістом. Варіант з реалізацією як додаткового модуля універсальніший, оскільки робить продукт універсальнішим, та дає можливість працювати з різними системами керування вмістом та їх конфігураціями.

1. Береза А.М. *Електронна комерція* / А.М. Береза. – К.: КНЕУ. – 2002. 2. Берко А.Ю. *Системи електронної контент-комерції. Монографія* / А.Ю. Берко, В.А. Висоцька, В.В. Пасічник. – Львів: Вид-во Нац. Ун-ту “Львівська політехніка”. – 2009. – 612 с. 3. Бондаренко С. *Обнаружение плагиата* / С. Бондаренко // 3DNews. – Режим доступу: http://www.3dnews.ru/software/plagiarism_detection. 4. Ландэ Д.В. *Основы моделирования и оценки электронных информационных потоков: монография* / Д.В. Ландэ, В.М. Фурашев, С.М. Брайчевский, О.М. Григорьев. – К.: ТОВ “Інжиніринг”, 2006. – 348 с. 5. Подиновский В.В. *О некорректности метода анализа иерархий* / В.В. Подиновский // CONTROL SCIENCES. – Режим доступу: <http://ru.mtas.ru/archive/Podinovski.pdf>. 6. Саати Т.Л. *Принятие решений при зависимостях и обратных связях: Аналитические сети* / Т.Л. Саати. – М.: Издательство ЛКИ, 2008. – 360 с. 7. Советов Б.Я. *Моделирование систем (2-е изд.)* / Б.Я. Советов, С.А. Яковлев. – М.: Высшая школа, 1998 р. 8. Шарапов Р.В. *Система проверки текстов на заимствования из других источников* / Р.В. Шарапов // Муромський інститут ГОУ ВПО. – Режим доступу: <http://ceur-ws.org/Vol-803/paper16.pdf>. 9. *Critical Perspectives of Web 2.0. Special issue of First Monday*. – Vol. 13, #3, 2008. – Режим доступу: <http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/issue/view/263/showToc>. 10. Graham P. *Web 2.0* / P. Graham. – Nov. 2005. – <http://www.paulgraham.com/web20.html>.

В. Мухін, Я. Корнага, Л. Снегірєв
Національний технічний університет України
“Київський політехнічний інститут”,
кафедра обчислювальної техніки

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МЕХАНІЗМІВ ПОШУКУ В БАЗАХ ДАНИХ НА ОСНОВІ К-ДЕРЕВ

© Мухін В., Корнага Я., Снегірєв Л., 2012

Запропоновано новий вид дерев для побудови індексів таблиць баз даних: К-дерева. Проведено експериментальне порівняння з В+-деревами з дослідження ефективності використання К-дерев для пошуку в базах даних.

Ключові слова: пошук даних, індекси, В+-дерева, К-дерева.

There are suggested the new type of trees for database tables indexes forming: K-trees. The experimental researches for the B-trees and K-trees features are performed. The researches show that the K-trees ensure the improvement of the data search.

Key words: data search, indexes, B+-trees, K-trees.

Вступ

У базах даних важливим елементом є пошук даних у таблицях, які можуть мати велику кількість рядків, що зберігаються в довільному порядку, та їх пошук за заданим критерієм шляхом послідовного перегляду таблиці рядок за рядком може займати багато часу. Індекс формується зі значень одного чи декількох стовпців таблиці і вказівників на відповідні рядки таблиці та дає змогу шукати рядки, що задовольняють критерій пошуку. Робота з використання індексів прискорюється насамперед за рахунок того, що індекс має структуру, оптимізовану під пошук: наприклад, збалансованого дерева [1, 2].

Зазвичай, що більше індексів, то більша продуктивність запитів до бази даних. Однак за надмірного збільшення кількості індексів падає продуктивність операцій зміни даних (запис, модифікація, видалення), збільшується розмір БД, тому до додавання індексів слід ставитися обережно [1–3].

Базовим апаратом для пошуку даних у БД є В-дерева. В основу цього механізму покладено такі ідеї. По-перше, оскільки йдеться про структури даних у зовнішній пам'яті, загальний час доступу до якої визначається переважно не обсягом послідовно розташованих даних, а часом підведення магнітних головок, то вигідно отримувати за одне звернення до зовнішньої пам'яті якомога більше інформації, враховуючи при цьому необхідність економного використання основної пам'яті. За сформованого підходу до організації основної пам'яті у вигляді набору сторінок однакового розміру природно вважати саме сторінку одиницею обміну із зовнішньою пам'яттю. По-друге, бажано забезпечити таку пошукову структуру у зовнішній пам'яті, з використанням якої пошук інформації за будь-яким ключем вимагає заздалегідь відомого числа обмінів із зовнішньою пам'яттю [4, 5].

Для створення індексів у сучасних базах даних використовуються В+-дерева, але під час роботи з ними виникають проблеми з компактністю, адже вузли у них заповнені не менше ніж на 1/2. Потрібно вирішувати проблему з компактнішим заповненням та відповідно з меншою кількістю рівнів дерева [6, 7].

Опис нового дерева та його порівняння з В+-деревом

Створимо новий тип дерева та назвемо їх К-дерево. Воно містить всі характеристики аналогічно В+-дереву та відрізнялося лише стратегією розщеплення та об'єднання вузлів.

Опишемо властивості К-дерева порівняно з В+-деревом, як зображено в табл. 1, для кращого розуміння побудови пошуку та механізмів зчеплення та розчеплення вузлів.

Таблиця 1

Властивості дерев пошуку, де n – це параметр дерева, що набуває значення не менше 2, а переважно від 50 до 2000

Властивість	В+-дерево	К-дерево
Кількість елементів, що містять корінь дерева	від 1 до $2n$ елементів	від 1 до $3n$ елементів
Кількість елементів, що містять вузли дерева (крім кореня)	від n до $2n$ елементів	від $3/2n$ до $2n$ елементів
Кількість нащадків у вузлів (крім листових), які містять m елементів	$m+1$ нащадків	$m+1$ нащадків
Листові сторінки дерева знаходяться	на одному рівні	на одному рівні
Розчеплення вузлів відбувається шляхом	розбиття вузла навпіл на два нові вузли	розбиття вузла з двома його сусідами на чотири нових вузла
Зчеплення вузлів відбувається шляхом	з'єднання двох вузлів в один новий вузол	з'єднання трьох вузлів у два нові вузли

Отже, відмінність між В+-деревом та К-деревом полягає в відмінностях пунктів 1, 2, 5 та 6 характерних властивостей. В К-дереві, відповідно до першого пункту характерних особливостей, більша кількість елементів в корені вузла, що потрібно для проведення операцій з'єднання та роз'єднання вузлів в К-дереві. А за другим пунктом різниця в мінімальному ступені наповненості вузла: в К-дереві, на відміну від В+-дерева, вона становить $3/4$ вузла. Це дає змогу економити місце на жорсткому диску та збільшити швидкість доступу до інформації за рахунок того, що зчитування вузлів К-дерева, як і В+-дерева, відбувається за блоками, і відповідно від наповненості блока залежить кількість інформації, яку ми отримаємо.

А тепер опишемо детальніше операції додавання і видалення елементів в К-дереві. Як видно з таблиці порівняння, в кожному вузлі К-дерева можуть зберігатися $2n$ ключів, а в корені їх може бути $3n$. Перед тим, як під час вставки ми спустимося до нащадка, ми перевіримо, чи він повний. Якщо це так, то ключі, що знаходяться в нащадку і двох суміжних до нього вузлах, об'єднуються і перерозподіляються. Якщо два суміжні вузли також заповнені, то додається новий вузол. Так, ми отримуємо вже чотири вузли, кожен з яких повний на $3/4$.

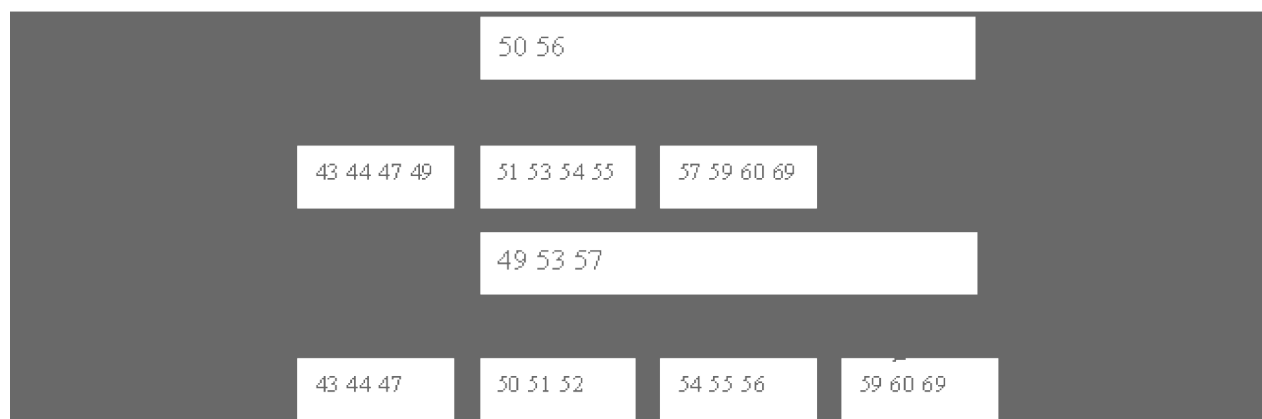


Рис. 1. Процедура розбиття вузлів К-дерева за повністю заповнених вузлів та вставки ключа 52

Перед тим, як під час видалення спуститися до нащадка, ми перевіримо, чи не повний чи він на $1/2$. Якщо це так, ключі нащадка і двох суміжних вузлів об'єднуються і перерозподіляються так, як показано на рис. 1.

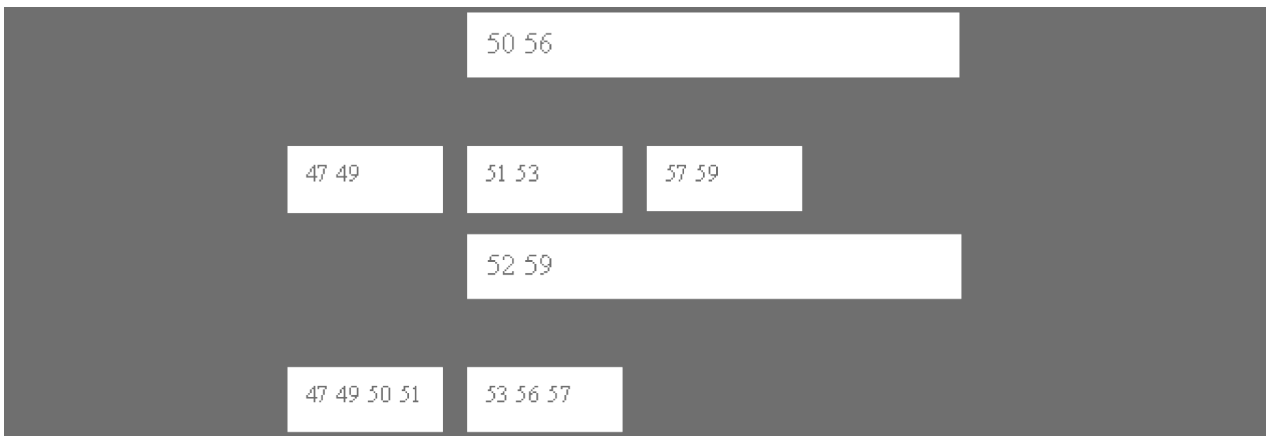


Рис. 2. Процедура з'єднання вузлів К-дерева у разі заповнення вузлів на 1/2 та вставки ключа 52

Якщо два суміжні вузли самі повні наполовину, вони зливаються в два вузли, кожен з яких повний на 3/4 – так, як це показано на рис. 2.

Тобто ми опиняємося посередині між наповненістю на 1/2 і повною наповненістю, що дає нам змогу очікувати однакового числа вставок і вилучень.

У кореневому вузлі зберігаються $3n$ ключів. Якщо під час вставки виявиться, що корінь повний, ми розподіляємо ключі за трьома новими вузлами, кожен з яких повний на 3/4. Це збільшує висоту дерева на одиницю.

Під час видалення ми досліджуємо нащадків. Якщо є тільки три нащадки і вони повні наполовину, переносимо їх вміст в корінь, у результаті чого висота дерева зменшується.

Іншими словами можна сказати, що ми збираємо три вузли, а потім розділяємо їх. Якщо нам потрібен додатковий вузол, ми ділимо на чотири вузли. Якщо вузол потрібно видалити, ми поділяємо на два вузли. Симетрія операцій дає змогу використовувати при реалізації вставки і видалення одні й ті самі об'єднувчі та роз'єднувчі функції під час програмної реалізації К-дерев, що дає змогу економити час на написання програмного коду та його редагування.

Подамо опис К-дерева мовою опису графів (дерев). Позначимо $G = (V, E)$ як множину всіх дерев. Тоді V – це непушта множина всіх вершин дерева, а E – множина пар вершин, що називаються ребрами. Як V , так і E є скінченними множинами. До множини V входять вершини $v_1 \dots v_n$, а до множини E – ребра $e_k = (v_i, v_j)$. Для прискорення пошуку за допомогою дерев потрібно створити матрицю, яка б описувала розміщення блоків з вузлами на жорсткому диску, тобто фактично показувала, яку наступну вершину зчитувати з нього. Для такої побудови використаємо матрицю списку дуг, тобто трирядковий масив, в якому в першому рядку зберігатимуться вершини v_i , з яких починається дуга, в другому рядку – вершини v_j , в яких закінчується дуга, а в третьому рядку – вага дуг. Вагу дуг визначають за номером по порядку та зліва направо.

Наведемо приклад матриці дуг для зберігання вершин дерева, як показано на рис. 3.

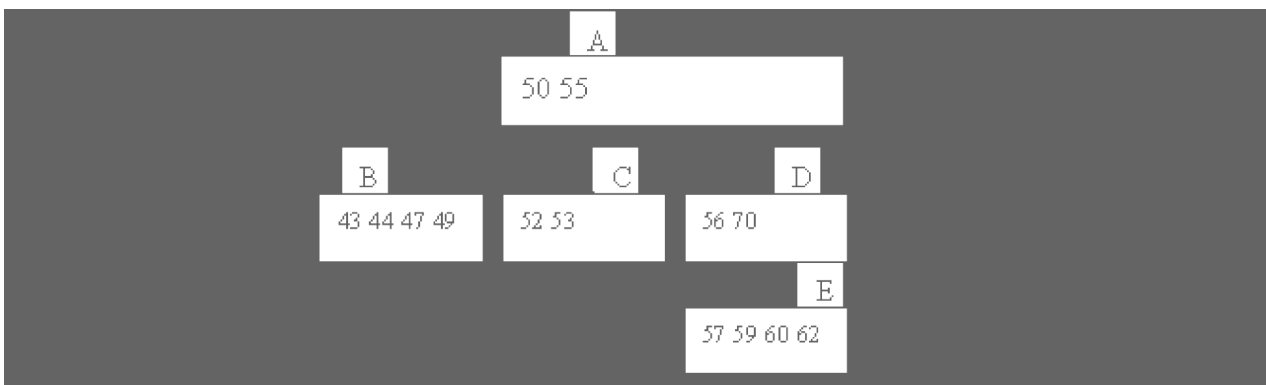


Рис. 3. Приклад К-дерева для опису матрицею дуг

Матриця дуг для зберігання опису розміщення дерев на жорсткому диску

Перша вершина дуги	A	A	A	D
Друга вершина дуги	B	C	D	E
Номер дуги	1	2	3	1

Отже, дуги з вершини А було пронумеровано так, як це потрібно для процесу пошуку, і чітко видно, що якщо ми шукаємо ключ 60, то перевіряємо два числа кореня дерева і йдемо по третій дузі з вершини А вправо донизу до вершини D, а звідти, своєю чергою, до вершини E, де і знаходимо цей ключ.

За цим методом можна пришвидшити знаходження вершин дерев та відповідно пошук загалом.

Аналітичне оцінювання параметрів пошуку на основі К-дерев

Для порівняння швидкості пошуку на основі В+-дерев та К-дерев визначимо кількість елементів, якими можна заповнити ніжній рівень дерева за максимального, середнього та мінімального заповнення. Якщо параметр n дорівнює 50, k – кількість елементів (записів) в дереві, m – рівень (висота) дерева, то формула розрахунку кількості елементів в дереві для відповідного рівня матиме вигляд:

– для В+-дерев

$$k_{B\max} = (2n+1)^m 2n, \quad (1)$$

$$k_{B\text{ave}} = (n+1)(3/2n+1)^{m-1} 3/2n, \quad (2)$$

$$k_{B\min} = 2(n+1)^{m-1} n, \quad (3)$$

– для К-дерев

$$k_{K\max} = (3n+1)(2n+1)^{m-1} 2n, \quad (4)$$

$$k_{K\text{ave}} = (3/2n+1)(7/4n+1)^{m-1} 7/4n, \quad (5)$$

$$k_{K\min} = 2(3/2n+1)^{m-1} 3/2n, \quad (6)$$

та формула знаходження висоти дерева при відомій кількості елементів k матиме вигляд:

– для В+-дерева

$$m_{B\max} = \log_{(2n+1)}(k/2n), \quad (7)$$

$$m_{B\text{ave}} = \log_{(3/2n+1)}(k/(3/2n(n+1))) + 1, \quad (8)$$

$$m_{B\min} = \log_{(n+1)}(k/2n) + 1, \quad (9)$$

– для К-дерева

$$m_{K\max} = \log_{(2n+1)}(k/2n(3n+1)) + 1, \quad (10)$$

$$m_{K\text{ave}} = \log_{(7/4n+1)}(k/(7/4n(3/2n+1))) + 1, \quad (11)$$

$$m_{K\min} = \log_{(3/2n+1)}(k/3n) + 1. \quad (12)$$

Основною умовою знаходження висоти дерева є те, що m завжди ціле число, тому його завжди потрібно округляти до цілого, причому завжди у більший бік.

Після визначення параметрів m для дерев визначимо час пошуку за індексом. Позначимо час пошуку за В+-деревом – $T_{ПВ}$ та відповідно за К-деревом – $T_{ПК}$. Час пошуку ділимо на дві частини: час T_d , за який відбувається пошук та зчитування відповідного вузла дерева на жорсткому диску, зчитування його в оперативну пам'ять та час T_B пошуку у цьому вузлі.

Для В+-дерев та К-дерев час зчитування вузла та час пошуку в вузлі є однаковими; відрізнитися може тільки пошук у корені К-дерева, але ця відмінність настільки мала, що нею можна знехтувати.

Отже, час пошуку даних за допомогою дерев визначають за формулами

$$T_{ПВ} = m_B(T_d + T_B), \quad (13)$$

$$T_{ПК} = m_K(T_d + T_B). \quad (14)$$

Відповідно до цих формул відношення часу пошуку даних за допомогою В+-дерев до часу пошуку даних за допомогою К-дерев відрізнитиметься за значенням параметра m . Позначимо коефіцієнт відмінності через s , а формула його знаходження виглядатиме так:

$$\frac{T_{ПВ}}{T_{ПК}} = \frac{m_B}{m_K} = s. \quad (15)$$

Знаючи коефіцієнт відмінності s ми можемо визначити, у скільки разів пошук за К-деревими буде швидший від пошуку за В+-деревими та наскільки зміниться час.

Розрахуємо конкретний час T_{Π} пошуку для індексів на основі дерев для таблиць з різною кількістю записів в них.

Таблиця 3

Час пошуку в мс для різного числа записів та середньої наповненості дерева

Число записів	V+-дерева	K-дерева
5000	24	12
10000	24	24
50000	24	24
100000	24	24
500000	36	24
1000000	36	36
5000000	36	36
10000000	36	36
50000000	48	36

Вважаємо, що час T_d , за який відбувається пошук та зчитування відповідного вузла дерева на жорсткому диску, дорівнює 10 мс, а час T_v пошуку у цьому вузлі – 2 мс, підставимо його до формул (13), (14) для розрахунку та запишемо отримані дані до табл. 3 та побудуємо графік.

На рис. 4 зображено графік залежності часу пошуку від числа записів у таблиці, за якою проводиться пошук та відповідно по осі x проставлено кількість записів в таблиці, за якими проводився пошук, а по осі y – час пошуку в мс.

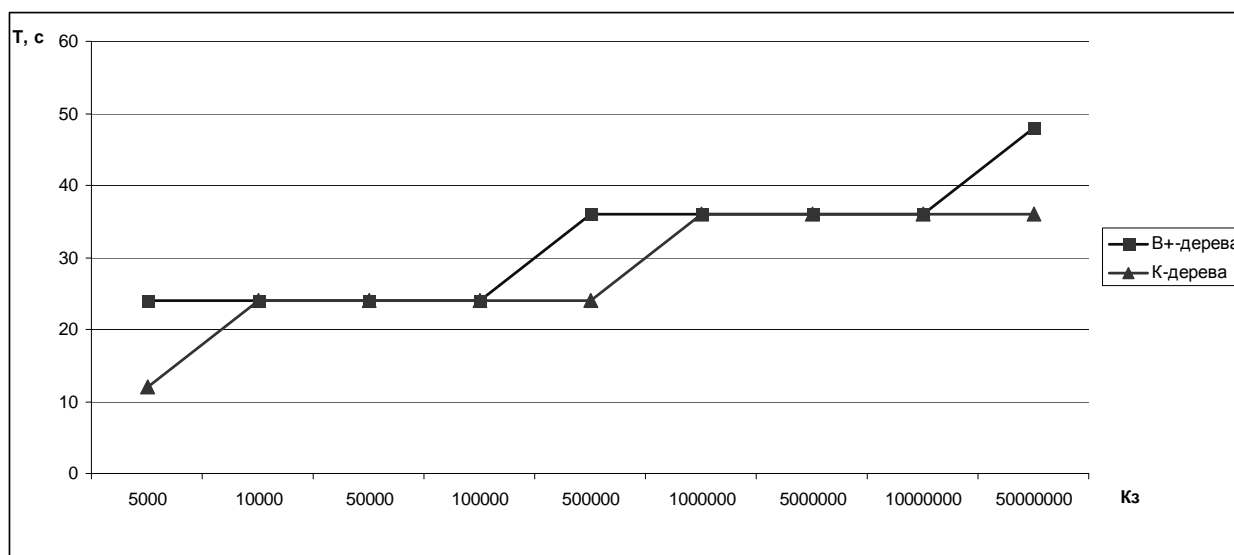


Рис. 4. Залежність часу пошуку від числа записів для різних методів

Висновок

Отже, на невеликій кількості записів K-дерева зменшують час пошуку на відповідних проміжках, яке становить в середньому 9 %, але зі збільшенням кількості записів час пошуку за допомогою індексів, основаних на K-деревях, значно зменшуватиметься порівняно з V+-деревями, що прискорить роботу самої бази даних та дасть змогу витратити ресурси серверів, на яких розміщені БД, з більшою ефективністю.

1. Реймонд Ф. Базы данных. Проектирование и разработка / Ф. Реймонд, Д. Джон, В.С. Крейг. – М.: ИТ Пресс, 2007. – 592 с. 2. Реймонд Ф. Проектирование и разработка баз данных. Визуальный подход / Ф. Реймонд, Д. Джон, В.С. Крейг. – М.: ИТ Пресс, 2007. – 592 с. 3. Ролланд Ф. Основные концепции баз данных / Ф. Ролланд. – М.: Вильямс, 2008. – 256 с. 4. Мишра С. Секреты Oracle SQL / С. Мишра, А. Бьюли. – М.: Симбо, 2009. – 368 с. 5. Бьюли А. Изучаем SQL / А. Бьюли. – М.: Символ-Плюс, 2007. – 312 с. 6. Макдональд К. Oracle PL/SQL для профессионалов. Практические решения / К. Макдональд, Х. Кау. – М.: ДиаСофтЮп, 2005. – 560 с. 7. Шварц Б. Оптимизация производительности баз данных / Б. Шварц, П. Зайцев, В. Ткаченко. – М.: Символ-Плюс, 2010. – 832 с.

І. Цмоць, О. Скорохода, Б. Андрієцький, Р. Ткаченко
Національний університет “Львівська політехніка”,
кафедра автоматизованих систем управління

АПАРАТНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНИХ КОМПОНЕНТІВ РОБОТОТЕХНІЧНИХ СИСТЕМ

© Цмоць І., Скорохода О., Андрієцький Б., Ткаченко Р., 2012

Проаналізовано нейромереві методи реалізації інтелектуальних компонентів, вибрано нейромереві на базі геометричних перетворень, виділено базові операції, розроблено на основі нейроподібної структури інтелектуальний компонент робототехнічної системи для збільшення точності давачів відстані.

Ключові слова: інтелектуальні компоненти, нейроподібні структури, модель геометричних перетворень.

Neural networks methods for intellectual components realization have been analyzed, neural networks based on geometric transformations have been selected, base operations have been determined, the intellectual component of robotic systems for accuracy-increasing of distance sensors based on the neurolike structure has been developed.

Key words: intelligent components, neurolike structures, model of geometric transformation.

Постановка задачі

Сучасний етап розвитку мобільних робототехнічних систем орієнтований на широке використання нейромеревих засобів для оцінювання даних, які надходять із давачів в умовах завад і неповної інформації, компресування та розпізнавання зображень і сцен. Зокрема для підвищення надійності та точності автономного управління робототехнічною системою необхідно підвищити точність вимірювання відстані давачами (ультразвуковими, інфрачервоними, лазерними тощо). Підсистема технічного зору здійснює попереднє опрацювання, компресування та розпізнавання зображень і сцен, що використовується для прийняття рішень у процесі управління робототехнічною системою.

Нейромереві засоби робототехнічних систем створюються на основі паралельних розподілених адаптивних технологій обробки даних у реальному часі, які здатні “вчитися” опрацювати дані, діючи в інформаційному середовищі. Особливістю нейромеревих засобів, які використовуються в робототехнічних системах, є вміння випрацювати правила та модифікувати їх в процесі функціонування таких систем. У робототехнічних системах опрацюють у реальному часі різні за інтенсивністю надходження потоки даних на апаратних засобах, що задовольняють обмеження щодо габаритів, енергоспоживання, вартості та часу розробки. Створення таких засобів вимагає широкого використання сучасної елементної бази (напівзамовних і замовних НВІС) та розроблення нових методів, алгоритмів і НВІС-структур для опрацювання та розпізнавання зображень на основі штучних нейронних мереж (ШНМ). Високої продуктивності і ефективності використання обладнання у нейромеревих засобах досягається розпаралеленням і конвеєризацією процесів обробки, апаратним відображенням структури алгоритму розв'язання задачі у архітектуру, яка адаптована до інтенсивності надходження потоків даних.

Розроблення інтелектуальних компонентів робототехнічних систем з високою ефективністю використання обладнання вимагає проблемно-орієнтованого підходу з поєднанням універсальних і спеціалізованих засобів. При цьому розроблення інтелектуальних компонентів робототехнічних систем із заданими технічними параметрами повинна зводитися до доповнення обчислювального ядра додатковими спеціалізованими модулями.

Тому *метою роботи* є аналіз відомих нейромережових методів для розроблення інтелектуальних компонентів робототехнічних систем, вибір типу нейромережі для їхньої апаратної реалізації та розроблення інтелектуального компоненту для збільшення точності давачів відстані.

Виклад основного матеріалу

Аналіз нейромережових методів для реалізації інтелектуальних компонентів робототехнічних систем. Інтелектуальні компоненти робототехнічних систем доцільно реалізовувати на основі нейронних мереж, оскільки це дає змогу здійснювати самонавчання та адаптацію даних компонентів до змінних умов зовнішнього середовища. Аналіз класичних структур нейромереж (багатошарового перцептрона, мережі Кохонена, RBF мережі та мережі Хопфілда), методів і алгоритмів їхнього навчання та функціонування, які сьогодні широко використовуються для реалізації інтелектуальних компонентів робототехнічних систем, показав, що вони мають такі недоліки [1–6]:

- значний час, необхідний для процесу навчання нейромереж;
- різні алгоритми, які використовуються для навчання та функціонування нейромереж;
- методи та нейромережні алгоритми для реалізації інтелектуальних компонентів робототехнічних систем, не орієнтовані на НВІС-реалізацію;
- нейромережові методи та алгоритми опрацювання в режимі реального часу інтенсивних потоків даних вимагають для своєї реалізації великих затрат обладнання;
- не враховують вимоги конкретних застосувань щодо габаритів і споживаної потужності.

З аналізу видно, що для забезпечення обробки потоків даних у реальному часі та одночасного зменшення габаритів і споживаної потужності необхідно використовувати спеціалізацію, конвеєризацію, просторовий паралелізм і орієнтацію архітектури нейромережних засобів на НВІС-реалізацію.

Альтернативою класичним нейромережам є нейроподібні структури (НС) на базі геометричних перетворень (ГП), які відомі під назвою функціонал на множині табличних функцій [7, 8]. Особливістю НС на базі ГП є однотипність алгоритмів навчання та функціонування. Топологія алгоритму навчання НС на базі ГП представляється у вигляді функціонального графу, вершини якого відповідають скалярному добутку вектора вхідних сигналів на вектор вагових коефіцієнтів та нелінійному перетворенню. Використання методів і алгоритмів НС на базі ГП забезпечує значне підвищення продуктивності як для програмного, так і для апаратного варіантів реалізації інтелектуальних компонентів робототехнічних систем.

Використання НС на базі ГП усуває, або зменшує негативні властивості існуючих засобів інформаційного моделювання – регресійних моделей, машин опорних векторів, класичних штучних нейронних мереж, індуктивних моделей, контролерів нечіткої логіки, статистичних процедур.

Порівняно із класичними архітектурами штучних нейронних мереж НС на базі ГП мають такі переваги:

- єдина методологічна основа побудови архітектури для різних завдань та предметних областей застосування;
- швидке неітераційне навчання, що відкриває можливість розв'язування завдань великих розмірностей;
- повна повторюваність результатів навчання;
- можливість отримання задовільних розв'язків для тренувальних вибірок зменшеного об'єму;
- розширення кібернетичного принципу “чорної скриньки” на користь “сірої скриньки”, оскільки НС на базі ГП володіють додатковими можливостями аналізу внутрішньої структури даних;
- висока точність та покращені генералізуючі властивості.

Незважаючи на помітні переваги НС на базі ГП за продуктивністю перед традиційними нейромережними засобами, найвищі параметри швидкодії для них досягаються у випадку апаратної реалізації.

Виділення базових операцій для синтезу апаратних нейроподібних структур для реалізації інтелектуальних компонентів робототехнічних систем. Аналіз алгоритмів навчання та функціонування НС на базі ГП показав, що вони є добре структурованими і реалізуються на основі однотипних операцій.

Робота НС на базі ГП передбачає етап попередньої підготовки даних та етап навчання та функціонування. Етап попередньої підготовки полягає у нормалізації вектора вхідних даних, який використовує такі операції: визначення максимального числа з групи чисел, обчислення кореня квадратного з суми квадратів різниць і ділення. Етап навчання та функціонування використовує такі базові операції: групового підсумовування, множення, обчислення скалярного добутку, обчислення передатної функції.

Алгоритми реалізації перерахованих операцій повинні бути орієнтовані на НВІС-реалізацію, враховувати вартість площі кристала, а також кількість вхідних і вихідних виводів. Число зовнішніх виводів НВІС-пристроїв обмежене рівнем технології та розміром кристала. В основу НВІС-реалізації даних алгоритмів необхідно покласти такі принципи:

- узгодженості інтенсивності надходження даних з обчислювальною інтенсивністю пристрою;
- конвеєризації та просторового паралелізму;
- регулярності, модульності та широкого використання стандартних елементів;
- локалізації та зменшення кількості зв'язків між елементами пристрою.

Формування вимог до інтелектуальних компонентів робототехнічних систем. Основні вимоги, які висувуються до інтелектуальних компонентів робототехнічних систем, – це мінімізація габаритів, енергоспоживання, забезпечення надійності, гнучкості та роботи у реальному часі. Створення таких інтелектуальних компонентів робототехнічних систем вимагає широкого використання сучасних компонентів (відеокамер, давачів, платформи-шасі, комп'ютерних систем), сучасної елементної бази для реалізації апаратних засобів комп'ютерних систем обробки та управління (напівзамовних і замовних НВІС, процесорів цифрової обробки сигналів, мікроконтролерів, нейрочипів) та розроблення нових методів і алгоритмів для опрацювання у реальному часі різних за інтенсивністю надходження потоків даних (відеопотоків, даних з давачів, команд управління).

Режим реального часу накладає обмеження на час розв'язання задачі T_p , який не повинен перевищувати часу обміну повідомленнями $T_{обм}$, тобто:

$$T_p \leq T_{обм}.$$

Час обміну залежить як від обсягу N , розрядності n і частоти F_d надходження вхідних даних, так і від кількості k каналів та їх розрядності n_k . Такий час визначається за формулою:

$$T_{обм} = \frac{Nn}{F_d k n_k}.$$

Для забезпечення опрацювання потоків даних в реальному часі за допомогою інтелектуальних компонентів робототехнічних систем їх продуктивність повинна становити:

$$\Pi \geq \frac{\beta R F_d k n_k}{Nn},$$

де R – складність алгоритмів розв'язання задач; β – коефіцієнт врахування особливостей засобів реалізації алгоритму.

Для підвищення ефективності використання обладнання в інтелектуальних компонентах робототехнічних систем необхідно узгодити інтенсивність надходження даних з обчислювальною здатністю апаратних засобів. Інтенсивність надходження даних залежить від кількості та розрядності каналів надходження даних і частоти надходження даних [9]:

$$P_D = knF_d,$$

де k – кількість каналів надходження даних; n – розрядність каналів надходження даних; F_d – частота надходження даних. Обчислювальну здатність апаратних засобів визначають так [9]:

$$D_k = \frac{m_m n_m}{T_k},$$

де m_m – кількість каналів надходження даних у сходинках конвеєра; n_m – розрядність каналів надходження даних у сходинках конвеєра; T_k – такт опрацювання даних.

Для вибору інтелектуальних компонентів робототехнічних систем доцільно використовувати критерій ефективності використання обладнання E , який зв'язує продуктивність з витратами обладнання та дає оцінку апаратними засобам за продуктивністю [9]. Кількісно величину ефективності використання обладнання для інтелектуальних компонентів робототехнічних систем визначають так:

$$E = \frac{R}{T_p W},$$

де W – витрати обладнання на реалізацію інтелектуальних компонентів робототехнічних систем.

Крім перерахованих вимог, інтелектуальні компоненти робототехнічних систем повинні забезпечувати:

- мобільність і автономність, достатню для виконання поставлених задач;
- використання сучасних інтерфейсів передавання даних і команд;
- інтерактивний режим управління робототехнічною системою.

Вибір елементної бази для реалізації інтелектуальних компонентів робототехнічних систем.

Вибирати елементну базу для реалізації інтелектуальних компонентів робототехнічних систем доцільно, порівнюючи за їх параметром ефективності, який враховує продуктивність, габарити, потужність споживання та ціну. Цей параметр визначається так:

$$E_i = n_i t_i g_i k_i p_i s_i c_i \frac{\Pi_i}{\beta P_i \alpha S_i C_i},$$

де $i=1, \dots, m$; m – кількість варіантів; n_i – пороговий коефіцієнт, який враховує вимоги за продуктивністю; t_i – пороговий коефіцієнт, який враховує вимоги за температурою; g_i – пороговий коефіцієнт, який враховує спеціальні вимоги; p_i – пороговий коефіцієнт, який враховує вимоги за потужністю споживання; s_i – пороговий коефіцієнт, який враховує вимоги габаритів; c_i – пороговий коефіцієнт, який враховує вартість системи; Π_i – продуктивність i -го варіанта системи; P_i – потужність споживання i -го варіанта системи; S_i – габарити i -го варіанта системи, C_i – вартість i -го варіанта системи; β – коефіцієнт врахування потужності споживання; α – коефіцієнт врахування габаритів.

Аналіз показав, що найчастіше для побудови інтелектуальних компонентів робототехнічних систем використовують таку елементну базу:

- нейросигнальні процесори (нейрочипи);
- систолічні процесори;
- процесори загального призначення;
- процесори цифрової обробки сигналів (ПЦОС);
- програмовані логічні інтегральні схеми (ПЛІС);
- замовні НВІС (ASIC);
- напівзамовні НВІС – ПЛІС (FPGA).

Ядро нейросигнальних процесорів являє собою типовий ПЦОС, а реалізована на кристалі додаткова логіка забезпечує виконання нейромережових операцій. Нейрочипи спеціально розроблено для застосування в нейромережах і більшість з них орієнтовані на конкретні спеціалізовані системи.

Систолічні процесори – це матричні процесори, які працюють за конвеєрним принципом, виконуючи ритмічні обчислення і ритмічне передавання даних в системі і об'єднують деяку кількість процесорних елементів, а вся інша логіка реалізована на базі периферійних схем. При цьому можна досягнути паралельності обробки даних високого ступеня. Недоліком цих систем є вузька спеціалізація обробних елементів, що призводить до того, що систолічна матриця повинна бути оточена великою кількістю периферійних схем, що реалізують додаткову логіку, наприклад, модуль пам'яті ваг.

Процесори загального призначення використовуються для реалізації програмних компонентів робототехнічних систем. Для цього розроблено бібліотеки для мов високого рівня, що реалізують різноманітні функції, які використовуються в інтелектуальних компонентах робототехнічних систем; розроблено розширення об'єктно-орієнтованих мов, які дають змогу оперувати з нейромережами на рівні об'єктів та візуальні засоби розробки нейроорієнтованих систем.

ПЦОС найчастіше застосовують для реалізації процесорних ядер інтелектуальних компонентів робототехнічних систем. Особливістю ПЦОС є високий ступінь спеціалізації та мала тривалість командного циклу, конвеєризація на рівні окремих мікроінструкцій та інструкцій, розміщення операндів більшості команд у регістрах, використання тінювих регістрів для збереження стану обчислень, поділ шин на шини команд і даних (гарвардська архітектура). Для ПЦОС характерною є наявність апаратного помножувача, що дає змогу виконувати множення чисел за один командний такт. Іншою особливістю ПЦОС є введення до системи команд таких операцій, як множення з накопиченням та різноманітних бітових операцій. У ПЦОС підтримується множинний доступ до пам'яті за один командний цикл.

Під час реалізації інтелектуальних компонентів робототехнічних систем на основі замовних НВІС забезпечується максимальна продуктивність (завдяки адаптації структури НВІС під конкретні алгоритми роботи) та найменша вартість кристала у разі серійного виробництва. Істотними недоліками такої реалізації є великі фінансові затрати на проектування та тривалий термін запуску кристала у виробництво. Через це для малосерійних партій це рішення є нерентабельним.

Реалізація інтелектуальних компонентів робототехнічних систем на основі ПЛІС має значну кількість переваг. На сучасному етапі на одному кристалі ПЛІС можна побудувати велику кількість помножувачів, які працюватимуть паралельно на доволі високій частоті. Окрім цього, сучасні ПЛІС характеризуються значною кількістю зовнішніх виводів та розвинутою структурою швидкісних між'єднань.

Сучасні НВІС-технології забезпечують можливість реалізації на одному кристалі ПЛІС як обчислювальних компонентів, що реалізують алгоритми роботи інтелектуальних компонентів робототехнічних систем, так і інтерфейсних компонентів, які забезпечують каскадування кристалів між собою та їхній зв'язок із навколишнім середовищем.

Використання технології ПЛІС для реалізації інтелектуальних компонентів робототехнічних систем забезпечує можливість реконфігурації структури кристала, яку використовують для адаптації до алгоритмів розв'язання конкретних задач. Можливість реконфігурації розширює спектр задач, які можна розв'язати на одних і тих самих апаратних засобах.

Реалізація інтелектуального компонента для збільшення точності давачів відстані здійснюється з використанням сингулярного спектрального аналізу. Сингулярний спектральний аналіз ґрунтується на дослідженні часового ряду методом головних компонент і не вимагає попередньої стабілізації ряду. Це дає змогу досліджувати структуру часового ряду, виділяти окремі його складові та прогнозувати як сам ряд, так і тенденції розвитку його складових. При сингулярному спектральному аналізі здійснюється розклад вихідної функції, що породжує часовий ряд, в аналог ряду Фур'є за базисом, що породжується самою функцією. У випадку виродженості системи багатовимірних векторів можна стверджувати про існування внутрішніх лінійних зв'язків між значеннями процесу. Однією з найбільших переваг цього методу є відсутність вимоги апріорного знання моделі ряду, що майже не впливає на якість отриманих результатів.

Розробляємо неймережу для підвищення точності вимірювання давачів відстані на основі базової структури НС на базі ГП [7], яка наведена на рис. 1.

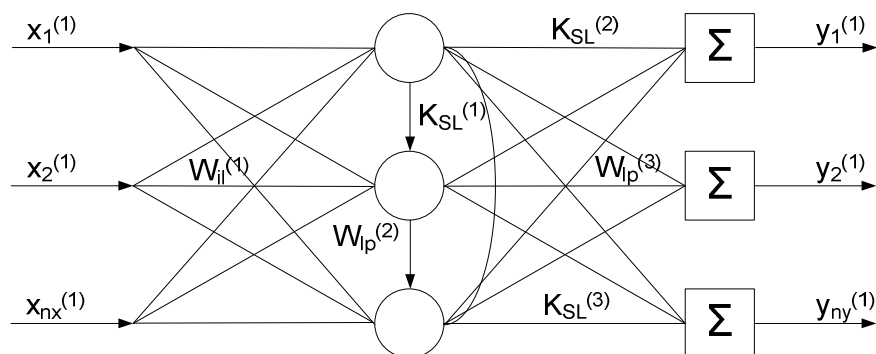


Рис. 1. Базова структура НС на базі ГП з проєктивними та впорядкованими латеральними зв'язками

Процедура адаптації НС на базі ГП для збільшення точності давачів відстані здійснюється за чотири етапи.

На *першому етапі* експериментально визначається розмір вікна L , тобто кількість головних компонентів, на які розкладатиметься сигнал. Розмір вікна вибирають так, щоб досягти найкращих результатів щодо точності.

На *другому етапі* здійснюється програмна реалізація НС на базі ГП з параметрами, визначеними на першому етапі.

На *третьому етапі* шляхом подання на вхід розробленої програмної нейромережі даних з давача, здійснюється розклад вхідного сигналу на L головних компонентів. Значення, отримане з давача, є сумою значень всіх L головних компонентів та певної константи. Цю константу обчислюють під час навчання нейромережі, а значення всіх головних компонентів є відхиленням в більший або менший бік від цієї константи.

Четвертий етап полягає в отриманні вимірних даних з підвищеною точністю. Підвищують точність вимірних даних, відкидаючи з початкового сигналу ті головні компоненти, які є шумом і не несуть корисної інформації. Для цього потрібно від значення початкового сигналу віднімати значення головних компонент, починаючи з останнього, і дивитися, за якого значення похибка вимірних даних відносно дійсного значення є мінімальною.

За допомогою використання нейромережевого сингулярного спектрального аналізу нам вдалося зменшити похибку вимірювання відстаней, отриманих з ультразвукового давача, з 1,96 % до 0,79 %.

Висновки

1. Використання нейронних мереж для реалізації інтелектуальних компонентів робототехнічних систем забезпечує адаптацію даних компонентів до змінних умов зовнішнього середовища шляхом самонавчання.

2. Аналіз існуючих типів нейромереж показав, що розробляти інтелектуальні компоненти робототехнічних систем доцільно на основі нейроподібних структур на базі геометричних перетворень, оскільки вони забезпечують реалізацію швидких, неітеративних обчислень без накопичення похибок і не мають обмежень на вимірність.

3. Базовими операціями для реалізації інтелектуальних компонентів робототехнічних систем на основі нейроподібних структур на базі геометричних перетворень є: визначення максимального числа з групи чисел, обчислення кореня квадратного з суми квадратів різниць, ділення, групове підсумовування, множення, обчислення скалярного добутку та обчислення передатної функції.

4. Реалізація інтелектуальних компонентів робототехнічних систем на базі ПЛІС забезпечує високу швидкодію та можливість зміни алгоритму функціонування.

1. Круглов В.В., Борисов В.В. *Искусственные нейронные сети. Теория и практика.* – 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002. – 382 с. 2. Ясницкий Л.Н. *Введение в искусственный интеллект.* – М.: ИЦ «Академия», 2005. – 176 с. 3. Шальто А.А. *Методы аппаратной и программной реализации алгоритмов.* – СПб.: Наука, 2005. – 780 с. 4. Хайкин С. *Нейронные сети: полный курс, 2-е изд.: Пер. с англ.* – М.: "Вильямс", 2006. 5. Осовский С. *Нейронные сети для обработки информации / Пер. с польск.* – М.: Финансы и статистика, 2002. – 344 с. 6. Рутковская Д., Пилиньский Л., Рутковский Л. *Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск.* – М.: Горячая линия-Телеком, 2007. – 452 с. 7. Ткаченко Р.О. *Моделирование методами нейронных сетей: навч.-метод. посібник / Р.О. Ткаченко, П.Р. Ткаченко, Н.О. Мельник; ЛІБС УБС НБУ.* – Львів, 2010. – 114 с. 8. Ткаченко Р.О., Юрчак І.Ю., Цимбал Ю.В. *Неітеративне навчання нейронних мереж прямого поширення // Вісник Держ. ун-ту "Львівська політехніка": Комп'ютерна інженерія та інформаційні технології.* – 1999. – № 380. – С. 109–115. 9. Цмоць І.Г. *Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі.* – Львів: УАД, 2005. – 227с.

О. Леонтьєва*, К. Обельовська

*TeleConsult-Austria GmbH, Schwarzbauerweg, 3, A-8043, Graz, Austria
Національний університет “Львівська політехніка”,
кафедра автоматизованих систем управління

ПОРІВНЯННЯ ПРОПУСКНИХ ЗДАТНОСТЕЙ СХЕМ ДОСТУПУ DCF ТА EDCA БЕЗПРОВІДНИХ ЛОКАЛЬНИХ МЕРЕЖ

© Леонтьєва О., Обельовська К., 2012

Проведено порівняльний аналіз пропускної здатності двох схем доступу безпроводних мереж DCF та EDCA в умовах насиченої завантаженості.

Ключові слова: безпроводні мережі, DCF, EDCA, пропускна здатність.

This paper is devoted to the comparison analysis of wireless networks access schemes DCF and EDCA in case of saturation condition.

Key words: wireless networks, DCF, EDCA, throughput.

Вступ

Стандарт IEEE 802.11 [1] є найпопулярнішим стандартом безпроводних локальних мереж, а отже роботи щодо його аналізу, вдосконалення та розширення є актуальними. Значно впливають на ефективність мереж, що відповідають даному стандарту, процеси, які відбуваються на MAC-підрівні. Одним з основних завдань MAC-підрівня є організація доступу до спільного фізичного середовища.

За стандартом IEEE 802.11g доступ до спільного фізичного середовища забезпечується базовою схемою розподіленого управління DCF (Distributed Coordination Function), яка ґрунтується на методі множинного доступу з прослуховуванням несучої та запобіганням колізій CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Додатково в мережах з точкою доступу стандарт передбачає використання схеми централізованої координації PCF (Point Coordination Function), яка дає змогу централізовано управляти колективним доступом до фізичного середовища. Метод DCF є головним та обов'язковим, PCF – додатковим. Розподілений режим доцільно використовувати в мережах, трафік яких є переважно пакетним.

З метою забезпечення мультимедійних ужитків в стандарті IEEE 802.11n введено гібридну схему управління HCF (Hybrid Coordination Function), яка, крім вільного від конкуренції доступу (HCF Controlled Channel Access – HCCA), передбачає також покращений розподілений доступ до каналу (Enhanced Distributed Channel Access – EDCA).

Порівняно пропускні спроможності схем доступу DCF та EDCA.

Постановка задачі

У розподіленому режимі DCF реалізується множинний доступ з контролем несучої та запобіганням колізій CSMA/CA. Станція, яка хоче передати кадр, перед початком передавання прослуховує середовище. Оскільки режим доступу DCF вимагає синхронізації станцій, то часові інтервали починають відраховуватись з моменту закінчення передавання чергового кадру. Станція, що прослуховує середовище, зафіксувавши закінчення передачі кадру в середовищі, відраховує час, який дорівнює міжкадровому інтервалу IFS (InterFrame Slot). Якщо після закінчення міжкадрового інтервалу IFS середовище залишається вільним, починається відрахунок слотів фіксованої довжини. Кадр можна починати передавати тільки на початку якогось слоту за умови, що середовище вільне. Номер слоту, в якому можна почати передавати, вибирається як випадкове ціле число, рівномірно розподілене на інтервалі $[0, CW]$, де CW (Contention Window) – конкурентне

вікно. Так, схема DCF забезпечує станціям конкурентний спосіб доступу до спільного фізичного середовища, зменшуючи при цьому продуктивність мережі. Слід зазначити, що приведений опис є спрощеним, а сама схема DCF не є орієнтованою на пріоритетне обслуговування, тобто пакети різних класів трафіку вона обслуговує однаково.

На відміну від базової схеми DCF, схема EDCA, що є її розширенням, призначена для підтримки якості обслуговування QoS (Quality of Services). Вона передбачає чотири категорії доступу AC (Access Categories) [1]:

- AC3 – для голосу;
- AC2 – для відео;
- AC1 – для передавання даних у міру можливості (звичайного трафіку);
- AC0 – для передавання у фоновому режимі.

Кожна категорія доступу відповідає певному пріоритету і характеризується власною множиною параметрів доступу до фізичного середовища. Для кожної категорії доступу існує своя черга передачі кадрів, визначено свої арбітражні міжкадрові проміжки AIFS (Arbitration Interframe Space) різної тривалості, крім того, змінюється розмір вікна конкуренції CW. Кожна з черг опрацьовується незалежно одна від одної. Якщо для двох (або більше) кадрів з різних черг однієї станції проміжки відтермінування одночасно дорівнюють 0, то внутрішня колізія вирішується наданням права на передавання кадру з найвищим пріоритетом – решта кадрів опрацьовується як при виникненні зовнішньої колізії. Тобто трафік з вищим пріоритетом має більше шансів бути відісланим, ніж трафік з нижчим пріоритетом.

У цій роботі необхідно проаналізувати, як впливає на пропускну здатність мережі заміна схеми DCF, що не підтримує QoS, на схему EDCA, яка QoS підтримує.

Вибір моделей та порівняння пропускних здатностей схем доступу DCF та EDCA

У роботі [2] наведено огляд аналітичних моделей для дослідження MAC-підрівня безпроводних мереж. Обґрунтовано розгляд при моделюванні насиченого режиму, тобто режиму, який передбачає, що в черзі станції завжди є кадри для передачі.

Для порівняння пропускних здатностей схем доступу DCF та EDCA були обрані дві аналітичні моделі, а саме: моделі Ксяо [3] та Жанга [4] (модель названо за прізвищем першого автора), які найповніше описують роботу безпроводної мережі при базовій схемі доступу DCF та пріоритетній схемі доступу EDCA.

Для обох моделей було прийнято такі припущення:

- 1) ідеальний стан фізичного каналу;
- 2) відсутність прихованих станцій в мережі, тобто усі станції знаходяться у зоні прослуховування;
- 3) число станцій в мережі та число класів доступу є фіксованими;
- 4) при моделюванні насиченого режиму в чергах станції завжди є кадри для передавання.

Весь трафік в обох моделях поділяється на N класів доступу: $i = 1, \dots, N$. Для кожного класу доступу i визначено значення мінімального вікна конкуренції, максимального числа відтермінувань, числа спроб передачі кадру та арбітражного міжкадрового інтервалу AIFS_{*i*}. Для опису алгоритму функціонування кожної станції в моделях Ксяо та Жанга було побудовано двонапрявлену марківську модель станів.

Нормована пропускна здатність каналу для i -го класу трафіку S_i визначається як частка часу зайнятості каналу успішною передачею корисного трафіку i -го класу:

$$S_i = \frac{E(\text{тривалість передачі корисної інформації класу } i)}{E(\text{довжина слоту часу})} = \frac{p_{s,i} T_{E(P_i)}}{(1 - p_b) T_e + p_s T_{s,i} + p_c T_{c,i}}, \quad (1)$$

де $p_{s,i}$ – ймовірність успішного передавання кадру класу i у певний слот часу, p_b – ймовірність зайнятості каналу при прослуховуванні, p_s – ймовірність успішного передавання кадру у певний слот часу, p_c – ймовірність потрапляння кадру у колізію у певний слот часу, $T_{E(P_i)}$ – тривалість передавання корисної інформації кадру класу i , T_e – тривалість порожнього слоту часу, $T_{s,i}$ – середня тривалість успішної передачі кадру класу i , $T_{c,i}$ – середня тривалість колізії кадру i .

Ймовірності успішної передачі кадру або його потрапляння в колізію знаходяться за допомогою побудованої марківської моделі станів та з урахуванням прийнятих авторами припущень [3, 4].

Середній час доставки $T_{s,i}$ та тривалість колізії $T_{c,i}$ можна обчислити, знаючи параметри MAC-підрівня та фізичного рівня – тривалість передачі заголовку кадру T_H , тіла кадру $T_{E(P_i)}$, короткого міжкадрового проміжку SIFS, квитанції підтвердження T_{ACK} :

$$T_{s,i} = T_H + T_{E(P_i)} + SIFS + AIFS_i + T_{ACK} + 2\delta, \quad (2)$$

$$T_{c,i} = T_H + T_{E(P_i)} + SIFS + AIFS_i + \delta, \quad (3)$$

де δ – затримка поширення сигналу. Оскільки в аналітичних моделях складно врахувати залежність часових характеристик від відстані між станціями, то для обох моделей величину затримки поширення сигналу було прийнято як сталу величину, що дорівнює 1 мкс.

При порівнянні пропускних здатностей схем доступу DCF та EDCA безпроводних мереж для обох моделей було прийнято однакові параметри MAC-підрівня та фізичного рівня стандарту 802.11g [1] (табл. 1).

Таблиця 1

Параметри MAC-підрівня та фізичного рівня стандарту 802.11g

Розмір кадру	2312 байт
MAC-заголовок	34 байта
PHY-заголовок	42 байта
ACK	14 байт
Розмір слоту	20 мкс
SIFS	10 мкс
DIFS	50 мкс
CW _{min}	31
CW _{max}	1023
Число повторних передач кадру	16
Швидкість передачі	54 Мбіт/с

При моделюванні роботи безпроводної мережі згідно із схемою доступу EDCA трафік рівнопропорційно поділявся на чотири класи з різними пріоритетами. Для кожної черги класу доступу визначено власні параметри MAC-підрівня [1] (табл. 2).

Таблиця 2

Параметри MAC-підрівня для категорій доступу

	AC0	AC1	AC2	AC3
CWmin	31	31	15	7
CWmax	1023	1023	31	15
AIFS (мкс)	150	70	50	50

На рис. 1 наведено графіки залежності нормованої пропускної здатності від кількості станцій мережі у випадку насиченої завантаженості для схеми доступу DCF, обчислені за моделями Ксяо і Жанга та за результатами імітаційного моделювання.

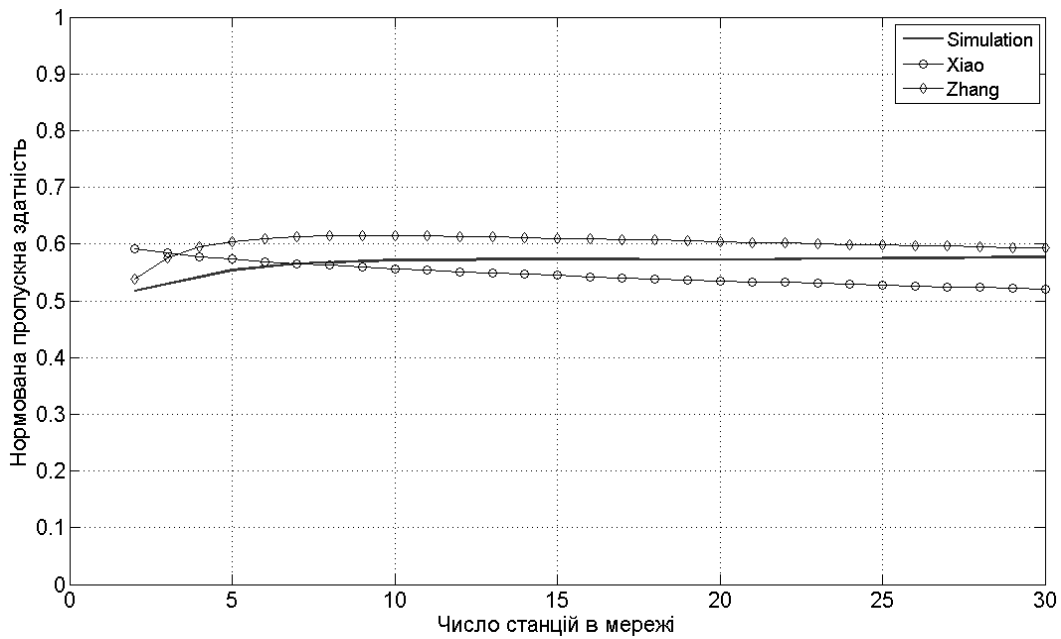


Рис. 1. Нормована пропускна здатність у випадку насиченої завантаженості для схеми доступу DCF

Як видно з рис. 1, нормована пропускна здатність є низькою, що спонукає за необхідності обслуговування додатків критичних до часових параметрів звертатись до схеми доступу EDCA.

На рис. 2 наведено графіки залежності нормованої пропускної здатності від кількості станцій в мережі у випадку насиченої завантаженості для схеми доступу EDCA за окремими класами доступу.

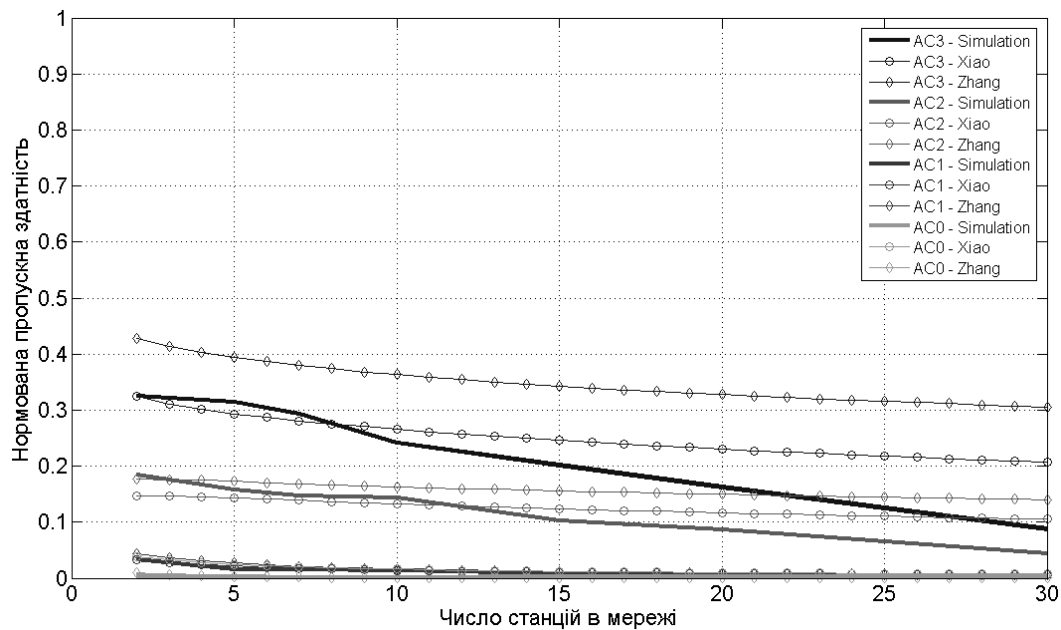


Рис. 2. Нормована пропускна здатність у випадку насиченої завантаженості схеми доступу EDCA для різних класів доступу

Для усіх залежностей можна виділити спадний характер пропускної здатності зі збільшенням кількості станцій у мережі. Для класів доступу AC0 та AC1 нормована пропускна здатність є найнижчою та становить 0.–4 % від швидкості передавання, оскільки кадри класів доступу AC2 та AC3 швидше отримують доступ до середовища та мають вищий пріоритет під час вирішення внутрішньої колізії між кадрами з різних черг.

Рис. 3 ілюструє співвідношення нормованої пропускної здатності схеми доступу DCF та загальної нормованої пропускної здатності схеми доступу EDCA, обчислені за моделями Ксяо і Жанга та за результатами імітаційного моделювання.

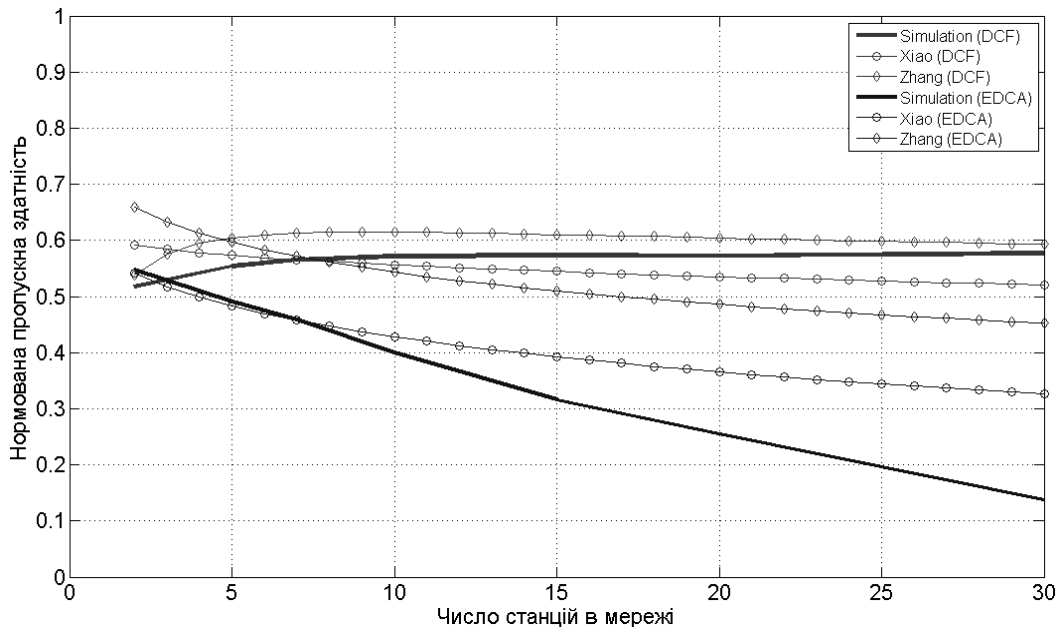


Рис. 3. Загальна нормована пропускна здатність у випадку насиченої завантаженості для схем доступу DCF та EDCA

Як видно з рис. 3, сумарна пропускна здатність безпроводної мережі за схемою доступу EDCA нижча, ніж за схемою доступу DCF. При цьому що більше станцій у безпроводній мережі, то меншу пропускну здатність забезпечує схема доступу EDCA у насиченому режимі порівняно зі схемою доступу DCF. Отже, схема доступу EDCA, забезпечуючи якість обслуговування трафіку високопріоритетних класів доступу, при насиченій завантаженості та великій кількості станцій в мережі менш ефективна загалом, порівняно зі схемою доступу DCF. Оскільки в чергах станцій завжди присутні кадри для передавання, в цьому випадку право на передавання з більшою ймовірністю отримують кадри з вищим класом доступу. Ймовірність передавання кадрів нижчого класу доступу збільшується у випадку, коли внаслідок колізії мінімальне вікно конкуренції для високопріоритетних класів доступу значно більше, ніж мінімальне вікно конкуренції низькопріоритетних класів доступу.

Як один з методів підвищення ефективності роботи безпроводної мережі можна запропонувати метод адаптивного переходу зі схеми доступу EDCA на схему DCF. Критерії та умови застосування методу адаптивного переходу будуть сформовані та досліджені в подальших роботах.

Висновок

За допомогою аналітичного та імітаційного моделювання здійснено порівняльний аналіз пропускних здатностей мереж з децентралізованою схемою доступу DCF та розширеним доступом EDCA. Показано, що забезпечення пріоритетного доступу схемою EDCA для мереж, що мають п'ять і більше станцій, призводить до зменшення загальної пропускної здатності мережі. Запропоновано як один з методів підвищення ефективності роботи безпроводної мережі метод адаптивного переходу зі схеми доступу EDCA на схему DCF. Критерії та умови застосування методу адаптивного переходу будуть сформовані та досліджені в подальших роботах.

1. IEEE Std 802.11™-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std., 2007.
2. Леонт'єва О. Аналіз аналітичних моделей безпроводних мереж стандарту 802.11. Технічні вісті, № 1(31)-2(32). – С. 113–117, 201.
3. Xiao, Y., A simple and effective priority scheme for IEEE 802.11, IEEE Communication Letters, Vol.7(2), pp. 70–72, 2003.
4. Zhang, W., Sun, J., Liu, J., Zhang H.-B., Performance analysis of IEEE 802.11e EDCA in wireless LANs // Journal of Zhejiang University SCIENCE A. – 2007. – 8(1). – P. 18–23.

СПЕКТРАЛЬНІ ВЛАСТИВОСТІ БУЛЕВИХ ФУНКЦІЙ, РЕАЛІЗОВНИХ ОДНИМ НЕЙРОННИМ ЕЛЕМЕНТОМ І СУМАТОРАМИ ЗА МОДУЛЕМ 2

© Гече Ф., Коцовський В., Батюк А., 2012

Розроблено методи синтезу комбінаційних схем із одного нейронного елемента та суматорів за модулем 2. Встановлено ті основні перетворення у спектральній області булевих функцій, за допомогою яких мовою характеристичних векторів можна встановити реалізованість булевих функцій комбінаційною схемою з одного нейронного елемента та суматорів за mod 2.

Ключові слова: ядро булевої функції, матриця толерантності, комбінаційна схема, спектр булевої функції.

We give new algorithms for the synthesis of the combinational circuits including one threshold device and binary adders. We show how we can use the characteristic vectors for checking the realizability of the Boolean function on such circuits.

Key words: the Boolean function kernel, tolerance matrix, combinational circuit, the spectrum of Boolean function.

Вступ

У роботі [1] показано, що булева функція $f(x_1, \dots, x_n)$ реалізується одним нейронним елементом із пороговою функцією активації, якщо в множині її зведених ядер $T(f)$ хоча б для одного зведеного ядра $K(f)_i$ можна вказати такі елементи σ, ξ відповідних симетричних груп S_n, S_q ($q = |K(f)|$) і таку матрицю толерантності $L \in E_n^-$, що $K_\xi^\sigma(f) = L(q)$. Природно виникає питання: чи можна розширити клас функцій алгебри логіки, які реалізуються одним НЕ, якщо над ними виконати деякі перетворення? У цій статті як таке перетворення застосовується логічна операція \oplus — сума за mod 2. Також досліджується питання: які перетворення у спектральній області булевих функцій відповідають логічним операціям, що перетворюють функції алгебри логіки на нейрофункції?

Синтез комбінаційних схем з одного нейронного елемента та суматорів за модулем 2

Нехай $K(f)$ — ядро булевої функції $f(x_1, \dots, x_n)$, елементи якого записані у рядки матриці $K_\xi(f)$ ($\xi \in S_q$). Вектор-стовпчик матриці $K_\xi(f)$ з номером i позначимо через \mathbf{i}_i і визначимо дію операторів A_j, B_j ($j \in \{1, 2, \dots, n\}$) над ядром $K(f)$ так:

1. $A_j(K(f), (i_1, \dots, i_t))$ — означає, що j -й вектор-стовпчик ядра $K(f)$ заміниться вектор-стовпчиком $\mathbf{i}_j \oplus \mathbf{i}_{i_1} \oplus \dots \oplus \mathbf{i}_{i_t}$, де $j, i_1, \dots, i_t \in \{1, 2, \dots, n\}$ і \oplus — покоординатне додавання булевих векторів за mod 2. Оператору $A_j(K(f), (i_1, \dots, i_t))$ присвоїмо індекс $r = 2^{n-i_1} + 2^{n-i_2} + \dots + 2^{n-i_t}$ і множину всіх різних рядків одержаної матриці позначимо через $A_{(r,j)}(K(f))$.

Функцію, ядро якої будується з елементів множини $A_{(r,j)}(K(f))$, позначимо через $A_{(r,j)}(f)$

Очевидно, що коли $K(f) = f^{-1}(i)$, то ядром функції $A_{(r,j)}(f)$ будуть булеві набори, на яких функція $A_{(r,j)}(f)$ набуває значення i , де $i \in \{0, 1\}$. Покладемо, що $A_{(r,0)}(K(f)) = K(f)$ і $A_{(r,0)}(f) = f$.

2. $B(K(f))$ — означає, що ядро $K(f)$ булевої функції $f(x_1, \dots, x_n)$ заміниться ядром $K(f')$ функції $f'(x_1, \dots, x_n) = x_j \oplus f(x_1, \dots, x_n)$ і $K(f)$ ($j \in \{1, 2, \dots, n\}$).

Визначимо композиції $B \circ A$, $A \circ A$ операторів B, A так:

$$(B \circ A)K(f) = B(A(K(f))),$$

$$(A \circ A)K(f) = A(A(K(f))) = A(K(f)),$$

де $i, j_1, j_2 \in \{0, 1, \dots, n\}$ і $j_1 \neq j_2$.

Теорема 1. Якщо в множині зведених ядер $T(f)$ булевої функції $f(x_1, \dots, x_n)$ хоча б для одного зведеного ядра $K(f)_i$ можна вказати такі оператори B, A , елементи $\sigma \in S_n, \xi \in S_q$ ($q = |K(f)|$) і таку матрицю толерантності $L \in E_n^-$, що

$$(B \circ A)K_\xi^\sigma(f)_i = L(q), \quad (1)$$

то булева функція $f(x_1, \dots, x_n)$ реалізується комбінаційною схемою, що складається з одного нейронного елемента та суматорів за mod 2.

Доведення. Розглянемо такі можливі випадки:

1. $B = B$ і $A = A$;
2. $B = B$ і $A \neq A$;
3. $B \neq B$ і $A = A$;
4. $B \neq B$ і $A \neq A$;

У випадку 1 маємо:

$$(B \circ A)K_\xi^\sigma(f)_i = B(A(K_\xi^\sigma(f)_i)) = B(K_\xi^\sigma(f)_i) = K_\xi^\sigma(f)_i.$$

Отже, $K_\xi^\sigma(f)_i = L(q)$ ($L \in E_n^-$) і функція $f(x_1, \dots, x_n)$ реалізується одним НЕ.

Якщо $r = 2^{n-i_1} + \dots + 2^{n-i_k}$, то в другому випадку з (1) випливає:

$$L(q) = B(A(K_\xi^\sigma(f)_i)) = A(K_\xi^\sigma(f)_i).$$

Тоді, згідно з оператором A , функція $f(x_1, \dots, x_{j-1}, x'_j, x_{j+1}, \dots, x_n)$ ($x'_j = x_j \oplus x_{i_1} \oplus \dots \oplus x_{i_k}$) реалізується одним НЕ. Отже, для реалізації функції $f(x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n)$ необхідно мати один НЕ, що реалізує функцію $f(x_1, \dots, x_{j-1}, x'_j, x_{j+1}, \dots, x_n)$ і один суматор за mod 2, який розташований між вхідними змінними x_1, \dots, x_n та нейронним елементом. Схематично це можна зобразити так:

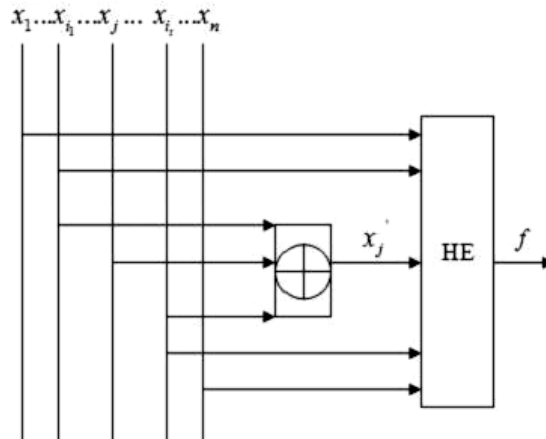


Рис. 1. Реалізація функції $f(x_1, \dots, x_j, \dots, x_n)$ у випадку 2

Нехай $B \neq B$ і $A = A$. Тоді з (1) маємо:

$$L(q) = B \left(A (K_{\xi}^{\sigma}(f)_i) \right) = B(K_{\xi}^{\sigma}(f)_i).$$

Отже, функція $f'(x_1, \dots, x_n) = x_s \oplus (x_1, \dots, x_n)$ реалізується одним НЕ. Звідси $f(x_1, \dots, x_n) = x_s \oplus f'(x_1, \dots, x_n)$. Схематично це зображається так:

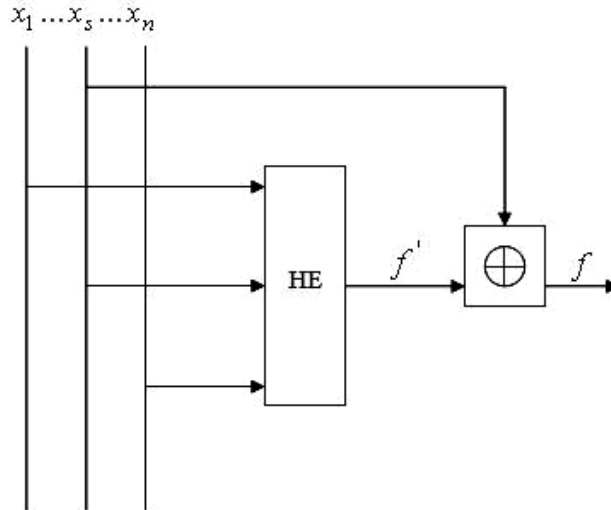


Рис. 2. Реалізація функції $f(x_1, \dots, x_n)$ у випадку 3

У четвертому випадку $L(q) = (B \circ A) K_{\xi}^{\sigma}(f)_i$, а це означає, що функція $f(x_1, \dots, x_n) = x_s \oplus A(f)_{(r,j)}$ реалізується комбінаційною схемою, зображеною на рис. 3.

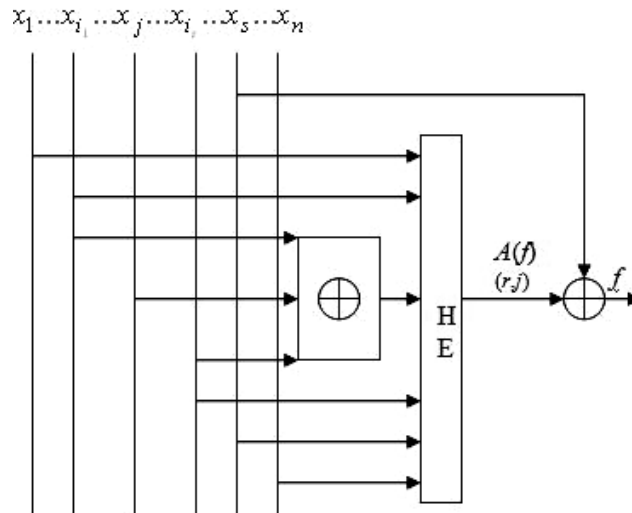


Рис. 3. Реалізація функції $f(x_1, \dots, x_n)$ у випадку 4

Отже, теорему доведено.

Наслідок. Якщо в множині зведених ядер $T(f)$ булевої функції $f(x_1, \dots, x_n)$ хоча б для одного ядра $K(f)_i$ можна вказати такі оператори B, A , елементи $\sigma \in S_n, \xi \in S_q$ і таку матрицю толерантності $L \in E_n^-$, що

$$(B \circ A) K_{\xi}^{\sigma}(f)_i = L(q),$$

то булева функція $f(x_1, \dots, x_n)$ реалізується комбінаційною схемою, що складається з одного нейронного елемента та суматорів за mod 2.

Розглянемо такий приклад. Чи реалізується булева функція $f(x_1, x_2, x_3, x_4)$, ядро якої визначається так: $K(f) = f^{-1}(1) =$

$$= \{\mathbf{a}_1 = (0,1,1,0), \mathbf{a}_2 = (1,1,1,0), \mathbf{a}_3 = (0,0,1,0), \mathbf{a}_4 = (1,0,1,0), \mathbf{a}_5 = (0,1,0,0), \mathbf{a}_6 = (0,1,0,1)\}$$

на одному НЕ або комбінаційною схемою, що складається з одного НЕ та суматорів за mod 2? На основі теореми 1.24 [2] робимо висновок, що булева функція $f(x_1, x_2, x_3, x_4)$ не реалізується одним НЕ, оскільки $\mathbf{a}_6 = \bar{\mathbf{a}}_4$. Побудуємо множину зведених ядер $T(f)$ і за ξ виберемо одиничний елемент групи S_6 .

$$T(f) = \left\{ K_\xi(f)_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, K_\xi(f)_6 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right\}.$$

Простою перевіркою легко переконатися в тому, що

$L_6(6) = A_{(1,3)}(K_\xi(f)_1)$, де $L_6 \in E_4^-$. За матрицею L_6 побудуємо вектор $\mathbf{w} = (-1; -2; -4; -4,5)$, що

задовольняє умову $(L_6 \square L_6^*) \cdot \mathbf{w}^T = \mathbf{c}_w^T$. Знаходимо вектор $\mathbf{w}_1 = \mathbf{a}_1 \mathbf{w} = (-1; 2; 4; -4,5)$ і число $\omega_0 = (\mathbf{a}_z, \mathbf{w}_1) = 1,5$, де \mathbf{a}_z — останній рядок матриці $A_{(1,3)}(K_\xi(f))$. Тоді за теоремою 1 булева функція $f(x_1, x_2, x_3, x_4)$ реалізується за такою комбінаційною схемою:

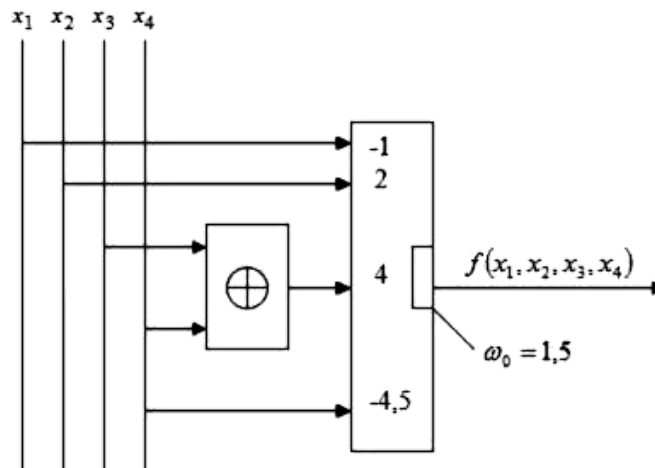


Рис. 4. Реалізація функції $f(x_1, x_2, x_3, x_4)$ у випадку 4

Нехай $f(x_1, \dots, x_n)$ — булева функція в алфавіті $\{0,1\}$ і $h_f(\mathbf{y}) = 1 - 2f(x_1, \dots, x_n)$ відповідна їй функція в алфавіті $\{-1,1\}$ ($y_i = 1 - 2x_i$).

На множині $G_n = \{\mathbf{g} = (\gamma_1, \dots, \gamma_n) \mid \gamma_i \in \{-1,1\}\}$ визначимо функції $\chi_{i_1 \dots i_t}(\gamma)$ ($t \leq n$) так:

$$\forall \mathbf{g} \in G_n \quad \chi_{i_1 \dots i_t}(\mathbf{g}) = y_{i_1}(\mathbf{g}) \dots y_{i_t}(\mathbf{g})$$

і

$$\chi_0(\mathbf{g}) = 1,$$

де $y_i(\mathbf{g})$ — значення змінної y_i на наборі $\mathbf{g} \in G_n$. Якщо в множині G_n задати операцію покоординатного множення векторів, то G_n можна розглядати як мультиплікативну абелеву групу типу $\underbrace{(2) \times \dots \times (2)}_n$, і функції $\chi_{i_1 \dots i_t}, \chi_0 \in$ характеристиками групи G_n над полем дійсних чисел R .

Під спектром булевої функції $h(y_1, \dots, y_n)$ будемо розуміти 2^n -вимірний вектор

$$\mathbf{s}_h = (s_0, s_1, \dots, s_n, s_{12}, \dots, s_{(n-1)n}, s_{123}, \dots, s_{12\dots n}),$$

Де

$$s_0 = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) \chi_0(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}), \quad (2)$$

$$s_{i_1 \dots i_t} = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) \chi_{i_1 \dots i_t}(\mathbf{g}). \quad (3)$$

Нехай B, A — наведені вище оператори і I_r — множина індексів $\{i_1, \dots, i_t\}$, які

задовольняють умову $r = 2^{n-i_1} + \dots + 2^{n-i_t}$. Далі розглянемо ті перетворення у спектральній області булевих функцій, які відповідають операторам B, A . Вводимо такі позначення: якщо булеві

функції $f(x_1, \dots, x_n)$ в алфавіті $\{0,1\}$ відповідає булева функція $h(y_1, \dots, y_n) = 1 - 2f(x_1, \dots, x_n)$ в алфавіті $\{1, -1\}$, то булевим функціям в алфавіті $\{0,1\}$, які визначаються операторами $A(K(f))$ і

$B(K(f))$, відповідають функції $A(h)$ і $B(h)$.

Теорема 2. Якщо \mathbf{s}_h — спектр булевої функції $h(y_1, \dots, y_n)$ і \mathbf{s}'_h — спектр булевої функції $A(h)$, то спектр \mathbf{s}'_h знаходимо з спектра \mathbf{s}_h так: у кожному індексі спектральних коефіцієнтів \mathbf{s}_h , що містять j , індекси з I_r викреслюються, якщо вони присутні, і додаються, якщо вони відсутні.

Доведення. За означенням оператора $A(K(f))$ ($r = 2^{n-i_1} + \dots + 2^{n-i_t}$) та з того, що додаванню за mod 2 в алфавіті $\{0,1\}$ відповідає множення в алфавіті $\{-1,1\}$, маємо:

$$A(h(y_1, \dots, y_{j-1}, y_j, y_{j+1}, \dots, y_n)) = h(y_1, \dots, y_{j-1}, y'_j, y_{j+1}, \dots, y_n),$$

де $y'_j = y_j y_{i_1} \dots y_{i_t}$ і $I_r = \{i_1, \dots, i_t\}$. Використовуючи формули (2) та (3), знаходимо $\mathbf{s}'_h = (s'_0, s'_1, \dots, s'_n, s'_{12}, \dots, s'_{12\dots n})$:

$$\begin{aligned} s'_0 &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \chi_0(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = \\ &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \chi_0(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \cdot 1 = s_0, \\ s'_j &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \chi_j(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = 2^{-n} \times \\ &\times \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \chi_{j i_1 \dots i_t}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = s_{j i_1 \dots i_t}, \\ s'_k &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \chi_k(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = 2^{-n} \times \\ &\times \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \chi_k(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = s_k, (k \neq j), \\ s'_{j i_1} &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \chi_{j i_1}(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = \\ &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) y_j(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \times \\ &\times y_{i_1}(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \\ &\times y_j(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) y_{i_1}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \dots \\ &\dots \times y_{i_t}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) y_{i_1}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = \\ &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) y_j(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \\ &\times y_{i_2}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \dots \times y_{i_t}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = s_{j i_2 \dots i_t}, \end{aligned}$$

оскільки $\forall \mathbf{g} \in G_n \setminus y_i(\mathbf{g})y_i(\mathbf{g}) = 1$,

$$\begin{aligned} s'_{jk} &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_m, \dots, \gamma_n) \chi_{jk}(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = \\ &= 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) y_j(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) \times \\ &\times y_k(\gamma_1, \dots, \gamma'_j, \dots, \gamma_n) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \\ &\times y_j(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) y_{i_1}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) \times \dots \\ &\dots \times y_{i_l}(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) y_k(\gamma_1, \dots, \gamma_j, \dots, \gamma_n) = s_{j i_1 \dots i_l k}, \end{aligned}$$

якщо $k \notin I_r$ і т.д. Отже, $s'_0 = s_0, s'_j = s_{j i_1 \dots i_l}, s'_k = s_k (k \neq j), s'_{j i_1} = s'_{j i_2 \dots i_l}, s'_{jk} = s_{j i_1 \dots i_l k}, s'_{j i_1 \dots i_l} = s_j, \dots$, і теорему доведено.

Теорема 3. Якщо s_n — спектр булевої функції $h(y_1, \dots, y_n)$ і s'_h — спектр булевої функції $B_j(h)$, то спектр s'_h знаходимо зі спектра s_n так: у кожному індексі спектральних коефіцієнтів s_n вилучається j там, де він є, і дописується туди, де він відсутній.

Доведення. Враховуючи, що операції \oplus — додавання за mod 2 в алфавіті $\{0,1\}$ відповідає операція — множення в алфавіті $\{-1,1\}$, функція $B_j(h)$ у явному вигляді задається так:

$$B_j(h(y_1, \dots, y_j, \dots, y_n)) = y_j \cdot h(y_1, \dots, y_j, \dots, y_n).$$

Тоді на основі формул (2), (3) маємо:

$$\begin{aligned} s'_0 &= 2^{-n} \sum_{\mathbf{g} \in G_n} B_j(h(\mathbf{g})) \chi_0(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} y_j(\mathbf{g}) \cdot h(\mathbf{g}) \cdot 1 = s_j, \\ s'_j &= 2^{-n} \sum_{\mathbf{g} \in G_n} B_j(h(\mathbf{g})) \chi_j(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} y_j(\mathbf{g}) \cdot h(\mathbf{g}) \cdot y_j(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) = s_0, \end{aligned}$$

оскільки $\forall \mathbf{g} \in G_n y_j(\mathbf{g})y_j(\mathbf{g}) = 1$,

$$\begin{aligned} s'_k &= 2^{-n} \sum_{\mathbf{g} \in G_n} B_j(h(\mathbf{g})) \chi_k(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} y_j(\mathbf{g}) \cdot h(\mathbf{g}) \cdot y_k(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) \cdot \chi_{jk}(\mathbf{g}) = s_{jk}, (k \neq j), \\ s'_{jk} &= 2^{-n} \sum_{\mathbf{g} \in G_n} B_j(h(\mathbf{g})) \chi_{jk}(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} y_j(\mathbf{g}) \cdot h(\mathbf{g}) \cdot y_j(\mathbf{g}) \times \\ &\times y_k(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) y_k(\mathbf{g}) = 2^{-n} \sum_{\mathbf{g} \in G_n} h(\mathbf{g}) \chi_k(\mathbf{g}) = s_k, \end{aligned}$$

і т.д. Отже,

$$s'_0 = s_j, s'_j = s_0, s'_k = s_{jk}, s'_{jk} = s_k, s'_{jkm} = s_{km}, s'_{km} = s_{jkm}, \dots, s'_{1 \dots j-1 j+1 \dots n} = s_{1 \dots j-1 j j+1 \dots n}$$

і теорему доведено.

Якщо через $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ відповідно позначити інваріантні операції [3], тоді теорема 2.1 [2] спектральною мовою переписується так:

Теорема 4. Якщо спектр s_h булевої функції $h(y_1, \dots, y_n) = 1 - 2f(x_1, \dots, x_n)$ ($f(x_1, \dots, x_n) \in \{0,1\}, x_i \in \{0,1\}, y_i = 1 - 2x_i$) за допомогою інваріантних операцій $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ та операторів $A_{(r,j)}(h)$ $B_j(h)$ можна перетворити так, щоб $(n+1)$ -вимірний початковий відрізок перетвореного спектра s'_h був канонічним характеристичним вектором деякої булевої функції, що реалізується одним нейронним елементом, то функція $f(x_1, \dots, x_n)$ може бути реалізована на комбінаційній схемі, що складається з одного нейронного елемента та суматорів за mod 2.

Застосування теореми 4 для синтезу комбінаційних схем із суматорів за mod 2 і одного нейронного елемента покажемо на прикладі. Знову розглянемо функцію

$$f : f^{-1}(1) = \{(0,1,1,0), (1,1,1,0), (0,0,1,0), (1,0,1,0), (0,1,0,0), (0,1,0,1)\},$$

за якою побудуємо $h = 1 - 2f$. Спектр \mathbf{s}_h функції h знаходимо за допомогою швидкого перетворення [4, 5]. Матриця швидкого перетворення C для знаходження спектральних коефіцієнтів функції h у системі базисних функцій Уолша–Адамара має такий вигляд:

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}.$$

Тоді $2^4 \mathbf{s}_h = C(C(C(Ch)))$. Результати обчислення наведемо у таблиці

y_1	y_2	y_3	y_4	h	Ch	C^2h	C^3h	$2^4 \mathbf{s}_h$	\mathbf{s}_h
1	1	1	1	1	2	2	0	4	s_0
1	1	1	-1	1	0	-2	4	-8	s_4
1	1	-1	1	-1	-2	2	-4	4	s_3
1	1	-1	-1	1	0	2	-4	8	s_{34}
1	-1	1	1	-1	2	-2	0	4	s_2
1	-1	1	-1	-1	0	-2	4	0	s_{24}
1	-1	-1	1	-1	2	-2	4	4	s_{23}
1	-1	-1	-1	1	0	-2	4	0	s_{234}
-1	1	1	1	1	0	2	4	-4	s_1
-1	1	1	-1	1	-2	-2	0	0	s_{14}
-1	1	-1	1	-1	0	2	0	-4	s_{13}
-1	1	-1	-1	1	-2	2	0	0	s_{134}
-1	-1	1	1	1	0	2	4	4	s_{12}
-1	-1	1	-1	1	-2	2	0	0	s_{124}
-1	-1	-1	1	-1	0	2	0	4	s_{123}
-1	-1	-1	-1	1	-2	2	0	0	s_{1234}

Спектральні коефіцієнти s_1, s_2, s_3, s_4, s_0 із точністю до сталої 2^4 задають відповідні координати характеристичного вектора \mathbf{b}_h , тобто $\mathbf{b}_h = (-4, 4, 4, -8; 4)$. За допомогою інваріантних операцій [3] від \mathbf{b}_h переходимо до канонічного характеристичного вектора $\mathbf{b}_h^* = (8, 4, 4, 4; 4)$. Вектор \mathbf{b}_h^* не знаходиться у таблиці канонічних характеристичних векторів нейрофункцій [3], тобто функція f не реалізується одним НЕ.

Розглянемо функцію $h_1 = A_{(1,3)}(h)$. Згідно з теоремою 2 побудуємо $2^4 \mathbf{s}_h$:

$$\begin{aligned} s'_0 &= s_0, s'_1 = s_1, s'_2 = s_2, s'_3 = s_{34}, s'_4 = s_4, \\ s'_{12} &= s_{12}, s'_{13} = s_{134}, s'_{14} = s_{14}, s'_{23} = s_{234}, s'_{24} = s_{24}, \\ s'_{34} &= s_3, s'_{123} = s_{1234}, s'_{124} = s_{124}, s'_{234} = s_{23}, s'_{1234} = s_{123}. \end{aligned}$$

Тоді $\mathbf{b}_{h_1} = (-4, 4, 8, -8; 4)$ і $\mathbf{b}_{h_1}^* = (8, 8, 4, 4; 4)$. Вектор $\mathbf{b}_{h_1}^*$ міститься в таблиці канонічних характеристичних векторів нейрофункцій. Згідно з теоремою 4 функція f реалізується комбінаційною схемою, що складається з одного суматора за mod 2 і одного НЕ (див. рис. 4).

Висновки

1. Введено операції над ядром булевих функцій, за допомогою яких задачу синтезу комбінаційних схем, що складається із одного НЕ з пороговою функцією активації та суматорів за модулем 2, зводять до задачі синтезу одного НЕ, що реалізує перетворену функцію.

2. Встановлено відповідність між перетвореннями над ядром і перетвореннями у спектральній області булевих функцій і мовою характеристичних векторів отримано критерій реалізованості булевих функцій комбінаційною схемою із одного НЕ з пороговою функцією активації та суматорів за модулем 2.

1. Батюк А.Є. Синтез высокопроизводительных специализированных структур для анализа и обработки изображений в пороговом базисе: Гл. 4 / А.Е. Батюк, В.В. Грицьук, Ф.Э. Гече [и др.] // Параллельная обработка информации: монография. В 5 т. Т. 5 / [авт. коллектив]: ред. Б.Н. Малиновский, В.В. Грицьук. – К.: Наук. думка, 1990. – С. 319–363. 2. Гече Ф. Аналіз дискретних функцій та синтез логічних схем у нейробазисі : монографія / Ф. Гече. – Ужгород : Видавництво В. Падяка, 2010. – 210 с. 3. Дертоузос М. Пороговая логика / М. Дертоузос. – М. : Мир, 1967. – 342 с. 4. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение / Л.А. Залманзон. – М. : Наука, 1989. – 493 с. 5. Ярославский Л.П. Введение в цифровую обработку изображений / Л.П. Ярославский. – М. : Советское радио, 1979. – 312 с.

КОМП'ЮТЕРНЕ ОПРАЦЮВАННЯ ДИХАЛЬНОГО ШУМУ СИНФАЗНИМ МЕТОДОМ ДЛЯ ПІДВИЩЕННЯ ІНФОРМАТИВНОСТІ АУСКУЛЬТАТИВНИХ СИСТЕМ

© Дедів І., 20132

Опрацьовано сигнал дихального шуму як періодично корельовану випадкову послідовність синфазним методом, який дає змогу оцінити стан дихальної системи з врахуванням механізму його породження (поєднання властивостей періодичності із стохастичністю). На основі результатів комп'ютерного опрацювання в середовищі Matlab 7.0 встановлено, що отримані оцінки кореляційних компонент є інформативно-інваріантними ознаками дихального шуму, котрі відповідають функціональному стану дихальної системи людини (норма або патологія) і зорієнтовані на підвищення інформативності аускультативних систем.

Ключові слова: дихальний шум, синфазний метод, аускультативна система

Processed signal respiratory noise as periodically correlated random sequence sinphase method which allows to assess respiratory health in view of the mechanism of its generation (combined frequency properties of stochasticity). Based on the results of computer processing environment in Matlab 7.0 installed, that estimates of correlation components is informative-invariant features of the respiratory noise, corresponding to the functional state of human respiratory system (normal or pathological) and aims to enhance information content auscultatory systems.

Key words: respiratory noise, phase method, auscultation system

Вступ

Як відомо, в медичній практиці широко використовується діагностика захворювань дихальної системи людини шляхом вислуховування дихальних шумів на поверхні грудної клітки. Вивчення фізичних механізмів генерації шумів дихання [1–3] показує, що найінтенсивнішими їх джерелами є пульсації тиску турбулентного потоку повітря у великих дихальних шляхах. Деякі захворювання (наприклад, бронхіти) [4] призводять до зміни характеру турбулентності потоку і, відповідно, до варіацій параметрів джерел шумів дихання, що надає специфічного забарвлення дихальним шумам, що і є діагностичною ознакою того або іншого захворювання.

Останнім часом велику увагу звертають на розроблення методів комп'ютерної діагностики захворювань дихальної системи. Опрацювання дихальних шумів при цьому передбачає формування опису їх на основі певної моделі з подальшим перетворенням одержаного подання до потрібної форми. Останнім кроком у процесі опрацювання є виділення і використання інформативного вмісту сигналу. Модель дихального шуму має містити інформативну характеристику – ознаку зміни в роботі відповідних органів чи систем.

У зв'язку з цим ведеться пошук об'єктивних характеристик, чутливих до змін акустичних властивостей дихальної системи і джерел дихальних шумів. У роботах [5–8] наголошують на інформативності при діагностиці захворювань дихальної системи спектрально-кореляційних характеристик шумів дихання, що реєструються на поверхні грудної клітки. За модель дихального шуму у цьому випадку приймають стаціонарний процес. До переліку інформативних характеристик дихального шуму входять просторова кореляційна функція і пов'язані з нею функції спектральної густини потужності та просторової когерентності.

Водночас, інтерпретація даних – результатів вимірювань цих характеристик часто утруднена через відсутність інформації про статистичні властивості джерел і особливості трансформації поля шумів дихання у разі розповсюдження в дихальній системі. Необхідно також враховувати спотворення сигналів навколишніми завадами.

Відомо, що функціональні порушення, спричинені патологічними станами, призводять до появи в дихальних шумах нестационарності при зображенні їх як кусково чи локально стаціонарних процесів, або зміни типу нестационарності [9]. Тому адекватним задачі медичної діагностики є подання дихального шуму як стохастичного нестационарного процесу.

У праці Я.П. Драгана, І.Ю. Дедів [10] на основі аналізу ймовірнісних характеристик дихального шуму обґрунтовано його модель як нестационарний випадковий процес у вигляді періодично корельованого випадкового процесу (ПКВП), що є адекватною задачі виявлення змін у функціональному стані дихальної системи.

Формулювання задачі

На основі математичної моделі дихального шуму як ПКВП поставлено задачу використання синфазного методу для опрацювання сигналу. Застосування цього методу дасть змогу розширити можливість діагностики стану дихальної системи людини шляхом впровадження в область аускультатії нового класу інформативних ознак.

Синфазний метод аналізу дихального шуму як періодично корельованого випадкового процесу

Одним із методів опрацювання дихального шуму (ДШ) як ПКВП є синфазний метод [2], який ґрунтується на тому, що відліки значень через період корельованості при різному виборі початку відліку (початкової фази) $t_0 \in [0, T)$ утворюють стаціонарну ергодичну векторну випадкову послідовність $\{\xi(t_0), t_0 \in [0, T)\}$, де $\xi(t_0) \equiv \{\xi(t_0 + kT), k \in Z\}$. За цим методом характеристику ДШ – оцінку коваріації $\hat{b}_\xi(t, u)$, компоненти якої дають змогу оцінити часову мінливість сигналу, обчислено за виразом:

$$\hat{b}_\xi(t, u) = \frac{1}{N} \sum_{k=0}^{N-1} \xi^0(t + u + kT) \xi^0(t + kT), \quad (1)$$

де $\xi^0(t) = \xi(t) - \hat{m}(t)$ – центрований ДШ $\xi(t)$.

Компонентний метод виходить з того, що характеристики ДШ є періодичними функціями від часу, а тому подані за допомогою розкладів типу рядів Фур'є:

$$\hat{b}_\xi(t, u) = \sum_{k \in Z} \hat{B}_k(u) \exp\left(ik \frac{2\pi}{T} t\right), \quad (2)$$

де $\hat{B}_k(u)$ – оцінки кореляційних компонент, котрі кількісно характеризують фазово-часову структуру сигналу дихального шуму (СДШ):

$$\hat{B}_k(u) = \frac{1}{T} \int_0^T \hat{b}_\xi(t, u) \exp\left(-ik \frac{2\pi}{T} t\right) dt, \quad k \in Z. \quad (3)$$

Алгоритм комп'ютерного опрацювання синфазним методом дихального шуму як періодично корельованої випадкової послідовності

Оцінки кореляційних компонент СДШ як періодично корельовані випадкової дискретної послідовності, отримані синфазним методом, обчислено за виразом:

$$\hat{B}_k(u) = \frac{1}{N_T} \sum_{n=0}^{N_T-1} \hat{b}_\xi(n\Delta t, u) \exp\left(-ik \frac{2\pi}{N_T} n\right), \quad (4)$$

де N_T – дискретний період корельованості СДШ, $N_T = T \cdot \Delta t$, u – зсув, Δt – крок дискретизації, $\hat{b}_\xi(n\Delta t, u) = \frac{1}{N} \sum_{k=0}^{N_k-1} \xi(n\Delta t + u + k1N_T) \xi^0(n\Delta t + k1N_T)$ – оцінка коваріаційних компонент, де N_k – кількість відгуків СДШ.

За виразом (4) розроблено формулу алгоритму комп'ютерного опрацювання ДШ синфазним методом (5) із використанням алгебри алгоритмів В.К. Овсяка, яка, на відміну від вербального та блок-схемного описів, забезпечує точний опис, мінімізацію за кількістю дій (унітермів) та дослідження алгоритмів.

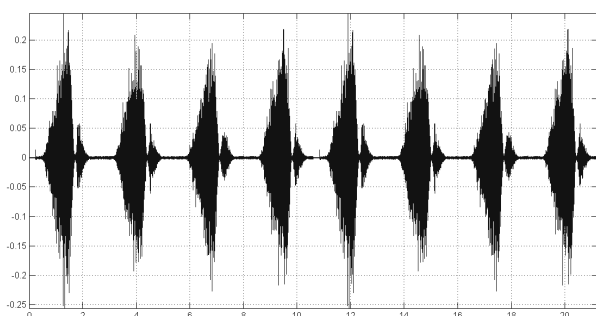
$$\begin{array}{l}
 \xi(i) \\
 ; \\
 NT=512 \\
 ; \\
 \mathcal{Z}(u \leq NT) \\
 \hline
 \mathcal{Z}(u \leq Nu) ; \\
 \mathcal{Z}(k1 \leq (Nk-1)) ; \quad c_{(u \leq NT)} ; \quad (u \leq Nu) \text{--?} \\
 \hline
 (b(u, u) = b(u, u) + ; c_{(u \leq Nu)} ; (k1 \leq (Nk-1)) \text{--?} \\
 * \xi(u + k1 * NT) * \\
 * \xi(u + k1 * NT + u) \\
 ; \\
 c_{(k1 \leq (Nk-1))} \\
 \hline
 \hline
 b(u, u) = b(u, u) / Nk ; \quad (u \leq NT) \text{--?} \\
 ; \\
 \mathcal{Z}(k \leq K) \\
 \hline
 \mathcal{Z}(u \leq Nu) \quad ; \quad |B_k(u)| / NT ; \quad (k \leq K) \text{--?} \\
 \hline
 \mathcal{Z}(u \leq NT) ; \quad c_{(k \leq K)} ; \quad (u \leq Nu) \text{--?} \\
 \hline
 B_{k+1}(u) = B_{k+1}(u) + ; c_{(u \leq Nu)} ; \quad (u \leq NT) \text{--?} \\
 + b(u, u) * \\
 * \exp(-j2\pi ku / NT) \\
 ; \\
 c_{(u \leq NT)}
 \end{array} \quad (5)$$

$u \in \overbrace{u_0; u_1; u_2; \dots; Nu}, \quad i \in \overbrace{1; 2; 3; \dots; N}, \quad k1 \in \overbrace{0; 1; 2; \dots; Nk-1}, \quad k \in \overbrace{0; 1; 2; \dots; K}, \quad n \in \overbrace{1; 2; 3; \dots; NT}.$

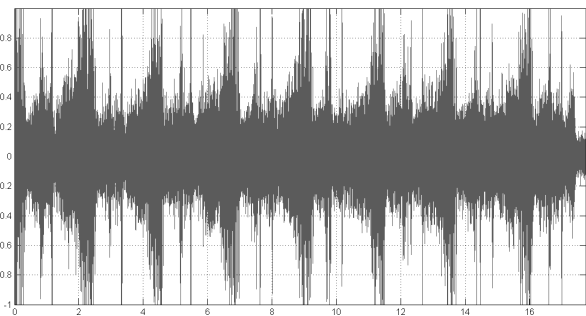
Розроблена формула алгоритмів комп'ютерного опрацювання ДШ (5) дає змогу розробити програмне забезпечення для оцінювання його характеристик з метою виявлення нових інформативних ознак в області аускультології на основі математичної моделі у вигляді періодично корельованої випадкової послідовності із дискретним часом.

Результати синфазного опрацювання

На рис. 2 зображено зареєстровані електронним стетоскопом фірми Littmann® модель 3000 дихальні шуми: пацієнта А з нормою та пацієнта Б із патологією.



Пацієнт А – норма



Пацієнт Б - патологія

Рис. 1. Реалізації експериментально зареєстрованих дихальних шумів

Враховуючи алгоритм опрацювання ДШ (рис. 2) синфазним методом (5), отримано результати, які зображено на рис.3.

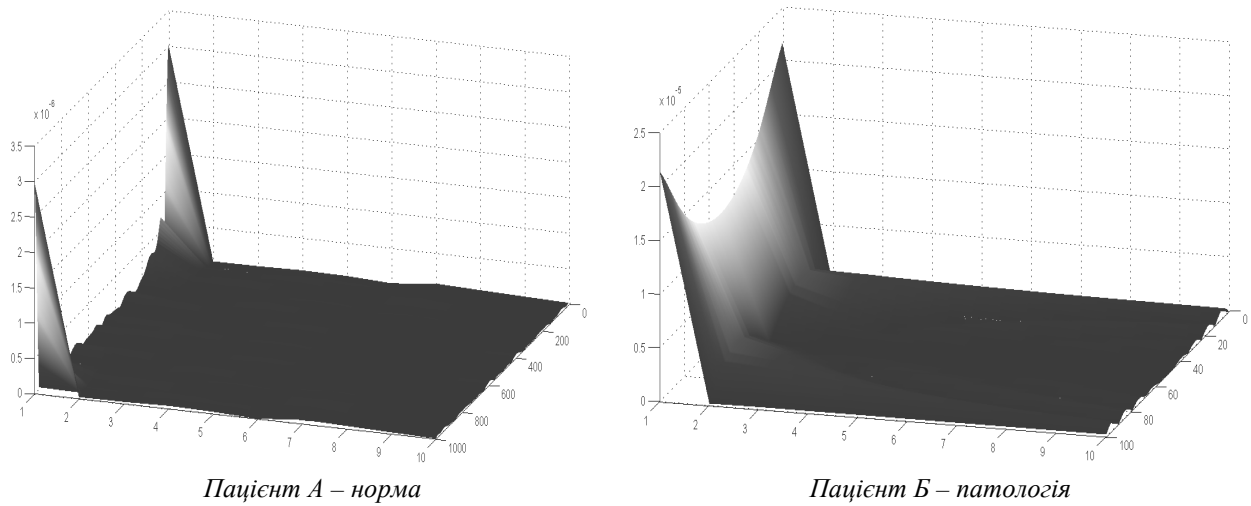


Рис. 2. Реалізації кореляційних компонент ДШ, обчислених синфазним методом

Для оцінювання кореляційних компонент (рис. 2) використано оцінку математичного сподівання:

$$M_u \{ \hat{B}_k(u) \} = \frac{1}{N_u} \sum_{u=1}^{N_u} \hat{B}_k(u), \quad u = \overline{1, N_u}, \quad k = \overline{1, N_k}, \quad (6)$$

де k – номер кореляційної компоненти ДШ; u – зсув; N_u – довжина зсуву; N_k – кількість компонент.

Реалізації оцінок математичних сподівань для кореляційних компонент ДШ зображено на рис. 3.

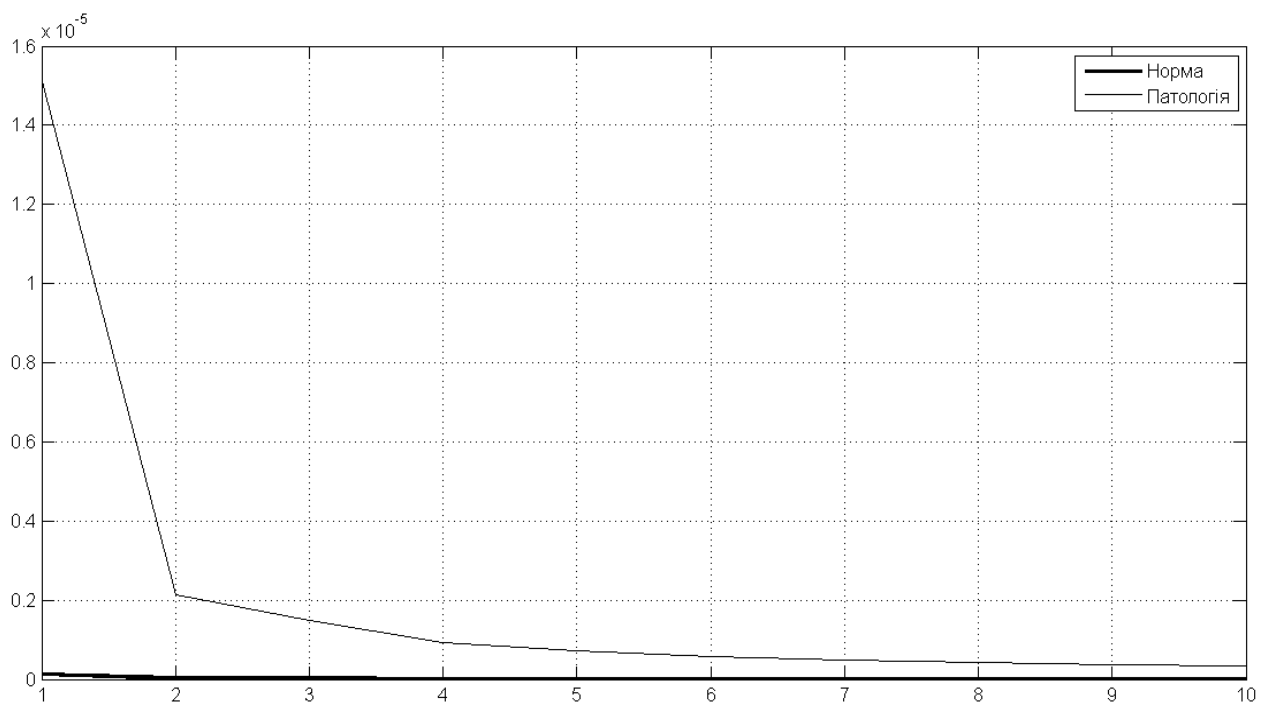


Рис. 3. Оцінки математичного сподівання кореляційних компонент ДШ

На рис. 3 видно, що для пацієнта з нормою та патологією кореляційні компоненти зосереджені на одних і тих самих компонентах, проте амплітудні значення компонент різні, що свідчить про чіткі зміни у функціонуванні дихальної системи людини (норма чи патологія).

Висновки

У результаті комп'ютерного опрацювання дихального шуму синфазним методом в середовищі Matlab 7.0 отримано нові інформативні ознаки – кореляційні компоненти, які відповідають функціональному стану дихальної системи людини і так дають змогу підвищити інформативність аускультативних систем.

1. *Болезни органов дыхания: Руководство для врачей. В 4-х томах / Ред. Н.Р. Палеева. – Т. 2. Частная пульмонология. – М: Медицина, 1989. – 512 с.* 2. Драган Я.П. *Енергетична теорія лінійних моделей стохастичних сигналів / Я.П. Драган. – Львів: Центр стратегічних досліджень еко-біо-технічних систем, 1997. – XVI+333 с.* 3. *Forgacs P. Lung sounds / P. Forgacs // Brit. J. Dis. Chest. – 1969. – 63. – P. 1–12.* 4. *Breath sounds and regional ventilation / Y. Ploysongsang, R. Martin, W. Ross, R. Loudon, P. Machlem // Am. Rev. Respir. Dis. – 1977. – 116. – P. 187–199.* 5. *Loudon R. Lung sounds / R. Loudon, R. Murphy // Am. Rev. Respir. Dis. – 1984. – 130. – P. 663–673.* 6. *Замотаев И.П., Магазаник Н.А., Водолазкий Л.А., Голиков В.А., Щедрина О.И. Спектральный анализ важнейших аускультативных признаков // Клиническая медицина. – 1974. – 52, № 5. – С. 97–101.* 7. *Gavriely N., Palti Y., Alroy G. Spectral characteristics of normal breath sounds // J. Appl. Physiol. – 1981. – 50, № 2. – P. 307.* 8. *Макаренков А.П., Рудницкий А.Г. Возможности диагностики легочных патологий при двухканальной обработке дыхательных шумов человека // Акуст. ж. – 1995. – 41, № 2. – С. 272–277.* 9. *Вовк И.В., Дахнов С.Л., Крижановский В.В., Олишник В.Н. Возможности и перспективы диагностики легочных патологий с помощью компьютерной регистрации и обработки шумов дыхания // Акуст. Вісн. – 1998. – 1, № 2. – С. 21–33.* 10. *Драган Я.П. Обґрунтування математичної моделі дихальних шумів у вигляді періодично корельованого випадкового процесу / Я.П. Драган, І.Ю. Дедів // Комп'ютерні системи та компоненти. – Чернівці: Чернівецький університет. – 2008. № 426. – С. 93–97.*

ПАСИВНИЙ Т-МІКРОМІКСЕР З БАР'ЄРАМИ ДЛЯ ЛАМІНАРНОГО ПОТОКУ

© Матвійків О., Дмитришин Б., 2012

Змішування в пасивних мікроміксерах залежить від їх геометрії. Отже, багато пасивних мікроміксерів ефективні тільки при великих числах Рейнольдса і вимагають складної структури каналів. Наведено можливості вдосконалення структури Т-мікроміксера для змішування двох рідин з введенням бар'єрів у мікроканалі. У дослідженні ми змінюємо 3 параметри: висоту бар'єра, зсув бар'єра від центру каналу та форму бар'єра. Модель демонструє високу ефективність змішування навіть при малих числах Рейнольдса $Re = 0,1$. Модифікований Т-мікроміксер можна легко інтегрувати в різноманітні лабораторії на чипах (ЛНЧ) або мікросистеми повного аналізу (МПА).

Ключові слова: мікропотоки, лабораторія на чипі, мікросистеми повного аналізу, числа Рейнольдса, змішування, Т-мікроміксер, ламінарний потік.

Mixing in the passive micromixers depends on their geometry. Thus many passive micromixers are effective only at high Reynolds numbers and require a complex structure of channels. The article presents the possibility of improving the structure of T-micromixer for mixing two fluids with barriers. In our investigation we change 3 parameters: barrier height, barrier offset from the centre of the channel and barrier shape. The model shows high efficiency of mixing even at low Reynolds numbers $Re = 0,1$. Modified T-micromixer can be easily integrated into various labs-on-chip (LOC) or micro-total-analysis systems (μ TAS).

Key words: Microfluidics, lab-on-chip (LOC), micro-total-analysis systems (μ TAS), Reynolds number, mixing, T-shape micromixer, laminar flow.

Вступ

За допомогою мікрорідинних пристроїв можна виконувати безліч операцій, таких як [1]:

- Передпідготовка зразків;
- Підготовка зразків;
- Змішування;
- Прокачування рідини (помпування);
- Аналіз;
- Секвенування ДНК;
- Виокремлення і виявлення клітин.

Змішування відіграє одну з найбільш важливих ролей в реалізації лабораторій-на-чипі (ЛНЧ) або мікросистем повного аналізу (МПА), тому що мікрорідинна обробка та операції, які здійснюються в цих чипах, вимагають швидкого перемішування реагентів і зразків. Швидке змішування двох або більше компонентів або аналітів важливо для багатьох мікрорідинних систем, що використовуються в біохімічному аналізі або аналізі ДНК або в мікрореакторах зі складним хімічним синтезом. Перемішування в мікромасштабі – дуже складний процес, тому що це відбувається за рахунок дифузії, яка є дуже повільним процесом в мікромасштабі (масовий коефіцієнт дифузії ($D \sim 10^{-10} \text{ м}^2/\text{с}$). У більшості мікрорідинних пристроїв потік ламінарний з дуже

низьким числом Рейнольдса і дифузія домінує в процесі змішування. Для потоку в мікроканалі число Рейнольдса, як правило, визначається як [2]:

$$Re = \frac{uD_h}{\nu}, \quad (1)$$

де D_h – гідравлічний діаметр мікроканалу (с), u – середня швидкість об'єкта відносно рідини (м/с), ν – кінематична в'язкість (м²/с). Ламінарний потік виникає при малих числах Рейнольдса, коли сили в'язкості переважають сили інерції і характеризується сталістю розподілу швидкостей рідини. Турбулентний потік виникає при великих числах Рейнольдса, коли переважають сили інерції – він зазвичай викликається хаотичними вихорами та іншими нестабільностями потоку. Цілковите й повне перемішування рідини при малих числах Рейнольдса займає дуже багато часу і вимагає великої довжини мікроканалу. Наприклад, Т-подібний мікроміксер потребує для повного змішування довжини мікроканалу в десятки сантиметрів. Це робить його дуже непрактичним для використання в мікрорідинних лабораторіях на чипі або мікросистем повного аналізу. Оскільки в пасивних мікроміксерах не застосовують жодних зовнішніх сил і перемішування відбувається лише шляхом дифузії, єдиним рішенням є зміна геометрії мікроканалів так, щоб змішування відбувалось швидше на коротшій відстані.

Геометрія мікроміксера

Проектування і моделювання Т-мікроміксера було зроблено в COMSOL Multiphysics. Структуру мікроміксера зображено на рис. 1.

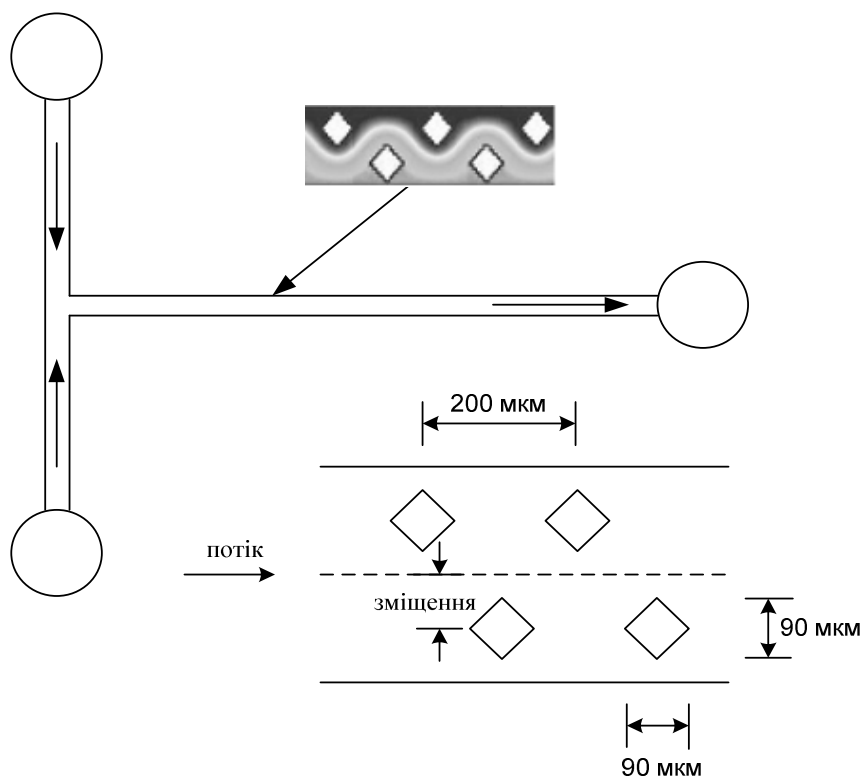


Рис. 1. Структура та розміри Т-мікроміксера з квадратними бар'єрами, поверненими під кутом 45°

Основна ідея цього дослідження полягає в оптимізації і покращенні параметрів змішування шляхом зміни геометрії мікроканалу завдяки введенню бар'єрів і зміни їх розміщення [3]. У нашому дослідженні ми змінюємо 3 параметри: висоту бар'єру, зсув бар'єру від центру мікроканалу та зміну форми бар'єру. Бар'єри у дослідженні квадратної форми розміром 90 x 90 мкм (Ш x Д), відстань між центрами вздовж каналу – 200 мкм, ширина мікроканалу 200 мкм, висота 50 мкм. Бар'єри мають 3 різні форми: квадратну, трикутну і круглу. Всі ці розміри наведено в табл. 1.

Довжина змодельованого мікроканалу становить 5 мм. Моделювання було зроблено для стаціонарної нестисливої течії, дві рідини мають фізичні властивості води з концентрацією частинок в рідині – перша 0, а друга 1, коефіцієнт дифузії $D \sim 10^{-10} \text{ м}^2 / \text{с}$. Швидкість $V \sim 0,1-1000 \text{ мм} / \text{с}$. Змішування припиняється, коли молярна інтенсивність дорівнює 0,5 для обох рідин. Також було здійснено поворот квадратних бар'єрів на кут у 45° , щоб запобігти утворенню мертвих об'ємів у мікроканалі (табл. 1).

Таблиця 1

Розміри і параметри Т-мікроміксерів з бар'єрами

Мікроканал	Розміри	
	Ширина	200 мкм
	Висота	50 мкм
	Довжина	5 мм
Бар'єри	Розміри	
	Ширина	90 мкм
	Висота	0-50 мкм
	Довжина	90 мкм
	Зміщення	0-50 мкм
	Інтервал	200 мкм
Бар'єри	Форма	
	Квадратна	◆
	Трикутна	◄
	Кругла	●

Результати моделювання модифікованого Т-мікроміксерів

Всі моделювання проводили при малих числах Рейнольдса $Re = 0,1$. Моделювання результатів для різних висот бар'єрів показано на рис. 2. Результати демонструють, що найкращого змішування – 48 % – було досягнуто, коли висота бар'єра в мікроканалі дорівнювала висоті мікроканалу (50 мкм), без бар'єрів ефективність змішування Т-подібного мікроміксерів становила лише 22 %.

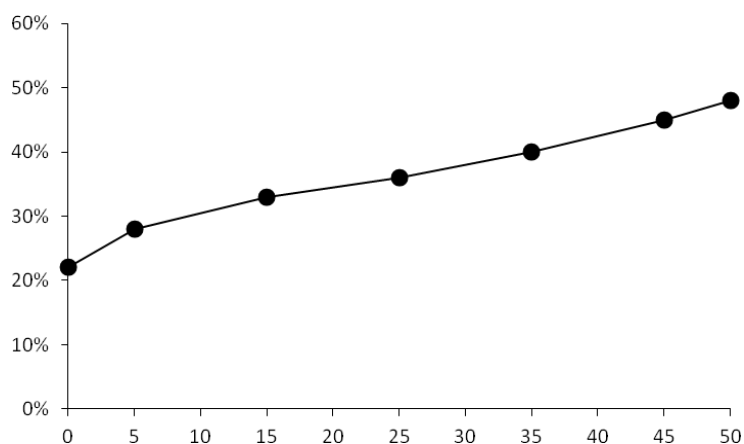


Рис. 2. Відсоток змішування для різних висот бар'єрів

Для дослідження впливу форми бар'єрів дослідження було проведено для 3-х типів геометрії: кругла, трикутна, квадратна (результати наведено в табл. 2). Кожна форма бар'єру має ті самі розміри 90 x 90 мкм і 50 мкм відступ від центральної осі каналу. В результаті дослідження круглі бар'єри продемонстрували найгірше змішування (42 %), трикутні бар'єри продемонстрували найкраще перемішування (51 %), краще, ніж квадратні бар'єри, повернуті під кутом 45° (49 %), але трикутні бар'єри утворюють мертві об'єми, а вони небажані для подальших досліджень.

Також було досліджено зміщення бар'єрів від центральної осі мікроканалу до його стінок за фіксованої висоти бар'єрів у 50 мкм і виявлено, що 20 мкм зсув від центральної осі дає 62 % змішування для 5 мм довжини мікроканалу і 40 мкм зсув дає тільки 39 % для 5 мм довжини мікроканалу, але перше зміщення дає тільки ~ 73 % змішування для 10 мм довжини мікроканалу, а друге дає ~ 79 % змішування для 10 мм довжини мікроканалу. Отримані результати наведено на рис. 3.

Ефективність змішування для 3 типів геометрії бар'єрів

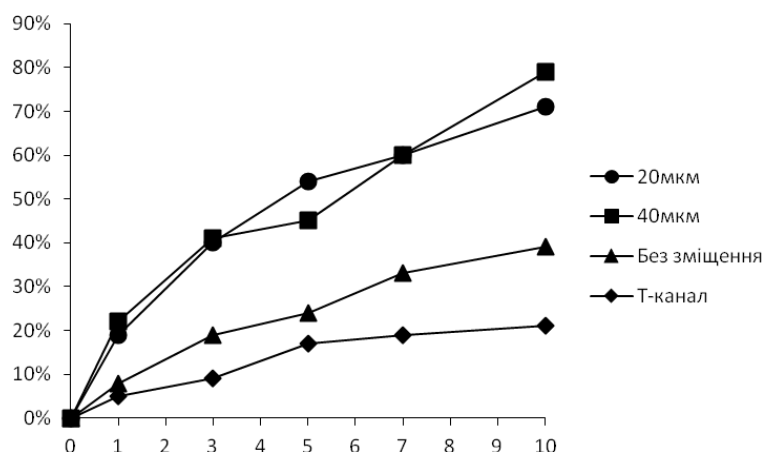
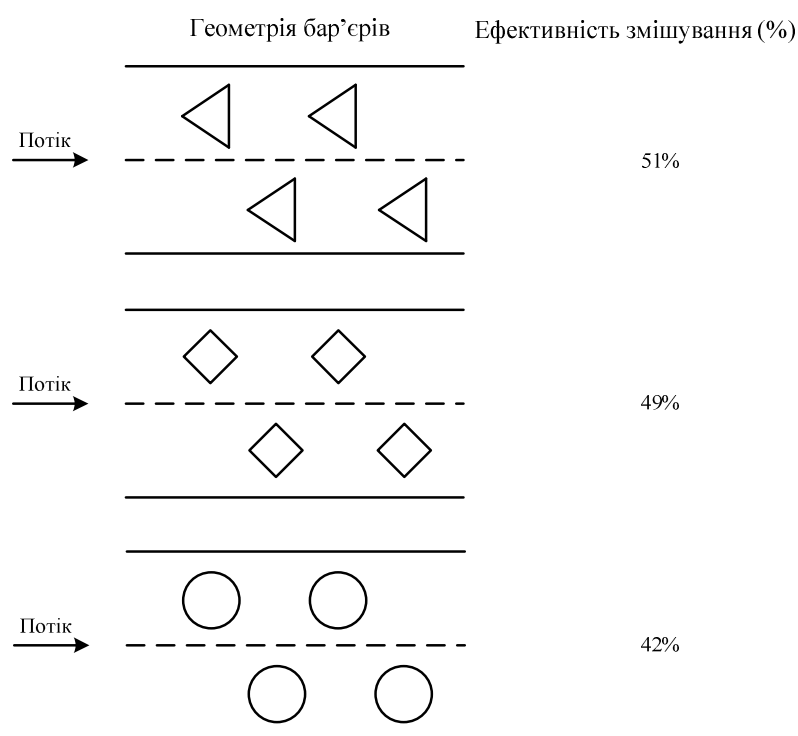


Рис. 3. Результати для різних зміщень бар'єрів від центральної осі мікроканалу

Висновки

Наведено модифікований Т-мікроміксер. Геометрію пасивного Т-мікроміксера було змінено завдяки інтеграції в структуру мікроканалу бар'єрів різної висоти, зміщених на різну величину від центру каналу та різних форм. Експериментально було встановлено, що оптимальних параметрів змішування було досягнуто, коли висота бар'єрів дорівнювала висоті мікроканалу (50 мкм), зі зміщенням бар'єрів від центру на 20–40 мкм і з квадратною формою бар'єрів, повернених під кутом 45°. Модифікація Т-мікроміксера забезпечує ефективне перемішування при малих числах Рейнольдса, і її надзвичайно легко реалізувати та інтегрувати в комплекс складніших мікропристроїв.

1. Berthier J., Silberzan P. *Microfluidics for biotechnology*. Artech House. – 2006. – P. 288–289.
2. Evans J.D., Liepmann D., Pisano A.P. *Planar Laminar Mixer, MEMS, Nagoya, Japan*. – 1997. – P. 96–101.
3. Melin J., Gimenez G. *A fast passive and planar liquid sample micromixer, Lab Chip* 4. – 2004. – P. 214.

ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ЗАСОБІВ ГЕНЕРУВАННЯ ПРОГРАМНИХ МОДЕЛЕЙ СПЕЦІАЛІЗОВАНИХ ПРОЦЕСОРІВ У САМОКОНФІГУРОВНІЙ КОМП'ЮТЕРНІЙ СИСТЕМІ

© Мельник В., 2012

Проаналізовано особливості засобів генерування програмних моделей обчислювальних пристроїв, порівняно характеристики цих засобів та запропоновано рекомендації щодо застосування цих засобів під час побудови самоконфігуровної комп'ютерної системи.

Ключові слова: самоконфігуровна комп'ютерна система, засоби генерування програмних моделей процесорів, програмні моделі спеціалізованих процесорів.

An overview and analysis of the features of the IP Cores generation facilities and a comparison of the characteristics of these facilities are presented in the paper. Recommendations in regard of their use in self-configurable computer system design are proposed.

Key words: self-configurable computer system, IP Cores generation facilities, specialized processors IP Cores.

Вступ

Протягом останнього десятиліття відбулися значні зміни у галузі проектування та мікроелектронного виробництва комп'ютерних засобів. Було розроблено технологію проектування програмних моделей обчислювальних пристроїв [1], яка описує процес розроблення архітектури обчислювальних пристроїв на рівні міжрегістрових передач з використанням мов опису апаратних засобів (*VHDL* [2], *Verilog* [3] та ін.) та їх реалізації у вигляді надвеликих інтегральних схем на базі програмованих логічних інтегральних схем (ПЛІС). Поширення ПЛІС сприяло відновленню інтересу до розроблення високопродуктивних апаратних прискорювачів, що функціонують у взаємодії з універсальними, зокрема й персональними, комп'ютерами. Апаратні прискорювачі, побудовані на базі ПЛІС (їх ще називають реконфігуровними прискорювачами [4]), мають істотні переваги порівняно з традиційними, а саме – вони можуть бути багаторазово перепрограмовані та мають значно нижчу вартість, ніж прискорювачі на базі замовних спеціалізованих надвеликих інтегральних схем (НВІС) у випадку несерійного виробництва.

Комп'ютерні системи, в яких універсальні комп'ютери об'єднані з реконфігуровним середовищем на основі мікросхем програмовної логіки, називають реконфігуровними комп'ютерними системами. Об'єднання універсального комп'ютера з реконфігуровним середовищем в єдину систему дає змогу виконувати обчислювальні завдання значно швидше, що забезпечується створенням в цьому середовищі спеціалізованих процесорів для апаратного виконання обчислювально найскладніших частин завдань.

Поряд з високою продуктивністю, яка забезпечується реконфігуровними комп'ютерними системами (РККС) на основі універсальних процесорів та реконфігуровних прискорювачів, є й деякі проблеми, пов'язані з їх застосуванням. Це зокрема їхня недостатня «динамічність» – перед тим, як виконувати обчислювальне завдання в РККС, її необхідно сконфігурувати, тобто синтезувати в її реконфігуровному середовищі спеціалізований процесор, який реалізує цей алгоритм.

Аналіз останніх досліджень

У роботі [5] запропоновано новий клас прискорювачів обчислень в комп'ютерах – самоконфігуровні прискорювачі та концепцію їх побудови. Цю концепцію покладено в основу побудови самоконфігуровних комп'ютерних систем (СККС), які позбавлені притаманних РККС проблем. СККС – це комп'ютерна система, яка містить універсальний комп'ютер, реконфігуровне середовище, побудоване на основі ПЛІС або інших типів інтегральних схем програмовної логіки, а також програмні засоби, які самостійно виділяють з комп'ютерної програми фрагменти, виконання яких в реконфігурованому середовищі прискорює роботу комп'ютерної системи, самостійно створюють в цьому середовищі пристрої (спеціалізовані процесори) для виконання виділених фрагментів програми та самостійно організують їх функціонування та взаємодію з універсальним процесором. У роботі [6] описано метод самоконфігурування на основі запропонованої концепції.

Постановка завдання

Однією з компонент СККС є система генерування, функцією якої є автоматичне створення програмних моделей спеціалізованих процесорів на основі високорівневого опису обчислювальних алгоритмів. Під програмною моделлю спеціалізованого процесора розуміють його представлення у вигляді програми мовою опису апаратних засобів на рівні міжрегістрових передач або нижчому.

Сьогодні існує значна кількість методів та програмних засобів генерування програмних моделей спеціалізованих процесорів [7–12]. У роботі [13] сформовано вимоги до системи генерування програмних моделей спеціалізованих процесорів, на основі яких можна проаналізувати особливості наявних засобів генерування щодо ефективності їх застосування в СККС. Разом з тим, питання використання цих засобів для побудови системи генерування програмних моделей спеціалізованих процесорів в СККС, яка відповідала б висунутим до неї вимогам, досі не опрацьовувалось. Саме це питання і є предметом нашого наукового дослідження.

1. Функція системи генерування програмних моделей спеціалізованих процесорів

Система генерування програмних моделей спеціалізованих процесорів (далі – система генерування) є одним з базових програмних засобів, що входять до складу СККС і надають їй властивості самоконфігуровності. Функцією системи генерування в СККС є автоматичне створення програмної моделі процесора для виконання обчислювальних алгоритмів, отриманих від іншої компоненти СККС – системи розподілу обчислювального навантаження [18]. Функцію системи генерування можна описати таким виразом:

$$PMSP(A_i, i = \overline{1...z}) = GS(PR^{SCCS}(A_i, i = \overline{1...z})), \quad (1)$$

де GS – система генерування; PR^{SCCS} – програма для апаратного виконання в реконфігурованому середовищі СККС, яка описує обчислювальні алгоритми $A_i, i = \overline{1...z}$; $PMSP$ – програмна модель спеціалізованого процесора для виконання обчислювальних алгоритмів $A_i, i = \overline{1...z}$; z – кількість обчислювальних алгоритмів.

2. Типи засобів генерування програмних моделей обчислювальних пристроїв

До засобів високорівневого проектування програмних моделей обчислювальних пристроїв, які пропонується використати для реалізації системи генерування, належать:

1. Генератори програмних моделей процесорів на основі бібліотек програмних моделей процесорів та процесорних компонент [13];
2. Генератори програмних моделей процесорів на основі конфігурованих програмних моделей процесорів [11, 12];
3. Системи автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня [8–10].

Проаналізуємо особливості застосування зазначених типів програмних засобів в системі генерування СККС відповідно до сформульованих в роботі [14] вимог, та визначимо, як задовольняють ці засоби такі вимоги в частині:

- 1) функціональної повноти системи генерування;
- 2) технічних характеристик генерованих системою генерування моделей процесорів;
- 3) архітектури генерованих системою генерування моделей процесорів.

3. Особливості застосування генераторів на основі бібліотек програмних моделей процесорів

Генератор на основі бібліотеки програмних моделей процесорів можна подати двома складовими: власне бібліотекою, реалізованою, наприклад, як база даних, та засобами пошуку програмної моделі процесора в цій бібліотеці. Пошук програмної моделі в бібліотеці виконується відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора [13]. Бібліотека містить готові програмні моделі процесорів, описані мовою опису апаратних засобів. Вона може бути як типовою і містити моделі процесорів для виконання наборів стандартизованих прикладних алгоритмів та найчастіше використовуваних обчислювальних операцій, так і виконаною на замовлення і орієнтованою на задачі певної галузі [15].

Отже, можна зробити такі висновки щодо відповідності генераторів на основі бібліотек програмних моделей процесорів поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі бібліотек програмних моделей процесорів внаслідок обмеженості бібліотек не може згенерувати модель спеціалізованого процесора для виконання довільного обчислювального алгоритму, тому не може бути функціонально повною.

2. Оскільки програмні моделі процесорів, що входять до складу бібліотеки, створюються заздалегідь, існує можливість врахування в цих моделях на етапі їх створення особливостей архітектури цільової ПЛІС, зокрема ПЛІС реконфігурованого середовища СККС, кількості виділеного в ПЛІС місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним комп'ютером та реконфігурованим середовищем. Це дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Необхідно зазначити, що бібліотека може вмішувати не програмні моделі процесорів, а файли конфігурації цільової ПЛІС, отримані в результаті логічного синтезу цих процесорів. Це дасть змогу скоротити та спростити процес компіляції програми в СККС, уникнувши етапу логічного синтезу процесорів.

Час генерування за допомогою генераторів на основі бібліотек програмних моделей процесорів можна визначити за виразом:

$$t_{gen}(G_{LPPM}) = t_{search} + t_{out}, \quad (2)$$

де t_{search} – час пошуку програмної моделі процесора в бібліотеці; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

4. Особливості застосування генераторів на основі бібліотек програмних моделей процесорних компонент

Генератор на основі бібліотеки програмних моделей процесорних компонент можна подати трьома складовими: бібліотекою, засобами пошуку програмних моделей відповідних компонент в цій бібліотеці і засобами компонування програмних моделей процесорів з компонент та генерування файлів опису верхнього рівня. Пошук програмних моделей у бібліотеці та компонування виконуються відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора.

Принциповими перевагами застосування бібліотеки програмних моделей процесорних компонент над бібліотекою програмних моделей процесорів є розширення функціональності генератора та зменшення об'єму бібліотеки [13]. Це однак не позбавляє цих засобів їх природної обмеженості в частині функціональної повноти. Можна зробити такі висновки щодо відповідності генераторів на основі бібліотек програмних моделей процесорних компонент поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі бібліотек програмних моделей процесорних компонент не може бути функціонально повною.

2. Існує можливість врахування в моделях процесорних компонент на етапі їх створення особливостей архітектури ПЛІС реконфігуровного середовища СККС, кількості виділеного в ній місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним комп'ютером та реконфігуровним середовищем, що дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Час генерування за допомогою генераторів на основі бібліотек програмних моделей процесорних компонент можна визначити з виразу:

$$t_{gen}(G_{LPCPM}) = t_{search} + t_{comp+TLGen} + t_{out}, \quad (3)$$

де t_{search} – час пошуку програмних моделей компонент процесора в бібліотеці; $t_{comp+TLGen}$ – час компонування моделі процесора та генерування файлів опису верхнього рівня; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

5. Особливості застосування генераторів на основі конфігуровних програмних моделей процесорів

Генератор на основі конфігуровних програмних моделей процесорів можна подати двома складовими: множиною конфігуровних програмних моделей, описаних мовою опису апаратних засобів, та засобами їх конфігурування. Конфігурування виконується відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора [11, 12].

Аналогічно до генераторів на основі бібліотек засобам генерування на основі конфігуровних програмних моделей процесорів також притаманна природна властивість функціональної обмеженості. Це пояснюється тим, що розробити таку конфігуровну програмну модель спеціалізованого процесора, яку можна сконфігурувати для виконання довільного обчислювального алгоритму, зберігаючи при цьому апаратну орієнтацію архітектури цього процесора, є нереальною задачею.

На основі цього можна зробити такі висновки щодо відповідності генераторів на основі конфігуровних програмних моделей процесорів поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі конфігуровних програмних моделей процесорів не може бути функціонально повною.

2. Існує можливість врахування в конфігуровних програмних моделях процесорів на етапі їх створення особливостей архітектури ПЛІС реконфігуровного середовища СККС, кількості виділеного в ній місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним процесором та реконфігуровним середовищем, що дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Необхідно зазначити, що можливості мов опису апаратних засобів в частині створення конфігуровних програмних моделей є достатніми для розроблення простих пристроїв, однак для складніших пристроїв, зокрема, спеціалізованих процесорів, цих можливостей часто буває замало [16]. Саме це й стало причиною розроблення інших, ефективніших методів конфігурування, зокрема й методу на основі бібліотеки процесорних компонент.

Час генерування за допомогою генераторів на основі конфігуровних програмних моделей процесорів можна визначити за виразом:

$$t_{gen}(G_{CPM}) = t_{search} + t_{conf} + t_{out}, \quad (4)$$

де t_{search} – час пошуку конфігуровної програмної моделі в множині цих моделей; t_{conf} – час конфігурування моделі; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

6. Особливості застосування систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня

Сьогодні багато світових фірм працюють у напрямі створення систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня та пропонують свої продукти, зокрема *Celoxica*, *Impulse*, *Nallatech*, *Synfora*, *Synopsys*, *Intron*, *Mentor Graphics* [8–10]. Принциповою відмінністю таких систем від генераторів іншого типу є те, що вхідними даними для них є не конфігураційні коди, що задають параметри процесора, а високорівневий алгоритмічний опис

роботи процесора у вигляді програми мовою високого рівня. Більшість засобів використовують для цього модифікований варіант мови C, наприклад, *Mentor Graphics* використовує *IEEE standard System C*, *Celoxica – Handle C*, *Nallatech – DIME C*, *Impulse – Impulse-C*, *Mitrionics – Mitrion-C*.

У зв'язку з тим, що в СККС високорівневий алгоритмічний опис роботи процесора формується системою розподілу обчислювального навантаження, застосування систем автоматизованого високорівневого проектування програмних моделей процесорів в СККС ставить умови використання стандартної мови *ANSI C* як вхідної. З відомих систем цій вимозі сьогодні відповідають системи *Chameleon* фірми *Intron*, *CHiMPS* фірми *Xilinx* та *Carte* фірми *SRC*.

Архітектурні підходи, покладені в основу побудови різних систем високорівневого автоматизованого проектування, значною мірою визначають продуктивність генерованих ними спеціалізованих процесорів. При цьому розробники деяких систем наголошують на високому ступені адаптованості кінцевого процесора до алгоритму, що дає змогу значно прискорити виконання задачі та коригувати співвідношення «продуктивність – затрати обладнання», при цьому не висуваючи вимог до кодів вхідних програм. Розробники інших систем ставлять до написання коду вхідної програми ряд вимог, виконання яких дають змогу досягти вищих показників продуктивності. Засоби ще інших виробників орієнтовані на роботу з алгоритмами визначених галузей (цифрова обробка сигналів, криптографія тощо) і забезпечують високу продуктивність кінцевих процесорів для виконання завдань саме у цих галузях.

На основі сказаного вище можна зробити такі висновки щодо відповідності систем автоматизованого високорівневого проектування програмних моделей процесорів сформованим в роботі [14] вимогам до системи генерування:

1. Системи автоматизованого високорівневого проектування програмних моделей процесорів можуть згенерувати модель спеціалізованого процесора для виконання довільного обчислювального алгоритму, що описується мовою програмування високого рівня, тому є функціонально повними.

2. Можливість врахування особливостей архітектури ПЛІС реконфігуровного середовища СККС у генерованих програмних моделях процесорів залежить від особливостей конкретної системи автоматизованого високорівневого проектування. За інформацією авторів, таку можливість сьогодні надає система *CHiMPS* фірми *Xilinx*, вихідний *VHDL*-код якої оптимізований для ПЛІС цієї фірми; засоби інших фірм такої можливості не надають. Разом з тим, жодна з відомих систем проектування не враховує особливостей інтерфейсу між універсальним комп'ютером та ПЛІС.

3. Продуктивність створених спеціалізованих процесорів залежить від архітектурних підходів, покладених в основу побудови конкретної системи автоматизованого високорівневого проектування. Зрозуміло, що за умови застосування вдалих архітектурних рішень машинна оптимізація архітектури обчислювального пристрою з метою досягнення ним максимальної продуктивності потенційно є ефективнішою ніж виконана інженером оптимізація.

4. Деякі засоби автоматизованого високорівневого проектування дають можливість коригувати співвідношення «продуктивність – затрати обладнання». Ефективне використання обладнання ПЛІС реконфігуровного середовища СККС, що є однією з вимог до системи генерування, викликає необхідність розроблення та реалізації методу взаємодії системи генерування на основі засобів автоматизованого високорівневого проектування з засобами логічного синтезу процесорів. Цей метод повинен забезпечити можливість налаштування системи генерування на таке співвідношення «продуктивність – затрати обладнання», у результаті якого створений спеціалізований процесор займатиме максимум з виділеного в ПЛІС місця під його реалізацію.

Відповідно до описаної в праці [17] послідовності виконання етапів автоматичного синтезу спеціалізованих процесорів від алгоритму до рівня міжрегістрових передач час генерування за допомогою систем автоматизованого високорівневого проектування програмних моделей процесорів можна визначити виразом:

$$t_{gen}(HLDS) = t_{IMCT} + t_{ASA} + t_{SPOS} + t_{out}, \quad (5)$$

де t_{IMCT} – час перетворення високорівневого опису вхідного алгоритму в проміжний код, який надає можливість подальшого опрацювання структури алгоритму; t_{ASA} – час визначення та аналізу структури алгоритму; t_{SPOS} – час виконання оптимізаційного синтезу схеми спеціалізованого процесора; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

7. Порівняльний аналіз засобів генерування програмних моделей спеціалізованих процесорів у системі генерування

Аналізуючи описані виразами (2)–(5) часові характеристики розглянутих засобів генерування, бачимо, що мінімальним часом виконання генерування характеризуються генератори на основі бібліотек програмних моделей процесорів, оскільки їхній час генерування $t_{gen}(G_{LPPM})$ визначається як сума двох величин: часу t_{search} пошуку в бібліотеці та часу t_{out} подачі файлів програмної моделі процесора на вихід. Дещо більшим є час роботи генераторів на основі бібліотек програмних моделей процесорних компонент $t_{gen}(G_{LPCPM})$, оскільки, крім зазначених складових, він містить час $t_{comp+TLGen}$ компонування моделі процесора та генерування файлів опису верхнього рівня.

Час генерування за допомогою генераторів на основі конфігурованих програмних моделей процесорів $t_{gen}(G_{CPM})$ є співмірним з часом генерування за допомогою генераторів на основі бібліотек програмних моделей процесорних компонент $t_{gen}(G_{LPCPM})$, оскільки він по суті, відрізняється лише однією складовою, а саме – замість часу $t_{comp+TLGen}$ компонування моделі процесора та генерування файлів опису верхнього рівня містить час t_{conf} виконання конфігурування моделі. Зрозуміло, що тривалість цих часів є одного порядку.

На відміну від розглянутих, час генерування за допомогою систем автоматизованого високорівневого проектування програмних моделей процесорів $t_{gen}(HLDS)$ визначається зовсім іншими складовими, за винятком часу t_{out} подачі програмної моделі процесора на вихід системи генерування. Ці складові відображають тривалість виконання завдань перетворення високорівневого опису вхідного алгоритму в проміжний код, визначення та аналізу структури алгоритму та виконання оптимізаційного синтезу схеми спеціалізованого процесора. Зрозуміло, що ці завдання по своїй суті є значно складнішими від завдання пошуку в бібліотеці чи генерування файлів опису верхнього рівня. Очевидно, що час $t_{gen}(HLDS)$ є більшим, ніж час роботи засобів генерування інших типів, що можна записати виразом:

$$t_{gen}(HLDS) > t_{gen}(G_{LPCPM}) \approx t_{gen}(G_{CPM}) > t_{gen}(G_{LPPM}). \quad (6)$$

На основі проведеного вище аналізу особливостей застосування засобів генерування програмних моделей процесорів сформуємо таблицю відповідності цих засобів вимогам до системи генерування. Також врахуємо значення часу роботи цих засобів, присвоївши їм швидкісні індекси від найшвидшого (1) до найповільнішого (3) згідно з виразом (6) (таблиця).

Відповідність засобів генерування програмних моделей процесорів вимогам до системи генерування

Тип засобів генерування програмних моделей процесорів*	1	2	3	4
Вимога до системи генерування				
Функціональна повнота	–	–	–	+
Врахування особливостей архітектури ПЛІС	+	+	+	–/+**
Врахування особливостей інтерфейсу між універсальним комп'ютером та ПЛІС	+	+	+	–
Максимальна на даному типі ПЛІС продуктивність процесора	+	+	+	–/+**
Максимальна ефективність використання обладнання ПЛІС	+	+	+	–/+***
Індекс часу виконання генерування	1	2	2	3

* 1 – на основі бібліотек програмних моделей процесорів.

* 2 – на основі бібліотек програмних моделей процесорних компонент.

* 3 – на основі конфігурованих програмних моделей процесорів.

* 4 – системи автоматизованого високорівневого проектування програмних моделей процесорів.

** – позитивно тільки для засобів *CHiMPS* фірми *Xilinx*.

*** – позитивно тільки для засобів, які дають можливість коригування співвідношення «продуктивність – затрати обладнання».

Проаналізуємо дані таблиці.

Система генерування СККС повинна бути функціонально повною. З даних таблиці видно, що єдиним типом засобів генерування програмних моделей процесорів, які характеризуються функціональною повнотою, є системи автоматизованого високорівневого проектування. Більше того, цей тип засобів є єдиним, де функціональні характеристики генерованого процесора задаються описом алгоритму його роботи мовою програмування високого рівня, що є необхідною умовою застосування в СККС. Тому як головний елемент системи генерування доцільно використовувати систему автоматизованого високорівневого проектування. Разом з тим, з даних таблиці видно, що цей тип засобів генерування переважно не відповідає вимогам в частині архітектури та в частині технічних характеристик процесорів; цим вимогам відповідають генератори на основі бібліотек та конфігурованих програмних моделей.

З огляду на це для побудови системи генерування доцільно використовувати декілька типів засобів генерування, що дасть можливість досягти максимального задоволення вимог до неї, зберігаючи при цьому її ортогональність. Крім того:

1. Для виконання часто вживаних та критичних за часом виконання обчислень та прикладних алгоритмів доцільно використати наперед розроблені, добре оптимізовані програмні моделі процесорів, які можна отримати з використанням генераторів на основі бібліотек;

2. Програмні моделі процесорів в бібліотеках доцільно представляти на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС;

3. Використання генераторів на основі конфігурованих програмних моделей процесорів не виглядає доцільним з огляду на те, що вони не мають жодних переваг над генераторами на основі бібліотек процесорних компонент, але поступаються їм за ефективністю конфігурування;

4. Оскільки функціональні характеристики процесорів для генераторів на основі бібліотек програмних моделей процесорних компонент задаються не програмою, а конфігураційними кодами, взаємодію такого генератора із системою розподілу обчислювального навантаження необхідно організувати шляхом прямої передачі команд від користувача до системи генерування.

Висновки

1. Проаналізовано засоби генерування програмних моделей процесорів на основі бібліотек програмних моделей процесорів та процесорних компонент, на основі конфігурованих програмних моделей процесорів, а також систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня в контексті їх застосування в СККС для побудови системи генерування. Досліджено відповідність цих засобів вимогам до системи генерування в частині функціональної повноти, в частині технічних характеристик моделей процесорів та в частині архітектури моделей процесорів.

2. Встановлено, що єдиним типом засобів генерування програмних моделей процесорів, які характеризуються функціональною повнотою, є системи автоматизованого високорівневого проектування, а також те, що цей тип засобів є єдиним, в якому функціональні характеристики генерованого процесора задаються описом алгоритму його роботи мовою програмування високого рівня, що є необхідною умовою застосування в СККС.

3. Встановлено, що вимогам в частині архітектури та в частині технічних характеристик процесорів відповідають генератори на основі бібліотек програмних моделей процесорів та процесорних компонент, а також на основі конфігурованих програмних моделей процесорів. При цьому програмні моделі процесорів в бібліотеках доцільно представляти на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС, що дасть змогу скоротити та спростити процес компіляції програми в СККС, уникнувши етапу логічного синтезу процесорів.

4. Досліджено часові характеристики засобів генерування програмних моделей процесорів, в результаті якого встановлено, що мінімальним часом генерування характеризуються генератори на основі бібліотек програмних моделей процесорів, дещо більшим – генератори на основі бібліотек програмних моделей процесорних компонент та на основі конфігурованих програмних моделей процесорів, і найбільшим – системи автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня.

5. Встановлено, що використання генераторів на основі конфігурованих програмних моделей процесорів є недоцільним з огляду на те, що вони не мають жодних переваг над генераторами на основі бібліотек процесорних компонент, але поступаються їм за ефективністю конфігурування.

1. Мельник А., Мельник В. *Технологія проектування ядер комп'ютерних пристроїв* // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні системи та мережі". – 2002. – № 463. – С. 3–9.
2. IEEE, *Standard VHDL Language Reference Manual. Standard 1076-1993*, New York, NY: IEEE, 1993.
3. IEEE, *Standard Verilog Hardware Description Language Reference Manual. Standard 1364-1995*, New York, NY: IEEE
4. А.О. Мельник, В.А. Мельник, Сарайрех З. Використання реконфігурованих прискорювачів для підвищення продуктивності персональних комп'ютерів // Науковий вісник Чернівецького ун-ту "Комп'ютерні системи та компоненти". – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. – Т. 1. – Вип. 1. – С. 20–25.
5. Мельник В., Сарайрех З. Самоконфігуровані апаратні прискорювачі обчислень в комп'ютерах // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні системи та мережі". – 2010. – № 688. – С.163–171.
6. Мельник В., Сарайрех З. Метод самоконфігурування апаратного прискорювача // *Мат. V Міжн. конф. молодих вчених "Комп'ютерні науки та інженерія 2011" (CSE-2011)*, 24–26 листопада 2011, Україна. – Львів. – С. 126–127.
7. Melnyk A. *IP Cores Generators in SoC Design* / A. Melnyk, W. Melnyk // *Proceeding of the 5th international Conference for Students and Young Scientists „Telecommunication in XXI Century“*, Poland, Wólka Milanowska, 24–26 November 2005. – P. 23–28.
8. Мельник А.О., Сало А.М., Клименко В., Цигилик Л., Юрчук А. ХАМЕЛЕОН – система високорівневого синтезу спеціалізованих процесорів // *Наук.-техн. журнал Нац. аерокосм. ун-ту ім. М.С. Жуковського «Харківський авіаційний інститут»*. – 2009. – № 5. – С. 189–195 с.
9. *Accelerate Software Algorithms on FPGAs*. [Електронний ресурс] / – Режим доступу: <http://www.nallatech.com>
10. *ANSI-C to VHDL Compiler*. [Електронний ресурс] / – Режим доступу: <http://www.nallatech.com/FPGA-Development-Tools/dimetalk.html>
11. *Xilinx Core Generator*. Xilinx Inc [Електронний ресурс] / – Режим доступу: http://www.xilinx.com/ise/products/coregen_overview.pdf – 2005.
12. *A Proven EDA Solutions Provider makes all the difference*. [Електронний ресурс] – Режим доступу: <http://www.aldec.com/en/>
13. Мельник А. Організація бібліотек ядер стандартизованих та замовних комп'ютерних пристроїв для високопродуктивних реконфігурованих прискорювачів / А. Мельник, В. Мельник // *IV Всеукр. наук.-практ. конф. «Комп'ютерні технології: наука і освіта»*, Україна, м. Луцьк, 9–11 жовтня 2009 р., Луцький інститут розвитку людини Університету «Україна». – С. 113–117.
14. Мельник В., Сарайрех З. Вимоги до системи генерування моделей процесорів самоконфігурованого апаратного прискорювача // *Мат. 5-ої Міжнар. наук.-техн. конф. «Сучасні комп'ютерні системи та мережі: розробка та використання» (ASCN-2011)*, 29 вересня – 01 жовтня 2011, Україна, Львів. – С. 255–258.
15. Melnyk V. *Set of Symmetric Block Cipheryng Soft-Cores* / V. Melnyk // *Proceedings of the VII-th International Conference CADSM 2003*, , Lviv-Slavsko., February 18–22, 2003. – P. 190–193.
16. Мельник В.А. Методи конфігурування моделей спеціалізованих процесорів / В.А. Мельник, Мохаммад Аль Хабабсах. // *Мат. IV Всеукр. наук.-практ. конф. «Комп'ютерні технології: наука і освіта»*, Україна, Луцьк, 9–11 жовтня 2009. – С. 121–125.
17. Мельник А.О. Автоматичний синтез спеціалізованих процесорів від алгоритму до рівня міжрегістрових передач / А.О. Мельник // *Мат. пробл.-наук. міжгал. Конф. «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління (ПНМК – 2010)»*, Бучач, 2010. – С. 27–29.
18. Мельник В., Степанов В., Сарайрех З. Система розподілу обчислювального навантаження між хост-комп'ютером та самоконфігурованим прискорювачем // *Наук. вісник Чернівецького університету імені Юрія Федьковича "Комп'ютерні системи та компоненти"*. – Чернівці, 2012. – Т. 3. – Вип. 1. – С. 6–16.

ЗАСТОСУВАННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ ПРЯМОГО ПОШИРЕННЯ СИГНАЛУ ДЛЯ РЕКОНСТРУКЦІЇ ЗОБРАЖЕНЬ БІПОЛЯРНОЇ ІМПЕДАНСНОЇ ТОМОГРАФІЇ

© Промович Ю., 2012

Для реконструкції зображень розподілу провідності за даними томографічного експерименту біполярної імпедансної томографії (БІТ) синтезовано структуру штучної нейронної мережі (ШНМ) прямого поширення сигналу, вхідним вектором якої є дані томографічного експерименту, а вектором виходу – зображення розподілу провідності. Шляхом комп'ютерного імітаційного моделювання досліджено метод реконструкції із застосуванням нейронної мережі.

Ключові слова: імпедансна томографія, штучна нейронна мережа, реконструкція, математична модель.

The artificial neuron network (ANN) of direct extension signal was used for the conductivity distribution image reconstruction from the bipolar impedance tomography (BIT) data. An input vector of network contains tomographic experiment data and a target vector is a conductivity distribution image. The neuron network reconstruction method by a computer imitation design is investigated.

Key words: impedance tomography, artificial neuron network, reconstruction, mathematic model, imitation design.

Вступ

Імпедансна томографія – метод візуалізації внутрішньої структури провідного тіла за поміряними потенціалами на електродах, що розміщені на його поверхні. Оскільки біологічні тканини мають різні значення електричних властивостей, зокрема питомої електричної провідності, а також те, що такі патологічні стани, як крововиливи та новоутворення призводять до зміни електричних властивостей тканин в зоні їх виникнення, то імпедансна томографія з врахуванням низької інвазивності може доповнювати методи скринінг-діагностики [1]. Імпедансну томографію за способом проведення вимірювань можна поділити на тетра- (ТІТ) та біполярну (БІТ) [2]. У ТІТ для одного вимірювання використовуються дві пари електродів – електроди прикладення струму і електроди вимірювання спаду напруги. У БІТ і прикладання струму, і вимірювання напруги проводяться з використанням однієї пари електродів.

Формулювання проблеми

Оскільки БІТ чутлива до контактного опору електродів [1] і математична модель, якою описується взаємодія електричного струму з провідним середовищем, для БІТ не дає змоги

використовувати формальні методи реконструкції, актуальним є розвиток БІТ, в якій використовуються інтелектуальні методи реконструкції, зокрема метод штучних нейронних мереж (ШНМ). Нейромереві методи для розв'язування задачі реконструкції зображень ТІТ використано зокрема у [3–6]. У [6] показано, що реконструкція зображень в електроімпедансній томографії потребує розв'язання нелінійної зворотної задачі, вхідні дані якої містять шум. Ця задача є некоректно поставленою і для свого розв'язання потребує застосування регуляризації, базованої на апріорних відомостях, які часто отримати неможливо. Для БІП роботи з вивчення застосування нейронних мереж для реконструкції зображень розподілу провідності не проводили.

Мета роботи

Відомі методи реконструкції зображень розподілу провідності в БІТ [7, 8] використовують інтегрування зворотних фільтрованих проєкційних даних, що призводить до накопичення похибки. При цьому використовують математичні методи попереднього опрацювання даних томографічного експерименту [9]. Мета роботи – застосувати для реконструкції зображення розподілу провідності метод нейронних мереж та дослідити чутливість методу до шумів у вхідних даних як при навчанні ШНМ, так і при її застосуванні.

Виклад основного матеріалу

1. Синтез ШНМ. Для того, щоб застосувати нейронну мережу для розв'язання прикладної задачі, потрібно: вибрати архітектуру мережі, сформувати навчальну базу даних, навчити мережу на основі отриманої бази даних. Розглянемо кожен з етапів реалізації нейронної мережі.

Вибір архітектури мережі. ШНМ є математичною структурою, яка реалізує представлення вхідних даних задачі у вихідні – її розв'язок. Найменшим структурним елементом нейронної мережі є нейрон. Використана в роботі модель нейрона – формальний нейрон. Його схему показано на рис. 1.

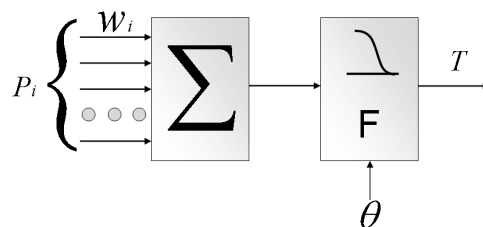


Рис. 1. Схема формального нейрона [10]

Формальний нейрон складається з зваженого суматора, вихід якого подається на елемент з нелінійною функцією активації F , для якої встановлено пороговий рівень θ . Формальний нейрон може мати безліч входів. Кожен вхідний сигнал P_i перед сумуванням множиться на коефіцієнт w_i . Один нейрон має лише один вихід T . Сигнал на виході формального нейрона можна подати у вигляді виразу:

$$T = F \left(\sum_i P_i w_i - \theta \right). \quad (1)$$

Шляхом об'єднання входів та виходів окремих нейронів формують мережі різних топологій. Для розв'язання задачі реконструкції застосовано найпростішу топологію мережі лінійного поширення сигналу – багатошаровий перцептрон. Вибір такої топології зумовлений тим, що задача реконструкції є статичною, тобто результат реконструкції повністю визначатиметься вихідними даними одного томографічного експерименту [10]. При цьому перцептрон здатний апроксимувати будь-яку передавальну характеристику. У перцептрона нейрони організовані у шари (див. рис. 2). Нейрони кожного наступного шару своїми входами пов'язані із виходами нейронів попереднього за принципом «кожен з кожним». У межах одного шару нейрони мають найчастіше одну функцію активації F .

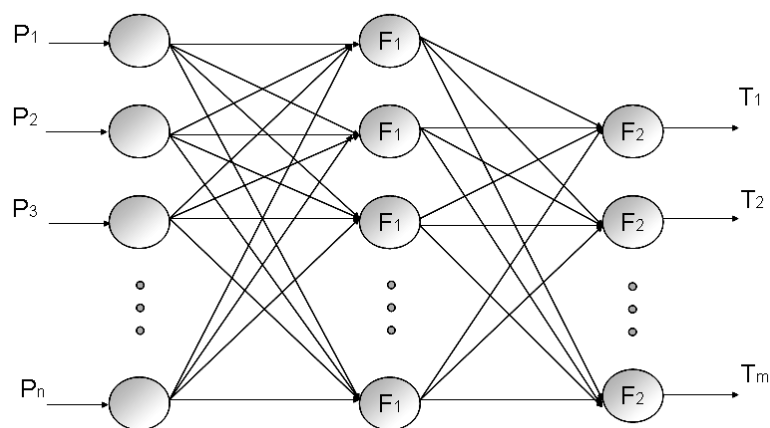


Рис. 2. Багатошаровий перцептрон

Кількість нейронів першого шару мережі відповідає кількості вхідних даних P_n . Дані одного томографічного експерименту подамо у вигляді одновимірного масиву, кожен з елементів якого є спадом напруги на відповідній парі вимірювальних електродів (див. рис. 3). У цій роботі сигнал томографічного експерименту містив 512 вимірів (на рис. 3 показано графік частини сигналу), тому кількість нейронів у вхідному (сенсорному) шарі ШНМ $P_n = 512$.

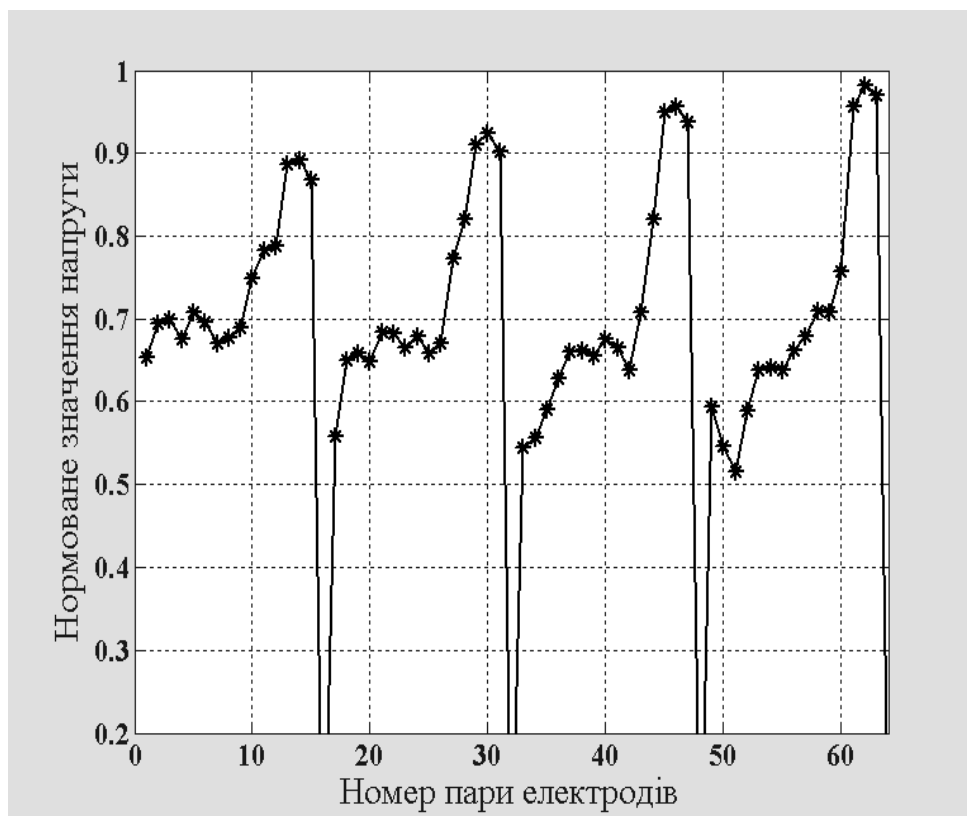


Рис. 3. Дані томографічного експерименту для 32-електродної вимірювальної системи томографа

Потрібно також з'ясувати, який зміст вкладатиметься в вихідний вектор цілей T . Результатом роботи мережі має бути зображення, яке відобразить форму тіла та розподіл провідності всередині. В роботі розподіл провідності досліджуваного провідного тіла представлено моделлю скінченних елементів. Кожному з прямокутних елементів зображення, що відповідає

ділянці перерізу досліджуваного тіла, відповідає індекс m . При цьому зображення описуватиметься одновимірним масивом цілей $T_i, i = (1..m)$, кожен елемент якого – значення провідності відповідної ділянки. При проведенні імітаційного моделювання як модель провідної області використано зображення розмірністю 32×32 квадратних елементів – пікселів, з них 713 повністю лежать всередині круглої області з центром всередині зображення. Отже, вихідний шар нейронної мережі містить 713 нейронів.

Крім сенсорного та вихідного шару, нейронна мережа може містити певну кількість внутрішніх, прихованих, шарів. Їх кількість вибирається часто експериментально, хоча є роботи, зокрема [10], в яких показано, що для статичних задач встановлення залежності між вхідними та вихідними даними реалізується нейронною мережею з одним прихованим шаром. У роботі використано нейронну мережу з трьома прихованими шарами, в кожному з яких містилося 256 нейронів з логістичною функцією активації.

Робота ШНМ полягає у реалізації лінійних та нелінійних перетворень вектора входів у вектор цілей. Шляхом послідовного здійснення лінійних комбінацій та нелінійних перетворень досягається апроксимація довільної багатовимірної функції при відповідному виборі параметрів мережі. Підбирають параметри ШНМ через процедуру навчання, суть якої полягає в покроковій зміні вагових коефіцієнтів та порогових рівнів нейронів мережі для досягнення якнайкращої відповідності (задається параметром) очікуваного результату роботи мережі від фактичного.

2. Формування навчальної бази. Навчання нейронної мережі проводилось з використанням даних комп'ютерного імітаційного моделювання. Формувалось дискретне зображення однорідного розподілу провідності розмірності 32×32 пікселів. На цьому зображенні формувалось включення у вигляді округлої зони з вищою на 10 % провідністю (див. рис. 4). Розташування включення в межах провідної області задавалось випадково.

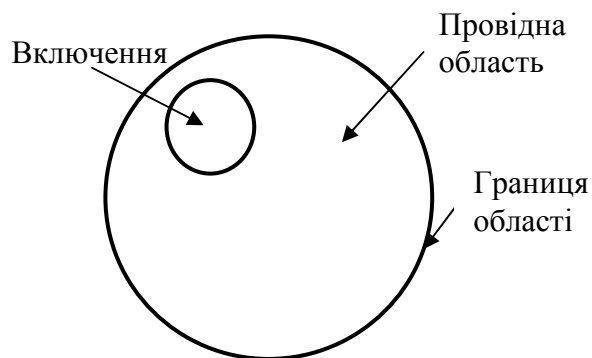


Рис. 4. Вхідні дані імітаційної моделі томографічного експерименту

Використовуючи комп'ютерну імітаційну модель БІТ [11] для кожного з тестових зображень, розв'язувалась пряма задача – пошук спадів напруг на границі області. Одновимірний масив напруг (дані томографічного експерименту) стосовно ШНМ є вхідним вектором, а одновимірний масив значень провідностей елементів тестового зображення – вектором цілей T_m . Навчання та перевірку працездатності мережі проводили на навчальній базі з 2000 пар вхідних і вихідних векторів. Для навчання ШНМ використано 90 % навчальних даних, решту – для перевірки роботи мережі. Під час навчання використовували метод зворотного розповсюдження похибки (Backpropagation Methode) [10].

3. Імітаційне моделювання. При імітаційному моделюванні ставилось завдання дослідити можливість навчання ШНМ з використанням зашумлених вхідних даних. Моделювання дасть змогу спрогнозувати можливість навчання ШНМ з використанням реальних даних, які, звичайно, міститимуть шум. При проведенні моделювання використовувались дві ситуації:

- 1) навчання ШНМ проводилось з використанням незашумлених вхідних даних. При тестуванні мережі використовувались зашумлені вхідні дані;
- 2) навчання ШНМ проводилось з використанням зашумлених вхідних даних. При тестуванні мережі також використовувались зашумлені вхідні дані.

Послідовність дій в першій ситуації така: а) формування тестових зображень розподілу провідності з випадково розташованим включенням; б) розв'язування прямої задачі БТ для кожного з тестових розподілів та отримання даних томографічного експерименту; в) навчання нейронної мережі; г) тестування нейронної мережі з використанням зашумлених вхідних даних. У другій ситуації всі вхідні дані – і навчальні, і ті, що використовувались для тренування, містили шум.

На рис. 5 показано приклад реконструкції. Ліворуч розташовано модельоване зображення, праворуч – реконструйоване ШНМ.

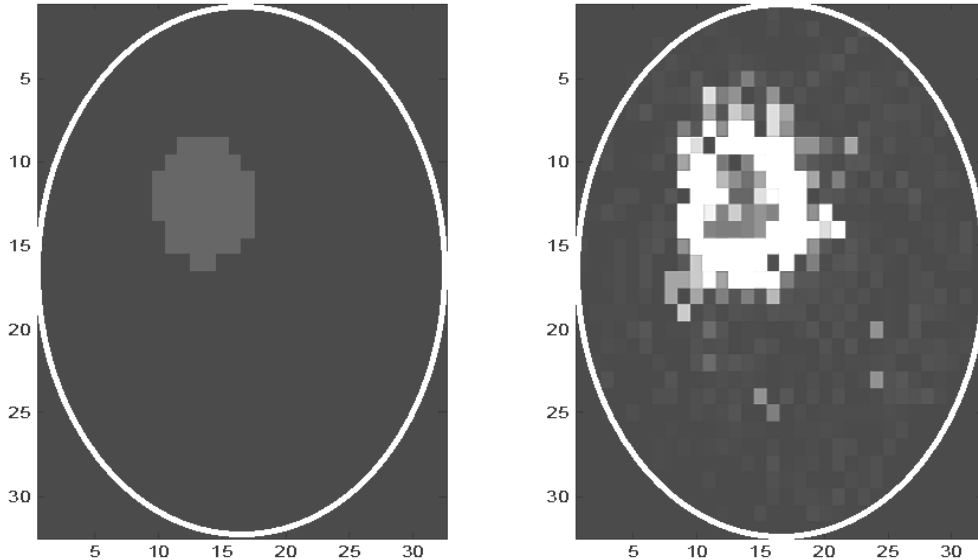


Рис. 5. Реконструкція з застосуванням нейронної мережі

Для кількісної характеристики якості реконструкції обчислено коефіцієнт кореляції між незашумленим тестовим I_T та реконструйованим I_R зображеннями.

У другій ситуації до даних томографічного експерименту (вектора входів) додано випадкову складову – нормально розподілений білий шум. Обрано такі точкові оцінки синтезованого шуму: математичне сподівання $M_\xi = 0$ та середньоквадратичне відхилення $\sigma_\xi = (0.1, 0.2, 0.3)$ – на графіку $SNR_{TRAIN} = 10, 20$ та 30% відповідно.

Результати застосування ШНМ, навченої з використанням незашумлених даних ($SNR_{TRAIN} = 0$) та даних, що містять шум, наведено на рис. 6.

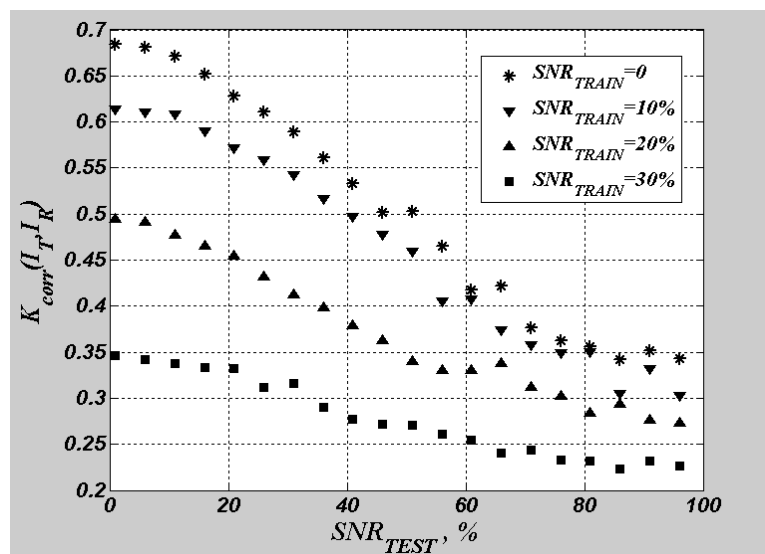


Рис. 6. Залежність коефіцієнта кореляції K_{corr} між тестовим I_T та реконструйованим I_R зображеннями від співвідношення сигнал/шум (SNR_{TEST}) вхідних даних

Найкращу якість реконструкції показала нейронна мережа, навчання якої проводили з використанням незашумлених даних. При цьому навіть за значного рівня шумів у тестових даних $SNR_{TEST} = (45 \div 50) \%$ на реконструйованому зображенні можна легко локалізувати область з підвищеною провідністю.

Висновок

1. Використання багат шарового перцептрона для розв'язування задачі реконструкції біполярної імпедансної томографії дало змогу отримати зображення розподілу провідності з коефіцієнтом кореляції 0.68.

2. Навчання ШНМ на реальних даних недоцільне, оскільки наявність шумів в останніх буде причиною зниження якості реконструкції.

3. Для навчання ШНМ потрібен синтез навчальних даних, отриманих імітаційним моделюванням з використанням детальних математичних моделей провідного тіла.

1. Пеккер Я.С. *Электроимпедансная томография* / Я.С. Пеккер, К.С. Бразовский, В.С. Усов и др. – Томск: Изд-во НТМ, 2004. – 192 с. 2. Корневский Н.А. *Проектирование электронной медицинской аппаратуры для диагностики и лечебных воздействий: Монография* / Н.А. Корневский, Е.П. Попечителев, С.А. Филист. – Курск: Курская городская типография, 1999. – 537с. 3. Netajatali A. *An Iterative Algorithm for Electrical Impedance Imaging Using Neural Networks* / A. Netajatali, I.R. Ciric // *IEEE Trans. Magn.* – Vol. 34, no. 5, pp. 2940–2943, September 1998. 4. Корженевский А.В. *Использование искусственных нейронных сетей для решения обратных задач электроимпедансной и магнитоиндукционной томографии* / А.В. Корженевский // *Журнал радиоэлектроники.* – 2001. – № 12. 5. Lampinen J. *Application of Bayesian Neural Network in Electrical Impedance Tomography* / Jouko Lampinen, Aki Vehtari, Kimmo Leinonen // *IJCNN'99. International Joint Conference on.*, – Vol. 6., pp. 3942–3947. 6. Adler, A. *A Neural Network Image Reconstruction Technique for Electrical Impedance Tomography* / A. Adler, R. Guardo // *IEEE transactions on medical imaging.* - vol. 13, no. 4, december 1994. 7. Soleimani M. *Electrical impedance tomography system: an open access circuit design [Електронний ресурс].* – режим доступу: <http://www.biomedical-engineering-online.com/content/5/1/28>. 8. Яворський Б.І. *Метод реконструкції зображень в електроімпедансній томографії* / Б.І. Яворський, Ю.Б. Промович // *Вісник ТДТУ ім. І. Пулюя.* – Т. 14. – 2009. – № 1. – С. 154–161. 9. Промович Ю.Б. *Мінімізація систематичної похибки мірювання електроімпедансного томографа* / Ю.Б. Промович // *Вісник Хмельницького національного університету.* 2010. – Т. 3. – С. 247–250. 10. Осовський С. *Нейронні мережі для обробки інформації* / С. Осовський. – М.: Фінанси й статистика, 2004. 11. Яворський Б.І. *Імітаційне моделювання прямої задачі електроімпедансної томографії* / Б.І. Яворський, Ю.Б. Промович, С.Б. Яворська // *Вісник Кременчуцького національного технічного університету імені Михайла Остроградського.* – 2011. – Вип. 3, Ч. 1. – С.49–53.

АДАПТИВНИЙ W-НЕЙРОН ТА ЙОГО НАВЧАННЯ В ЗАДАЧАХ ПРОГНОЗУВАННЯ І ВИЯВЛЕННЯ РОЗЛАДНАНЬ

© Бодянський Є., Винокурова О., 2012

Розглянуто структуру адаптивного W-нейрона та його метод навчання. Запропонований алгоритм навчання має підвищену швидкість збіжності та забезпечує покращені апроксимуючі властивості за рахунок настроювання усіх параметрів вейвлет-функцій. Введена підсистема виявлення розладнань для W-нейрона, що дає змогу розв'язувати задачі діагностування в on-line режимі стохастичних процесів.

Ключові слова: W-нейрон, методи навчання, прогнозування, виявлення розладнань.

Adaptive W-neuron and its learning algorithm are considered. Proposed learning algorithm has increased convergence rate and provides improved approximative properties because of the all wavelet parameters tuning. The fault detection subsystem for W-neuron that allows to solve a stochastic process diagnosing problems in on-line mode.

Key words: W-neuron, learning algorithm, prediction, fault detection.

Вступ

Сьогодні штучні нейронні мережі набули поширення для розв'язання великого класу задач обробки інформації – насамперед для ідентифікації, емуляції, інтелектуального керування, прогнозування часових рядів довільної природи за умов структурної та параметричної невизначеності.

Ще однією важливою задачею є виявлення розладнань у довільного типу стохастичних процесах, що тісно пов'язана з проблемою діагностування об'єктів і систем різного призначення. Останні роки характеризуються сплеском досліджень у галузі діагностуючих нейронних мереж, що засновані переважно на багат шарових архітектурах.

Альтернативою традиційним багат шаровим мережам з сигмоїдальними функціями активації є радіально-базисні нейронні мережі, що мають один прихований шар, що складається із так званих R -нейронів, при цьому навчання цих мереж реалізується на рівні вихідного шару, який складається із лінійних асоціаторів [1–5]. На відміну від P -нейронів з сигмоїдальною функцією активації, R -нейрони мають, як правило, дзвонувату функцію активації $f_j(x)$, аргументом якої є відхилення (зазвичай у метриці Евкліда) між поточним значенням сигналу на вході $x(k)$ і центром c_j – j -го нейрону. Основною перевагою радіально-базисних мереж є висока швидкість навчання у вихідному шарі, що визначається тим, що налаштовувані параметри входять до опису мережі лінійно. Водночас залишається відкритою проблема розташування центрів R -нейронів, невдале вирішення якої призводить до виникнення «прокляття розмірності».

Поряд з нейронними мережами для обробки сигналів довільної природи достатньо часто використовується вейвлет-перетворення [7–9], що забезпечує компактне локальне подання сигналів як в частотній, так і в часовій областях. На стику теорії штучних нейронних мереж і вейвлетів виникли вейвлет-нейронні мережі [10–18], що підтвердили свою ефективність в задачах обробки нестационарних нелінійних сигналів і процесів.

У статті розглянуто питання синтезу адаптивного W-нейрона та його метод (алгоритм) навчання, що має підвищену швидкість збіжності та забезпечує покращені апроксимуючі властивості в задачах прогнозування і виявлення розладнань стохастичних послідовностей.

Архітектура адаптивного W-нейрона

Розглянемо структуру адаптивного W-нейрона, наведену на рис. 1, яка збігається по суті з радіально-базисною нейронною мережею з одним виходом.

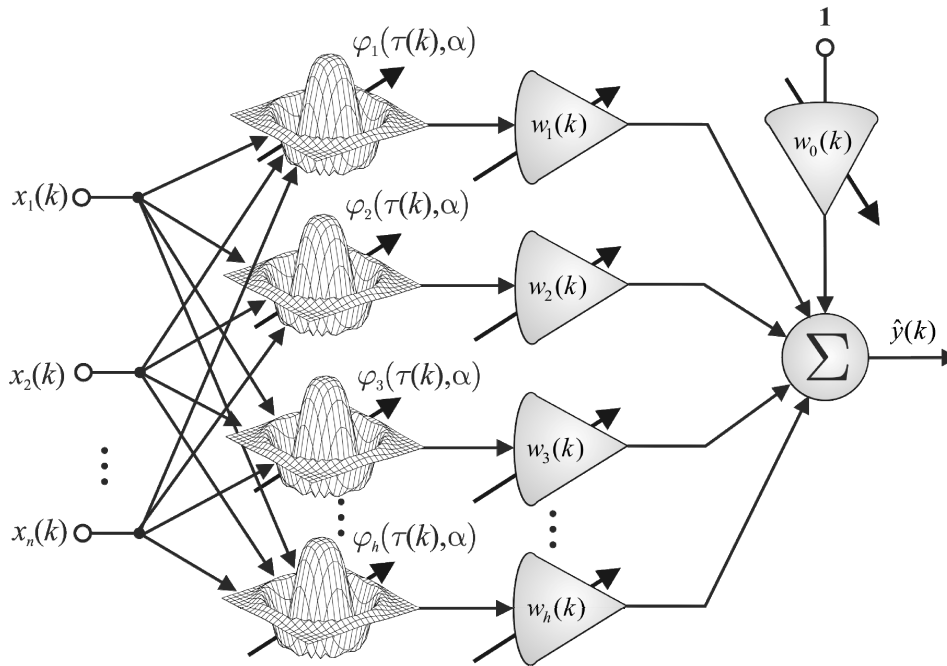


Рис. 1. Архітектура адаптивного W-нейрона

Нульовий шар архітектури є рецепторним і в поточний момент дискретного часу k на нього або подається сигнал у формі вектора $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$, або формується вектор входів за допомогою елементів затримки вхідного сигналу. Прихований шар, на відміну від радіально-базисних мереж, складається не із R -нейронів, а із вейвлонів з багатовимірними активаційними функціями-вейвлетами вигляду

$$\varphi_j(x(k)) = \varphi_j\left((x(k) - c_j)^T Q_j^{-1}(x(k) - c_j)\right), \quad j = 1, 2, \dots, h, \quad (1)$$

в яких замість параметрів ширин σ_{ji} використовується матриця Q_j^{-1} , тобто використовується не евклідова метрика, а відстань Ітакури–Сайто [19]. Це приводить до того, що рецепторні поля W-нейронів (1) можуть мати довільну орієнтацію відносно координатних осей простору X , що розширює функціональні можливості адаптивного W-нейрона.

На основі вейвлет-функції «Mexican Hat» [8] введемо налаштовувану активаційну функцію у структуру адаптивного W-нейрона, що має вигляд [20, 21]:

$$\varphi_j(x(k)) = (1 - \alpha_j \tau_j^2) \exp\left(-\frac{\tau_j^2}{2}\right), \quad (2)$$

де $\tau_j(x(k)) = \left((x(k) - c_j(k))^T Q_j^{-1}(k)(x(k) - c_j(k))\right)$, α_j – налаштовуваний параметр ($0 \leq \alpha \leq 1$).

Параметр α_j дає змогу налаштувати форму активаційної функції у процесі навчання адаптивного W-нейрона, при цьому при $\alpha = 0$ маємо гауссову функцію активації, при $\alpha = 1$ маємо вейвлет-функцію «Mexican Hat», а при $0 < \alpha < 1$ – гібридну функцію активації. На рис. 2 наведені форми двовимірної активаційної функції W-нейронів (2) при довільних матрицях Q_j^{-1} і параметрі α_j .

І, нарешті, вихід W-нейрона – це звичайний адаптивний лінійний асоціатор із синаптичними вагами w_j , що налаштовуються:

$$\hat{y}(k) = w_0 + \sum_{j=1}^h w_j \varphi\left((x(k) - c_j) Q_j^{-1} (x(k) - c_j)\right) = w^T \varphi(x(k)), \quad (3)$$

де $\varphi_0(x(k)) \equiv 1$, $w = (w_0, w_1, w_2, \dots, w_h)^T$, $\varphi(x(k)) = (1, \varphi_1(x(k)), \varphi_2(x(k)), \dots, \varphi_h(x(k)))^T$.

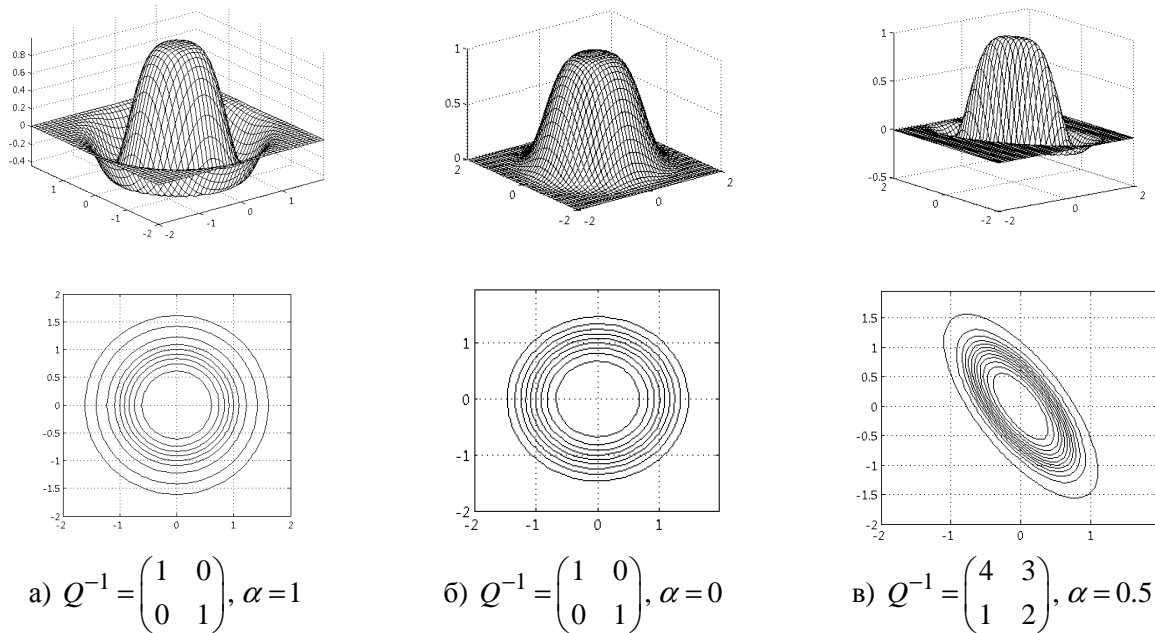


Рис. 2. Активаційні функції W-нейронів при довільних матрицях Q_j^{-1} і параметрі α

Налаштовуваними параметрами структури адаптивного W-нейрона, що підлягають визначенню в процесі навчання, є $h+1$ синаптичних ваг w_j , h ($n \times 1$) – параметрів векторів c_j , h ($n \times n$) – елементів матриць Q_j^{-1} і h параметрів α_j .

Метод навчання адаптивного W-нейрона

Оскільки $(h+1) \times 1$ вектор синаптичних ваг w входить до опису мережі лінійно, для його уточнення може використовуватися будь-який із алгоритмів адаптивної ідентифікації [22] і насамперед традиційний рекурентний метод найменших квадратів:

$$\begin{cases} w(k+1) = w(k) + \frac{P(k)(y(k) - w^T(k)\varphi(x(k)))\varphi(x(k))}{1 + \varphi^T(x(k))P(k)\varphi(x(k))}, \\ P(k+1) = P(k) - \frac{P(k)\varphi(x(k+1))\varphi^T(x(k+1))P(k)}{1 + \varphi^T(x(k+1))P(k)\varphi(x(k+1))}, \end{cases} \quad (4)$$

що має згладжувальні властивості.

Для налаштування параметрів W-нейронів (векторів c_j , матриць Q_j^{-1} , параметрів α_j) використовуватимемо градієнтну мінімізацію локального критерію навчання

$$E(k) = \frac{1}{2} e^2(k) = \frac{1}{2} (y(k) - \hat{y}(k))^2, \quad (5)$$

при цьому на відміну від покомпонентного навчання, що розглянуто у [6], будемо проводити уточнення у векторно-матричній формі, що, по-перше, простіше з обчислювального погляду, а, по-друге, дає змогу оптимізувати процес навчання за швидкодією.

У загальному випадку алгоритм навчання можна записати у вигляді

$$\begin{cases} c_j(k+1) = c_j(k) - \eta_{c_j} \nabla_{c_j} E(k), \quad j=1,2,\dots,h, \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) - \eta_{Q_j^{-1}} \left\{ \frac{\partial E(k)}{\partial Q_j^{-1}} \right\}, \quad j=1,2,\dots,h, \\ \alpha(k+1) = \alpha(k) - \eta_\alpha \nabla_\alpha E(k), \end{cases} \quad (6)$$

де $\nabla_{c_j} E$, $\nabla_\alpha E$ – $(n \times 1)$ -вектори-градієнти критерію (5) за c_j та α відповідно; $\alpha(k) = (\alpha_1(k), \alpha_2(k), \dots, \alpha_h(k))$; $\left\{ \frac{\partial E(k)}{\partial Q_j^{-1}} \right\}$ – $(n \times n)$ -матриця, що складається з частинних похідних E за компонентами Q_j^{-1} ; η_{c_j} , $\eta_{Q_j^{-1}}$ і η_α – параметри кроку алгоритму навчання.

Для функції активації (2) можна записати

$$\begin{cases} \nabla_{c_j} E(k) = 2e(k)w_j(k)Q_j^{-1}(k)(x(k) - c_j(k)) \cdot \\ \cdot \left(\alpha_j \tau_j^3(x(k)) - (2\alpha_j + 1)\tau_j(x(k)) \right) \exp(-\tau_j^2(x(k))/2) = e(k)J_{c_j}(k), \\ \left\{ \frac{\partial E(k)}{\partial Q_j^{-1}} \right\} = e(k)w_j(k)(x(k) - c_j(k))(x(k) - c_j(k))^T \cdot \\ \cdot \left(\alpha_j \tau_j^3(x(k)) - (2\alpha_j + 1)\tau_j(x(k)) \right) \exp(-\tau_j^2(x(k))/2) = -e(k)J_{Q_j^{-1}}(k), \\ \nabla_\alpha E(k) = -e(k)w(k) \odot \tau^2(x(k)) \odot \exp(-\tau^2(x(k))/2) = -e(k)J_\alpha(k). \end{cases} \quad (7)$$

де $\tau(x(k)) = (\tau_1(x(k)), \tau_2(x(k)), \dots, \tau_h(x(k)))^T$; $\tau^2(x(k)) = \tau(x(k)) \odot \tau(x(k))$; \odot – прямий (скотовий) добуток.

Тоді алгоритм навчання W-нейрона з урахуванням (7) має вигляд

$$\begin{cases} c_j(k+1) = c_j(k) - \eta_{c_j} e(k)J_{c_j}(k), \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \eta_{Q_j^{-1}} e(k)J_{Q_j^{-1}}(k), \\ \alpha(k+1) = \alpha(k) + \eta_\alpha e(k)J_\alpha(k), \end{cases} \quad (8)$$

при цьому швидкість збіжності до оптимальних значень c_j і Q_j^{-1} повністю визначається параметрами кроку η_{c_j} і $\eta_{Q_j^{-1}}$.

Підвищити швидкість збіжності можна, використовуючи складніші, ніж градієнтні, процедури типу Хартлі або Марквардта, що для налаштування параметрів c_j і α можуть бути записані в загальній формі [21]

$$\begin{cases} c_j(k+1) = c_j(k) - \lambda_c (J_{c_j}(k)J_{c_j}^T(k) + \eta_c I)^{-1} J_{c_j}(k)e(k), \\ \alpha(k+1) = \alpha(k) + \lambda_\alpha (J_\alpha(k)J_\alpha^T(k) + \eta_\alpha I)^{-1} J_\alpha(k)e(k), \end{cases} \quad (9)$$

де I – $(n \times n)$ – одинична матриця; $\lambda_c, \lambda_\alpha$ – додатні параметри; η_c, η_α – параметри регуляризації.

Використовуючи лему обернення матриць, після достатньо простих перетворень можна отримати простий і ефективний алгоритм навчання центрів W-нейронів і параметра a у вигляді

$$\begin{cases} c_j(k+1) = c_j(k) - \lambda_c \frac{e(k)J_{c_j}(k)}{\eta_c + \|J_{c_j}(k)\|^2}, \\ \alpha(k+1) = \alpha(k) + \lambda_\alpha \frac{e(k)J_\alpha(k)}{\eta_\alpha + \|J_\alpha(k)\|^2} \end{cases} \quad (10)$$

з точністю до позначення, що збігається з оптимальним (при $\lambda_c = 1, \eta_c = 0, \lambda_\alpha = 1, \eta_\alpha = 0$) за швидкодією алгоритмом Качмажа–Уїдроз–Гоффа.

Для налаштування матриць Q_j^{-1} можна скористатися матричною модифікацією алгоритму (10) у вигляді [21]

$$Q_j^{-1}(k+1) = Q_j^{-1}(k) + \lambda_{Q_j^{-1}} \frac{e(k)J_{Q_j^{-1}}(k)}{\eta_{Q_j^{-1}} + \text{Tr}(J_{Q_j^{-1}}^T(k)J_{Q_j^{-1}}(k))}, \quad (11)$$

де $\lambda_{Q_j^{-1}}$, $\eta_{Q_j^{-1}}$ мають те саме значення, що і відповідні параметри в (9).

Отже, алгоритм навчання параметрів W-нейронів прихованого шару в оптимальному за швидкістю варіанті можна записати у формі

$$\begin{cases} c_j(k+1) = c_j(k) - \|J_{c_j}(k)\|^{-2} e(k)J_{c_j}(k), \quad j=1,2,\dots,h, \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \left(\text{Tr}(J_{Q_j^{-1}}^T(k)J_{Q_j^{-1}}(k)) \right)^{-1} e(k)J_{Q_j^{-1}}(k), \quad j=1,2,\dots,h, \\ \alpha(k+1) = \alpha(k) + \|J_\alpha(k)\|^{-2} e(k)J_\alpha(k). \end{cases} \quad (12)$$

Відомо, що однокрокові алгоритми типу Качмажа, маючи високу швидкість, не мають фільтрувальних властивостей, тобто погано працюють за умов інтенсивних збурень та завад. Для надання процесу навчання згладжувальних властивостей використовують підхід, запропонований у [24], і можна ввести таку процедуру навчання, включаючи і вектор синаптичних ваг:

$$\begin{cases} w(k+1) = w(k) - \lambda_w \frac{e(k)J_w(k)}{\eta_w(k)}, \quad \eta_w(k+1) = \gamma_w \eta_w(k) + \|J_w(k+1)\|^2, \\ c_j(k+1) = c_j(k) - \lambda_{c_j} \frac{e(k)J_{c_j}(k)}{\eta_{c_j}(k)}, \quad \eta_{c_j}(k+1) = \gamma_{c_j} \eta_{c_j}(k) + \|J_{c_j}(k+1)\|^2, \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \lambda_{Q_j^{-1}} \frac{e(k)J_{Q_j^{-1}}(k)}{\eta_{Q_j^{-1}}(k)}, \quad \eta_{Q_j^{-1}}(k+1) = \gamma_{Q_j^{-1}} \eta_{Q_j^{-1}}(k) + \text{Tr}\left(J_{Q_j^{-1}}^T(k+1)J_{Q_j^{-1}}(k+1) \right), \\ \alpha(k+1) = \alpha(k) + \lambda_\alpha \frac{e(k)J_\alpha(k)}{\eta_\alpha(k)}, \quad \eta_\alpha(k+1) = \gamma_\alpha \eta_\alpha(k) + \|J_\alpha(k+1)\|^2, \end{cases} \quad (13)$$

(тут $0 \leq \gamma_w \leq 1$, $0 \leq \gamma_{c_j} \leq 1$, $0 \leq \gamma_{Q_j^{-1}} \leq 1$, $0 \leq \gamma_\alpha \leq 1$ – параметри зважування застарілої інформації), яка є нелінійним гібридом алгоритму Качмажа–Уїдроу–Гоффа і Гудвіна–Ремеджа–Кейнеса і має як слідкувальні, так і фільтрувальні властивості.

Підсистема виявлення розладів у стохастичних послідовностях

Алгоритм навчання (13) призначено для налаштування усіх параметрів адаптивного W-нейрона в задачах обробки нестационарних зашумлених сигналів, при цьому що більше значення параметрів зважування γ , то яскравіші згладжувальні властивості, а наявність різких змін потребує зменшення значення γ – скорочення пам'яті алгоритму.

У зв'язку з цим уявляється доцільним введення в прогнозувальну систему на основі адаптивного W-нейрона додаткової підсистеми виявлення розладів, що дає змогу у on-line режимі виявляти різкі зміни у контрольованому сигналі. Для цього можна скористатися діагностуючим алгоритмом Т. Хеглунда [25], що у прийнятих нами позначеннях набуває вигляду

$$\begin{cases} \theta(k+1) = \lambda_\theta \theta(k) + w(k+1) - w(k) = \lambda_\theta \theta(k) + \Delta w(k+1), \quad 0 \leq \lambda_\theta < 1, \\ T(k+1) = \text{sign}(\theta^T(k+1)\Delta w(k+1)). \end{cases} \quad (14)$$

При цьому поява на декількох тактах контролю підряд значення діагностуючого сигналу $T(k+1)=1$ свідчить про розлад. У цьому випадку значення параметрів зважування застарілої

інформації повинні бути зменшені на деяке значення $\Delta\gamma$. Якщо ж діагностувальний сигнал $T(k+1)$ змінює свої знаки, то параметри зважування можуть бути збільшені, що, своєю чергою, покращує фільтрувальні властивості алгоритму.

Структурну схему підсистеми виявлення розладів з W-нейроном на основі алгоритму (14) наведено на рис. 3.

Задачу прогнозування та виявлення розладів у стохастичних послідовностях за допомогою підходу, що розвивається, можна розв'язати на основі адаптивного W-нейрона, у якого нульовий шар утворюється лінією елементів чистого запізнювання z^{-1} (як показано на рис. 3), а поточні значення синаптичних ваг, що обчислюються у вихідному шарі, подаються у підсистему виявлення розладів, що керує зміною параметрів зважування застарілої інформації.

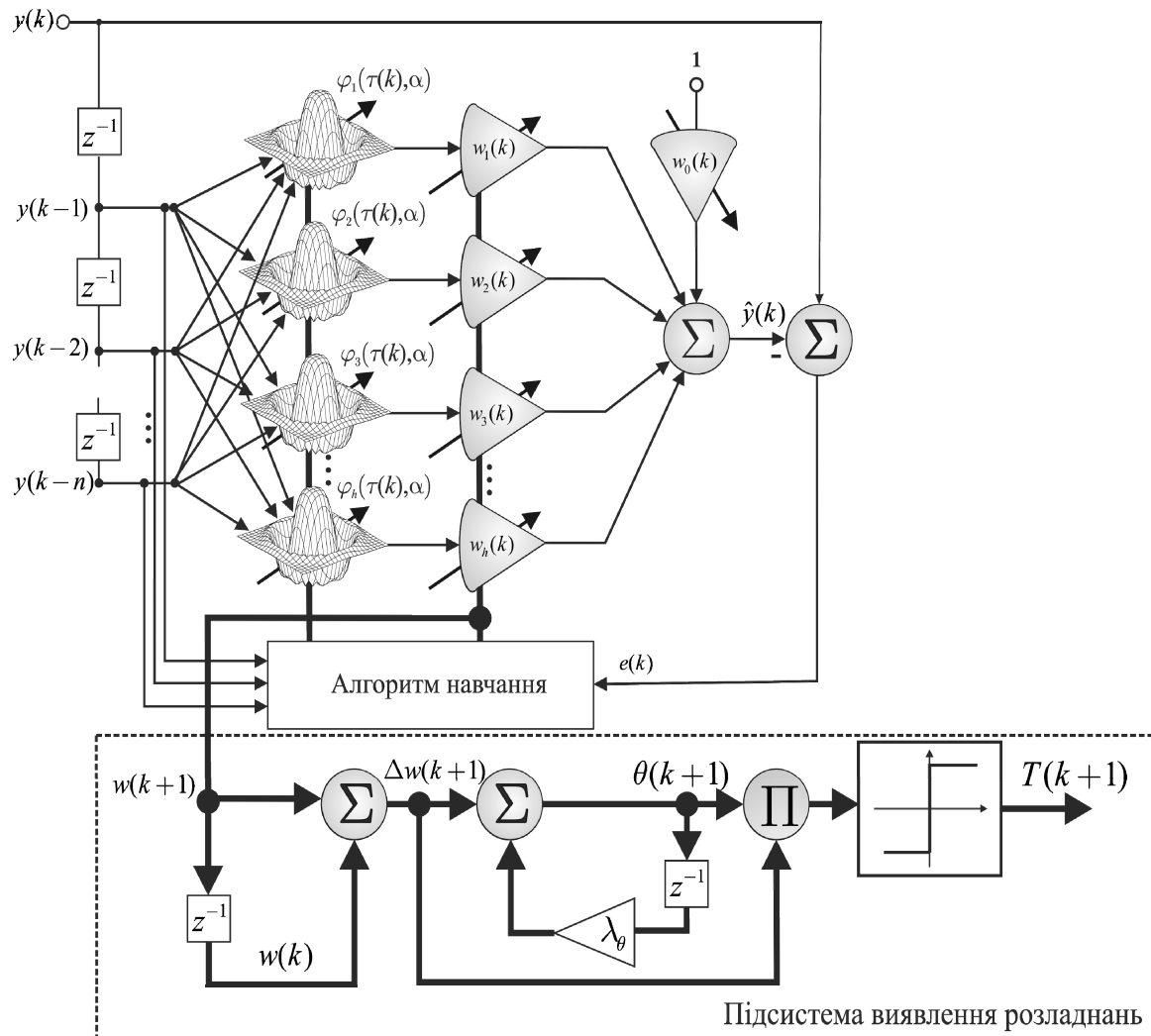


Рис. 3 – Структура W-нейрона з діагностувальною підсистемою

Висновки

Запропоновано простий та ефективний алгоритм навчання адаптивного W-нейрона, що має як слідкувальні, так і фільтрувальні властивості і дає змогу у on-line режимі обробляти нестационарні нелінійні сигнали та процеси. Структуру адаптивного W-нейрона можна використати і як самостійну мережу, і як елемент складнішої нейро-фаззії системи. Налаштуванням усіх параметрів W-нейрона покращено апроксимувальні властивості мережі. Для розв'язання задачі виявлення розладів в on-line режимі у стохастичних послідовностях до структури W-нейрону введено діагностувальну підсистему. Імітаційне моделювання підтверджує ефективність запропонованого підходу порівняно з відомими системами прогнозування та діагностування.

1. Moody J., Darken C. J. Fast learning in networks of locally-tuned processing units // *Neural Computation*. – 1989. – 1. – P. 281–294. 2. Park J., Sandberg I. W. Universal approximation using radial-basis-function networks // *Neural Computation*. – 1991. – 3. – P. 246–257. 3. Leonard J. A., Kramer M. A., Ungar L. H. Using radial basis functions to approximate a function and its error bounds // *IEEE Trans. on Neural Networks*. – 1992. – 3. – P. 614–627. 4. Sunil E. V. T., Yung C. Sh. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems // *IEEE Trans. on Neural Networks*. – 1994. – 5. – P. 594–603. 5. Poggio T., Girosi F. A Theory of Networks for Approximation and Learning // A. I. Memo № 1140, C.B.I.P. Paper № 31. – Massachusetts Institute of Technology, 1994. – 63 p. 6. Bishop C. M. *Neural Networks for Pattern Recognition*. – Oxford: Clarendon Press, 1995. – 482 p. 7. Chui C. K. *An Introduction to Wavelets*. – New York: Academic, 1992. – 264 p. 8. Daubechies I. *Ten Lectures on Wavelets*. – Philadelphia, PA: SIAM., 1992. – 228 p. 9. Meyer Y. *Wavelets: Algorithms and Applications*. – Philadelphia, PA: SIAM., 1993. – 133 p. 10. Billings S. A., Wei H.-L. A new class of wavelet networks for nonlinear system identification // *IEEE Trans. on Neural Networks*. – 16 (4). – 2005. – P. 862–874. 11. Zhang Q. H., Benveniste A. Wavelet networks // *IEEE Trans. on Neural Networks*. – 3 (6). – 1992. – P. 889–898. 12. Zhang J., Walter G. G., Miao Y., Lee W. N. W. Wavelet neural networks for function learning // *IEEE Trans. on Signal Process.* – 43(6). – 1995. – P. 1485–1497. 13. Zhang Q. H. Using wavelet network in nonparametric estimation // *IEEE Trans. on Neural Networks*. – 8(2). – 1997. – P. 227–236. 14. Bodyanskiy Ye., Pavlov O., Vynokurova O. Outliers resistant learning algorithm for radial-basis-fuzzy-wavelet-neural network in stomach acute injury diagnosis tasks // Eds. by K.Markov, K.Ivanova, I.Mitov. – International Book Series “Information Science and Computing”, Number 2. – Sofia: Institute of Information Theories and Application FOI ITHEA, 2008. – P.55–62. 15. Bodyanskiy Ye., Pliss I., Vynokurova O. Adaptive wavelet-neuro-fuzzy network in the forecasting and emulation tasks // *Int. Journal on Information Theory and Applications*. – 2008. – V. 15. – 1. – P. 47–55. 16. Bodyanskiy Ye., Vynokurova O. Robust learning algorithm for wavelet-neural fuzzy network based on Polywog wavelet // *Системные технологии*. – 2008. – Т.2. – № 3(56). – С. 129–134. 17. Бойко В. В., Бодянский Е. В., Винокурова Е. А., Сушков С. В., Павлов А. А. Анализ клинических данных в медицинских исследованиях на основе методов вычислительного интеллекта. – Харьков: ТО Эксклюзив, 2008. – 120 с. 18. Reyneri L. M. Unification of neural and wavelet networks and fuzzy systems // *IEEE Trans. on Neural Networks*. – 1999. – 10. – P. 801-814. 19. Itakura F. Maximum prediction residual principle applied to speech recognition // *IEEE Trans. on Acoustics, Speech, and Signal Processing*. – 1975. – 23. – P. 67–72. 20. Бодянский Е.В., Винокурова Е.А. Составной адаптивный вэйвлон и алгоритм его обучения // *Управляющие системы и машины*. – 2009. – 1 (219). – С. 47–53. 21. Bodyanskiy Ye., Vynokurova O., Yegorova E. Radial-basis-fuzzy-wavelet-neural network with adaptive activation-membership function // *ICGST Int. J. on Artificial Intelligence and Machine Learning (AIML)*. – 2008. – 8. – II. – P. 9–15. 22. Льюнг Л. Идентификация систем. Теория для пользователя. – Москва: Наука. ГРФМЛ, 1991. – 432 с. 23. Бодянский Е. В. Адаптивные алгоритмы идентификации нелинейных объектов управления // *АСУ и приборы автоматизики*. – Харьков: Выща шк., 1987. – Вып. 81. – С. 43–46. 24. Bodyanskiy Ye., Kolodyazhniy V., Stephan A. An adaptive learning algorithm for a neuro-fuzzy network / Ed. by B. Reusch "Computational Intelligence. Theory and Applications." – Berlin – Heidelberg – New York: Springer, 2001. – P. 68–75. 25. Hagglund T. Adaptive control of systems subject to large parameter changes // *Proc. IFAC 9-th Triennial WorldCongress*. – Budapest, 1984. – P. 993–998.

ДОКАЗОВА ТА ПРАКТИЧНА СТІЙКІСТЬ R-СХЕМИ БЛОЧНОГО ШИФРУВАННЯ ДО ДИФЕРЕНЦІАЛЬНОГО КРИПТОАНАЛІЗУ

© Яковлєв С., 2012

Наведено аналітичні оцінки верхніх меж імовірностей існування нетривіальних диференціалів та диференціальних характеристик для немарковської R-схеми блочного шифрування.

Ключові слова: симетричні блочні шифри, диференціальний аналіз, схема Фейстеля, R-схема.

Upper bounds for differential probabilities and differential characteristic probabilities of non-Markov R-scheme are estimated.

Key words: symmetric block ciphers, differential analysis scheme Feystelya, R-scheme.

Вступ

Схема Фейстеля є однією з класичних схем синтезування алгоритмів блочного шифрування [1]. Їй притаманні простота, зручність та менша вибагливість до обчислювальних ресурсів порівняно з підстановково-перестановковими мережами; окрім того, оскільки американський стандарт шифрування DES побудовано за схемою Фейстеля, вона є добре вивченою та детально дослідженою. Різні дослідники розробляли модифікації схеми Фейстеля, що зберігали плюси та надавали ті чи інші додаткові переваги. Наприклад, японський криптограф М. Мацуї запропонував так звану MISTY-схему (або L-схему) [2], що піддавалась паралелізації під час обчислення; цю схему покладено в основу шифрів сімейства MISTY. Коли ж виявилось, що L-схема вразливіша до деяких суто теоретичних атак, ніж оригінальна схема Фейстеля, було запропоновано іншу модифікацію – R-схему [3, 10], яку проаналізовано у цій роботі.

Диференціальний аналіз є одним з найпотужніших сучасних засобів криптоаналізу симетричних блочних шифрів. У перших відкритих публікаціях з диференціального аналізу (Е. Біхам, А. Шамір, 1991–1992 рр. [4, 5]), розглянуто застосування цього виду атак до шифрів DES та FEAL, і продемонстровано його неординарну потужність. Надалі вимога стійкості до диференціального аналізу та різних його модифікацій стала однією з необхідних під час синтезування новітніх алгоритмів шифрування.

Задачу доказової стійкості схеми шифрування до диференціального аналізу сформулювали К. Ніберг та Л.Р. Кнудсен у 1994 р. [8]; вони одержали верхню межу імовірності існування нетривіальних диференціалів DES-подібної схеми Фейстеля. Результати Ніберга та Кнудсена поліпшили Аокі та Ота [9] для випадку, коли як раундові перетворення DES-подібної схеми Фейстеля використовуються біективні функції. М. Мацуї навів аналогічні оцінки для L-схеми [2], а Канеко, Сано та Сакураї – для R-схеми (і для деяких інших модифікацій схеми Фейстеля) [10]. Також, вслід за Кнудсеном [12] та Кандою [13], почали розрізняти *теоретичну* (доказову) та *практичну* стійкість до диференціального аналізу; перша стосується оцінки імовірностей існування диференціалів, друга – диференціальних характеристик.

Однак необхідно зауважити, що всі згадані результати стосувались саме DES-подібних шифрів, а у ширшому розумінні – алгоритмів, що належать класу так званих *марковських шифрів* [6, 7]. Властивості марковських шифрів значно спрощують аналіз та оцінювання стійкості до

диференціальних атак; для немарковських шифрів ці властивості не виконуються, а тому в загальному випадку не будуть коректними результати, одержані для марковських шифрів.

Теоретичні дослідження немарковських схем шифрування, зокрема, ГОСТ-подібних шифрів, наведено в серії робіт Л.В. Ковальчук [14–16]. Зокрема, нею одержані оцінки верхніх меж існування нетривіальних диференціалів для немарковських схеми Фейстеля та L-схеми, аналогічні оцінкам Аокі, Оти та Мацуї. У цій роботі ми продовжуємо цей напрям. Нами будуть наведені точніші оцінки для диференціальних імовірностей немарковської R-схеми, які визначають доказову стійкість до диференціального аналізу, а також оцінки для імовірностей диференціальних характеристик R-схем, які визначають практичну стійкість. Дані оцінки виявились справедливими і для двох інших схем.

У першому розділі статті наведено формальний опис ітеративного блочного шифру, схеми Фейстеля, L-схеми та R-схеми, а також введено інші необхідні терміни та позначення. Другий розділ присвячено основним теоретичним положенням диференціального криптоаналізу. В третьому розділі ми коротко формулюємо існуючі та отримані результати, виводимо оцінки стійкості немарковської R-схеми до диференціального криптоаналізу.

1. Необхідні терміни та визначення

У цьому розділі ми наведемо визначення ітеративного симетричного блочного шифру та інші необхідні означення.

Раундове перетворення F_k – перетворення виду $F_k : V_q \times K \rightarrow V_q$, де V_q – множина бітових векторів довжини q , K – множина раундових ключів (ключовий простір) та $k \in K$. Ми будемо також вживати термін *раунд шифрування*, або просто *раунд*, маючи на увазі раундове перетворення.

Якщо раундове перетворення за структурою лінійно виражається через деяку іншу (нелінійну) функцію f_k меншої розмірності, то таку функцію ми називатимемо *раундовою функцією*.

Ітеративний r-раундовий блочний шифр E – перетворення виду $E : V_q \times K^r \rightarrow V_q$, що є композицією r раундових перетворень: $E = F_{k_r}^{(r)} \circ F_{k_{r-1}}^{(r-1)} \circ \dots \circ F_{k_1}^{(1)}$. Зауважимо, що тут і надалі ми вважаємо, що раундові ключі (k_1, k_2, \dots, k_r) є випадковими, незалежними та рівномірно розподіленими в ключовому просторі.

Якщо $Y = E_k(X)$, то будемо розглядати пов'язану із шифром послідовність (X_0, X_1, \dots, X_r) , де $X_0 = X$, $X_i = F_{k_i}^{(i)}(X_{i-1})$, $Y = X_r$.

Підхід, запропонований Хорстом Фейстелем для побудови блочних шифрів [1], полягав у тому, що на кожному раунді шифрування виконуються відносно прості математичні перетворення, а необхідний рівень стійкості забезпечується достатньо великою кількістю раундів та різними ключами на кожному раунді. Зокрема, Фейстель запропонував розбивати вхідний блок навпіл та опрацьовувати на кожному раунді лише половину блоку, змішуючи її наприкінці з іншою половиною; в цьому випадку, наприклад, 64-бітові блочні шифри можуть бути ефективно реалізовані на 32-бітній архітектурі.

Схема Фейстеля – ітеративний блочний шифр, кожен раунд шифрування якого має таку структуру:

$$F_k : (V_n)^2 \times K \rightarrow (V_n)^2,$$

$$F_k(x, y) = (y, x \oplus f_k(y)),$$

де f_k – відповідна раундова функція, а \oplus – операція побітового додавання двійкових векторів (XOR). Схему Фейстеля із r раундів, на яких використовуються раундові функції $f_k^{(1)}, f_k^{(2)}, \dots, f_k^{(r)}$, традиційно позначають через $\Psi[f_k^{(1)}, f_k^{(2)}, \dots, f_k^{(r)}]$.

L- та R-схеми визначаються так само, як і схема Фейстеля, відмінність полягає лише у структурі раундового перетворення.

MISTY-схема (або *L-схема*) – ітеративний блочний шифр, кожен раунд шифрування якого має таку структуру:

$$F_k(x, y) = (y, y \oplus f_k(x)).$$

R-схема – ітеративний блочний шифр, кожен раунд шифрування якого має таку структуру:

$$F_k(x, y) = (y \oplus f_k(x), f_k(x)).$$

За аналогією ми будемо використовувати позначення $L[f_k^{(1)}, f_k^{(2)}, \dots, f_k^{(r)}]$ та $R[f_k^{(1)}, f_k^{(2)}, \dots, f_k^{(r)}]$ для L- та R-схеми із r раундів, на яких використовуються раундові функції $f_k^{(1)}, f_k^{(2)}, \dots, f_k^{(r)}$.

Наведемо ще одне необхідне нам позначення.

Дужки Айверсона записуються як [твердження]. Вважається, що [твердження] = 1, якщо твердження істинне, та [твердження] = 0, якщо твердження хибне. Для дужок Айверсона виконується очевидна рівність $[P \wedge Q] = [P] \cdot [Q]$, де P та Q – деякі твердження.

2. Теоретичні відомості з диференціального аналізу

Нагадаємо основні означення диференціального аналізу.

Диференціал (звичайної) булевої функції f – пара двійкових векторів (α, β) така, що виконується співвідношення $f(z \oplus \alpha) \oplus f(z) = \beta$.

Імовірність диференціалу (α, β) функції f – величина

$$d^f(\alpha, \beta) = \frac{1}{2^q} \sum_{z \in V_q} [f(z \oplus \alpha) \oplus f(z) = \beta].$$

Вважатимемо, що диференціал є *неможливим*, якщо імовірність його існування дорівнює нулю. Якщо імовірність диференціалу дорівнює одиниці, такий диференціал назвемо *тривіальним*. Диференціали, що не є неможливими та не є тривіальними, будемо називатимемо *нетривіальними*.

Для функцій, що параметризовані ключем, диференціали розглядаються в кожній точці окремо:

Середня за ключами імовірність диференціала (α, β) функції f_k в точці z – величина

$$d^{f_k}(z, \alpha, \beta) = \frac{1}{|K|} \sum_{k \in K} [f_k(z \oplus \alpha) \oplus f_k(z) = \beta].$$

Середня імовірність диференціала (α, β) функції f_k – величина

$$d^{f_k}(\alpha, \beta) = \frac{1}{2^q} \sum_{z \in V_q} d^{f_k}(z, \alpha, \beta).$$

Ми також позначатимемо диференціал звичайної функції символом $\alpha \xrightarrow{f} \beta$ або просто $\alpha \rightarrow \beta$, якщо функція буде зрозуміла з контексту. Диференціал функції, що параметризована ключем, в точці z позначатимемо $\alpha \xrightarrow{f_k, z} \beta$, $\alpha \xrightarrow{f_k} \beta$, якщо з контексту зрозуміла точка, або $\alpha \rightarrow \beta$, якщо з контексту зрозуміла й функція також.

Диференціальна характеристика r -раундового ітеративного блочного шифру E – послідовність $\Omega = (\omega_0, \omega_1, \dots, \omega_r) \in (V_q \setminus \{0\})^{r+1}$ така, що (ω_{i-1}, ω_i) – диференціал раундової функції $f_{k_i}^{(i)}$.

Загалом диференціальною характеристикою може бути довільна послідовність ненульових двійкових векторів потрібної довжини.

Середня за ключами ймовірність диференціальної характеристики (у точці X_0) визначається як

$$DP(\Omega, X_0) = P(\omega_0 \xrightarrow{f_{k_1}^{(1)}, X_0} \omega_1, \omega_1 \xrightarrow{f_{k_2}^{(2)}, X_1} \omega_2, \dots, \omega_{r-1} \xrightarrow{f_{k_r}^{(r)}, X_{r-1}} \omega_r).$$

Імовірність диференціала (ω_0, ω_r) шифру E тоді обчислюється через імовірності всіх можливих відповідних диференціальних характеристик:

$$d^E(z, \omega_0, \omega_r) = \sum_{\omega_1 \neq 0, \dots, \omega_{r-1} \neq 0} DP(\Omega, z).$$

Перетворення називається *марковським*, якщо імовірності кожного диференціала на кожному раунді не залежать від точки входу, тобто $d^{f_k}(z, \alpha, \beta) = d^{f_k}(0, \alpha, \beta) = d^{f_k}(\alpha, \beta)$. У цьому випадку диференціальна характеристика утворює марковський ланцюг, а відповідна імовірність розбивається у добуток незалежних диференціальних імовірностей кожного раунду, що значно спрощує аналіз. Для немарковських перетворень таке розбиття буде некоректним, однак можна побудувати оцінку згори, як це показано у лемі 4.

Оцінка стійкості немарковського шифру до диференціального аналізу визначається величиною

$$MDP(E) = \max_{\alpha \neq 0, \beta, z} d^E(z, \alpha, \beta).$$

Сформулюємо декілька лем, що описують властивості диференціалів та диференціальних характеристик. Доведення цих лем можна знайти у [2, 6–11, 14–16].

Лема 1. Для довільної функції f (f_k) справедливі співвідношення:

а) $P(0 \rightarrow \beta) = [\beta = 0]$

б) $P(\alpha \rightarrow 0) = [\alpha = 0]$, якщо функція f – бієктивна (f_k – бієктивна при заданому значенні ключа).

Лема 2. Для диференціалів функції f виконуються співвідношення:

а) $\forall \alpha : \sum_{\beta} P(\alpha \xrightarrow{f} \beta) = 1.$

б) $\forall \beta : \sum_{\alpha} P(\alpha \xrightarrow{f} \beta) = 1$, якщо f – бієктивна.

Для диференціалів функції f_k , що параметризована ключем, виконуються співвідношення:

в) $\forall \alpha \forall z : \sum_{\beta} P(\alpha \xrightarrow{f_k, z} \beta) = 1.$

г) $\forall \beta \forall z : \sum_{\alpha} P(\alpha \xrightarrow{f_k, z} \beta) = 1$, якщо f_k бієктивна для кожного значення k .

Лема 3. Нехай q – максимальна імовірність існування нетривіального r -раундового диференціала ітеративного шифру. Тоді імовірність існування нетривіального $(r+1)$ -раундового диференціала того самого шифру не перевищує q .

Лема 4. Для r -раундової диференціальної характеристики $\Omega = (\omega_0, \omega_1, \dots, \omega_r)$ ітеративного шифру E_K виконується нерівність:

$$\forall z : DP(\Omega, z) \leq \prod_{i=1}^r \max_x d^{f_{k_i}^{(i)}}(x, \omega_{i-1}, \omega_i).$$

3. Основні результати

Аналітичні оцінки верхньої межі імовірності існування нетривіальних диференціалів немарковських схеми Фейстеля та MISTY-схеми були вперше одержані Ковальчук у [15]. Наведемо ці результати у вигляді наступних двох теорем.

Теорема 1 (Ковальчук)

Нехай $E = \Psi[f_1, f_2, f_3]$ – трираундова схема Фейстеля, раундові функції якої бієктивні за кожного значення ключа, а $p = \max\{MDP(f_1), MDP(f_2), MDP(f_3)\}$. Тоді існує співвідношення:

$$MDP(E) \leq p^2,$$

тобто імовірність довільного нетривіального трираундового диференціала E не перевищує величини p^2 .

Теорема 2 (Ковальчук)

Нехай $E = \mathcal{L}[f_1, f_2, f_3]$ – трираундова L-схема, раундові функції якої бієктивні за кожного значення ключа, а $p = \max\{MDP(f_1), MDP(f_2), MDP(f_3)\}$. Тоді існує співвідношення:

$$MDP(E) \leq p^2,$$

тобто імовірність довільного нетривіального трираундового диференціала E не перевищує величини p^2 .

Зауважимо, що ці результати аналогічні оцінкам, які було одержано для марковських DES-подібних шифрів: Аокі та Отою для схеми Фейстеля, Мацуї для L-схеми, Канеко, Сано та Сакураї для R-схеми. У наступній теоремі ми подамо аналогічні, але уточнені оцінки для диференціальних імовірностей R-схеми.

Теорема 3

Нехай $E = R[f_1, f_2, f_3]$ – трираундова R-схема, раундові функції якої бієктивні за кожного значення ключа, і $p_i = MDP(f_i)$, $i = \overline{1, 3}$. Тоді існує співвідношення:

$$MDP(E) \leq \max\{p_1 p_2, p_1 p_3, p_2 p_3\}.$$

Доведення. Зафіксуємо деяку вхідну точку $X = (x_1, x_2)$ та розглянемо довільну нетривіальну диференціальну характеристику $\Omega = (\omega_0, \omega_1, \omega_2, \omega_3)$ (для зручності запису ми будемо опускати відповідні вхідні точки). Зі схеми шифрування видно, що якщо $P(\Omega) \neq 0$, то мають виконуватись такі співвідношення: $\omega_0 = (a_1, a_2)$, $\omega_1 = (a_2 \oplus b_1, b_1)$, $\omega_2 = (c_2 \oplus b_1, c_2)$, $\omega_3 = (c_1, c_2)$, де a_1, a_2, b_1, c_1, c_2 – деякі бітові вектори. Отже,

$$DP(\Omega, X) = P\left(a_1 \xrightarrow{(1)} b_1, a_2 \oplus b_1 \xrightarrow{(2)} c_2, c_2 \oplus b_1 \xrightarrow{(3)} c_1\right)$$

(тут число у дужках під стрілочками показує номер раундової функції).

Для довільного нетривіального диференціала (a, c) маємо:

$$d^E(X, a, c) = \sum_{b_1} DP(\Omega, X), \text{ де } \omega_0 = a, \omega_3 = c,$$

тобто, при фіксуванні вхідної та вихідної різниць в характеристиці залишається невизначеною лише змінна b_1 .

Очевидно, що за різних значень (a, c) деякі з диференціальних переходів тривіалізуються. Розглянемо всі можливі випадки.

1. Нехай $a_1 = 0$. За лемою 1 маємо $a_2 \neq 0$ та $b_1 = 0$. Значення b_1 фіксується, тому

$$d^E(X, a, c) = P\left(a_2 \xrightarrow{(2)} c_2, c_2 \xrightarrow{(3)} c_1\right) \leq p_2 p_3.$$

2. Нехай тепер $a_1 \neq 0$, але $c_1 = 0$; тоді $c_2 \neq 0$ та $b_1 = c_2 \neq 0$ (також за лемою 1). Значення b_1 знов фіксується, отже, маємо:

$$d^E(X, a, c) = P\left(a_1 \xrightarrow{(1)} c_2, a_2 \oplus c_2 \xrightarrow{(2)} c_2\right) \leq p_1 p_2.$$

3. Нехай тепер $a_1 \neq 0$, $c_1 \neq 0$, але $c_2 = 0$. У цьому випадку $b_1 = a_2$ та $b_1 \neq 0$, а тому

$$d^E(X, a, c) = P\left(a_1 \xrightarrow{(1)} a_2, c_2 \xrightarrow{(3)} c_1\right) \leq p_1 p_3.$$

4. Нарешті, нехай $a_1 \neq 0$, $c_1 \neq 0$ та $c_2 \neq 0$. У цьому випадку значення b_1 не фіксується, отже, можемо записати:

$$\begin{aligned} d^E(X, a, c) &= \sum_{b_1} P\left(a_1 \xrightarrow{(1)} b_1, a_2 \oplus b_1 \xrightarrow{(2)} c_2, c_2 \oplus b_1 \xrightarrow{(3)} c_1\right) = \\ &= \sum_{b_1} P\left(a_1 \xrightarrow{(1)} b_1\right) \cdot P\left(a_2 \oplus b_1 \xrightarrow{(2)} c_2, c_2 \oplus b_1 \xrightarrow{(3)} c_1 \mid b_1\right) \leq p_2 p_3 \sum_{b_1} P\left(a_1 \xrightarrow{(1)} b_1\right) = p_2 p_3, \end{aligned}$$

де ми спочатку застосували лему 4, а потім – лему 2.

Підсумовуючи всі випадки, бачимо, що $d^E(X, a, c) \leq \max\{p_1 p_2, p_1 p_3, p_2 p_3\}$ для довільного диференціала (a, c) та довільної точки X , звідки й випливає твердження теореми.

Наслідок. Якщо $p = \max\{p_1, p_2, p_3\}$, то $MDP(E) \leq p^2$.

Отже, ми можемо знайти верхню межу імовірності існування нетривіальних трираундових диференціалів R-схеми, використовуючи відповідні параметри раундових функцій. Оскільки раундові функції R-схеми мають вдвічі меншу розмірність, знайти їх диференціальні імовірності значно легше в обчислювальному плані.

З леми 3 випливає, що одержана оцінка буде справедливою для довільного r -раундового диференціала R-схеми при $r \geq 3$. Отже, за теоремою 3 можна встановити доказову (теоретичну) стійкість шифрів, побудованих на основі R-схеми, до диференціального аналізу.

З теореми 3 також безпосередньо можна одержати оцінки практичної стійкості – оцінки верхньої межі імовірності існування довільних диференціальних характеристик. Цю оцінку подамо у теоремі 4.

Теорема 4. Нехай $E = R[f_1, f_2, \dots, f_r]$ – r -раундова R-схема, раундові функції якої бієктивні за кожного значення ключа, і $p = \max_i MDP(f_i)$. Тоді для довільної диференціальної характеристики Ω та довільної вхідної точки X існує співвідношення:

$$DP(\Omega, X) \leq p \left\lceil \frac{r}{3} \right\rceil.$$

Доведення. З доведення теореми 3 випливає, що з довільних послідовних трьох раундів R-схеми тривіалізуватись може не більше одного, тобто з трьох послідовних раундів щонайменше два будуть нетривіальними. Звідси одразу ж випливає твердження теореми 4.

Потрібно зауважити, що для схеми Фейстеля та L-схеми існують аналогічні уточнені оцінки як для імовірностей диференціалів, так і для імовірностей диференціальних характеристик. Доведення цих фактів майже ідентичне доведенню теорем 3 та 4.

Висновки

У цій роботі ми навели аналітичні оцінки верхніх меж імовірностей існування нетривіальних диференціалів та диференціальних характеристик для R-схеми блочного шифрування у загальному вигляді. Такі оцінки є параметрами, що визначають доказову (теоретичну) та практичну стійкість блочних шифрів, побудованих на основі R-схеми, до диференціального криптоаналізу. Зауважимо, що, на відміну від попередньо опублікованих результатів, одержані нами оцінки є точніші та справедливі для загального випадку немарковських шифрів.

Результати цієї роботи можна застосовувати для теоретичного та практичного оцінювання стійкості до диференціального криптоаналізу як складніших схем блочного шифрування, так і конкретних алгоритмів.

1. Feistel H. *Cryptography and Computer Privacy* – *Scientific American*, v. 228, n. 5, May 1973, pp. 15–23. 2. Matsui M. *On a Structure of Block Ciphers with Provable Security against Differential and Linear Analysis* – *IEICE Trans. Fundamentals*, vol. E82-A – 1999 – #1 – P. 117–122. 3. Gilbert H., Minier M. *New Results on the Pseudorandomness of Some Blockcipher Constructions* – *Lecture Notes in Computer Science*, 2002, Volume 2355, *Fast Software Encryption*. – P. 159–178 4. Biham E., Shamir A. *Differential cryptanalysis of DES-like cryptosystems* // *Journal of Cryptology*. – 1991. – V. 4. – № 1. – P. 3–72. 5. Biham E., Shamir A. *Differential cryptanalysis of the full 16-round DES* // *Advances in Cryptology – CRYPTO'92, Proceedings*. – Springer Verlag, 1993. – P. 487–496. 6. Lai X., Massey J.L., Murphy S. *Markov ciphers and differential cryptanalysis* // *Advances in Cryptology – EUROCRYPT'91, Proceedings*. – Springer Verlag, 1991. – P. 17–38. 7. O'Connor L., Golic J.D. *A unified Markov approach to differential and linear cryptanalysis* // *Advances in Cryptology – ASIACRYPT'94, Proceedings*. – Springer Verlag, 1994. – P. 387–397. 8. Nyberg K., Knudsen L.R. *Provable Security Against a Differential Attack* // *Journal of Cryptology*, Vol. 8, no. 1 (1995). 9. Aoki K., Ohta K. *Stricter Evaluation for the Maximum Average of Differential Probability and the Maximum Average of Linear Probability* // *Proceedings of SCIS'96, SCIS96-4A (1996)* (японською мовою). 10. Kaneko Y., Sano F. and Sakurai K. *On Provable Security against Differential and Linear Cryptanalysis in Generalized Feistel Ciphers with Multiple Random Functions* – *Proc. of SAC'97, 1997*, pp. 185–199. 11. Vaudenay S. *On the security of CS-cipher* // *Fast Software Encryption. – FSE'99, Proceedings*. – Springer Verlag, 1999. – P. 260–274. 12. Knudsen L.R. *Practically secure Feistel cipher* // *Fast Software Encryption. – FSE'94, Proceedings*. – Springer Verlag, 1994. – P. 211–221. 13. Kanda M. *Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function* // *Selected Areas in Cryptography. – SAC 2000, Proceedings*. – Springer Verlag, 2001. – P. 324–338. 14. Ковальчук Л.В. *Обобщенные марковские шифры: построение оценки практической стойкости относительно дифференциального криптоанализа* // *Математика и безопасность информационных технологий. Материалы конференции в МГУ 25–27 октября 2006 г.* – М.: МЦНМО, 2007. – С. 595–599. 15. Ковальчук Л.В., Шерстюк А.О. *Дослідження різницевих характеристик раундової функції блочних шифрів MISTY1 та MISTY2* // *Прикладная радиоэлектроника*. – 2009. – № 3. – С. 15–27. 16. Ковальчук Л.В., Пальченко С.В., Скрипник Л.В. *Застосування теорії узагальнених марковських шифрів для оцінювання стійкості сучасних блокових алгоритмів шифрування до методів різницевого криптоаналізу* // *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні* – К.: НДЦ «Тезіс», 2009. – № 2 (19). – С. 45–56.

***В. Вартузов, С. Бабічев, В. Литвиненко, А. Фефелов**

*Державний науково-дослідний експертно-криміналістичний центр МВС України,
Херсонський національний технічний університет

МОДЕЛЬ АНАЛІТИЧНОЇ СИСТЕМИ ПРОФІЛЮВАННЯ НАРКОТИЧНИХ РЕЧОВИН НА ОСНОВІ МЕРЕЖІ БАЙЕСА

© Вартузов В., Бабічев С., Литвиненко В., Фефелов А., 2012

Наведено аналітичну модель системи профілювання наркотичних речовин, що дозволяє оцінити ймовірність шляху транзиту наркотиків та місце їх виготовлення. Побудована мережа Байєса, що враховує характер взаємозв'язку між вузлами системи, в якій як цільовий був прийнятий вузол, що визначає назву лабораторії виробництва наркотичної речовини. Показано реалізацію запропонованої моделі на прикладі трьох лабораторій, шести дилерів, які поставляють товар до місця призначення і чотирьох компонент наркотичної речовини, які додають дилери в процесі транспортування товару.

Ключові слова: байєсовська мережа, профілювання наркотичних речовин.

In the article presents the analytical model of the system profiling of the drugs allowing estimating the probability of drug transit route of the drugs and placing their manufacture. Bayesian network is constructed, taking into account the nature of the relationship between the nodes of the system, which has as its target was adopted node, defines the name of the laboratory drug production. Shows the implementation of the model proposed by the example of the three laboratories, six dealers that supply goods to the destination and the four components of a narcotic substance, added dealers during its transportation.

Key words: Bayesian network profiling of drugs.

Вступ

Зловживання забороненими наркотиками – серйозна проблема всесвітнього характеру, яку можна вирішити лише у межах міжнародного співробітництва. Один з найефективніших заходів боротьби з наркотиками – запобігання їх виробництву. У разі синтетичних наркотиків контроль над прекурсорами є важливим компонентом стратегії запобігання їх виробництву. Дослідження в області створення аналітичної системи профілювання наркотичних речовин можуть забезпечити правоохоронні органи корисною інформацією. Може бути виявлено подібність між пробами за хімічним складом, а матеріал з різних вилучених партій можна класифікувати за групами родинних проб. Найважливішим наслідком для правоохоронних органів є той факт, що цим можна виявити специфічні зв'язки, наприклад, між постачальниками наркотиків і особами, що вживають наркотики, визначити тенденції поширення наркотиків і мережі їх збуту, а також ідентифікувати джерела, зокрема географічне походження проб.

Постановка проблеми

У [1] показано, що незаконні наркотики та психотропні речовини, як правило, являють собою складну суміш, і лише в окремих випадках містять чисту наркотичну речовину. Поряд з наркотичною або психотропною речовиною проби можуть містити одну або більше різних типів основних компонентів. Кожну компоненту залежно від характеру аналізу речовини можна охарактеризувати за допомогою різних якісних або кількісних характеристик. У цей момент часу одні характеристики є відомими, а інші – невідомими. Наприклад, хроматограма наркотичної речовини або процентний вміст його компонент, визначений шляхом хімічного аналізу, є підмножиною відомих характеристик, а

випускна лабораторія і шлях поставки – підмножиною невідомих. Завдання полягає в тому, щоб, маючи набір відомих характеристик досліджуваної хімічної речовини визначити його невідомі характеристики. У роботах [2–5] автори для розв’язання таких типів задач запропонували використовувати мережу Байєса. Основна проблема при цьому полягає у виборі структури мережі з урахуванням характеру взаємозв’язку між вузлами і апріорне визначення умовних ймовірностей вузлів предків мережі. Однак у криміналістиці і зокрема для виявлення шляхів постачання наркотичних речовин мережі Байєса сьогодні не одержали належного поширення.

Мета роботи – розроблення моделі системи профілювання наркотичних речовин на основі мережі Байєса.

Викладення основного матеріалу

Байєсовська мережа [4] являє собою пару $\langle G, V \rangle$, у якій перша компонента G – це спрямований нециклічний граф, що відповідає випадковим змінним і записується як набір умов незалежності: кожна змінна незалежна від її батьків в G . Друга компонента пари V – це множина параметрів, що визначають мережу. Кожна змінна $X^{(i)} \in G$ представляється у вигляді вершини. Повна спільна ймовірність БМ обчислюється за формулою:

$$P_B(X^{(1)}, \dots, X^{(N)}) = \prod_{i=1}^N P_B(X^{(i)} | Pa(X^{(i)})).$$

З математичного погляду БМ – це модель подання наявних і відсутніх імовірнісних залежностей. При цьому зв’язок $A \rightarrow B$ є причинним, коли подія A – причина виникнення B , тобто коли існує механізм, відповідно до якого значення, прийняте A , впливає на значення, прийняте B .

Нехай Ω – множина подій випадкових експериментів. Цей вибірковий простір містить всі можливі значення випадкової змінної. Припустимо, що дві змінні E і H деяким чином пов’язані між собою. Умовна ймовірність події E визначається за виразом:

$$p(E | H_k) = \frac{p(E \cap H_k)}{p(H_k)}, \quad (1)$$

де \cap – операція перетину множин. Ця залежність дає змогу визначити ймовірність події E , якщо існує конкретна подія H_k . Якщо E_1, E_2, \dots, E_n – взаємовиключні події такі, що $\cup_{i=1}^n E_i = \Omega$, то кажуть, що події E_i формують повну (вичерпну) множину. Дві змінні не зв’язані (не перетинаються), якщо вони не мають однакових значень. Якщо дві змінні є вичерпними і незв’язаними, то можна записати, що

$$E = \cup_i (E \cap H_i), \quad (E \cap H_i) \cap (E \cap H_j) = \emptyset, \quad i \neq j. \quad (2)$$

Теорія побудови байєсівських мереж ґрунтується на припущенні, що події є вичерпними і не перетинаються. Якщо ця умова не виконується, то результати застосування мережі будуть неконсистентними (тобто неточними). У випадку, коли події вичерпні і не перетинаються, то ймовірність події E можна обчислити за допомогою умовних ймовірностей:

$$p(E) = \sum_{i=1}^n p(E \cap H_i) = \sum_{i=1}^n p(E | H_i) \cdot p(H_i). \quad (3)$$

Використовуючи рівняння (3), суму перетинів події E з H можна подати так:

$$p(E \cap H_k) = p(E | H_k) \cdot p(H_k) = p(H_k | E) \cdot p(E). \quad (4)$$

З рівності $p(E | H_k) \cdot p(H_k) = p(H_k | E) \cdot p(E)$ знайдемо, що

$$p(H_k | E) = \frac{p(E | H_k) \cdot p(H_k)}{p(E)},$$

а із врахуванням (3) отримаємо вираз:

$$p(H_k | E) = \frac{p(E | H_k) \cdot p(H_k)}{\sum_{i=1}^n p(E | H_i) \cdot p(H_i)}, \quad (5)$$

який являє собою формулу Байеса. На основі цієї формули будується байесовська мережа.

В (5) H_k означає будь-яку гіпотезу з n можливих. Ймовірності $p(E | H_k)$ задають експерти апріорно або розраховують за навчальними даними [6, 7]. Тобто їх можна розглядати як відповідь на запитання: “Якою буде ймовірність деякого виміру, якщо відомо, яка гіпотеза була реалізована?”. Ймовірності $p(E | H_k)$ є дуже корисними, тому що, як правило, легше знайти ймовірність послідовності подій типу причина–наслідок, ніж навпаки. Значення $p(H_k)$ називають апріорними ймовірностями; вони визначають початкові ймовірності для всіх гіпотез. Сила байесівського методу полягає в тому, що апріорні ймовірності можна уточнювати (оновлювати) відповідно до реалій проходження досліджуваного процесу. Це дає змогу уточнювати ймовірності подій при надходженні додаткової інформації. Знаменник виразу (5) можна розглядати як нормуючий член, який встановлює значення ймовірності між 0 та 1.

Профілювання – це процедура ідентифікації й аналізу домішок у наркотичних речовинах, вилучених з незаконного обігу і на основі цієї інформації встановлення, якими методами вони були синтезовані, а також визначення спільних шляхів походження.

Профілювання передбачає систематичне збирання і розділення за групами вилучених з незаконного обігу наркотичних речовин на основі фізичної і хімічної інформації, отриманої внаслідок їх дослідження, яке містить аналіз і визначення домішок, щоб порівняти різні наркотичні зразки. Отримані результати забезпечують важливе джерело інформації для подальших аналізів.

Профіль вмісту хімічних домішок в пробі – це профіль вмісту домішок, які є результатом спільного екстрагування і виготовлення (профіль вмісту домішок), а також розріджувачів.

Наркотики та психотропні речовини, як правило, є складною сумішшю, лише в окремих випадках містять чисту наркотичну речовину, незалежно від того, чи мають вони рослинне походження. Поряд з наркотичною чи психотропною речовиною проби можуть містити один або більше трьох різних типів основних компонентів [1]:

- **природні компоненти**, присутні в сировині (наприклад, лист коки, опій) і використовувані у виробництві певних наркотиків “рослинного походження”, таких як кокаїн або героїн; в процесі виробництва наркотику їх екстрагують спільно з наркотичною речовиною і не видаляють повністю із кінцевого продукту;
- **побічні продукти**, що утворюються під час виробництва наркотику і пов’язані із способом його виготовлення;
- **розчинники та розбавлювачі**, медичні препарати тощо, які можуть бути додані на будь-якому наступному за виготовленням етапі в ланцюзі збуту наркотиків.

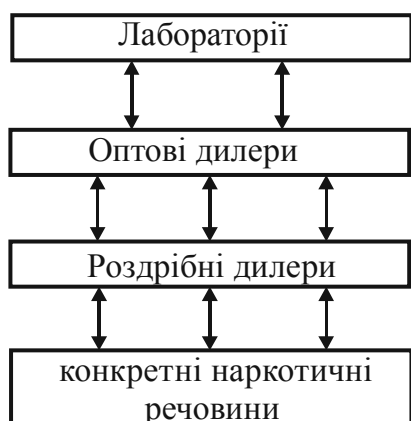


Рис. 1. Загальна схема моделі системи профілювання наркотичних речовин

Проби синтетичних наркотиків, таких як амфетаміни, можуть містити різні синтетичні домішки, зокрема залишки основних хімічних речовин, що використовувалися в процесі виготовлення наркотиків, а також побічні продукти, що є результатом побічних реакцій. Наявність і відносна концентрація таких домішок залежать від якості вихідних матеріалів, шляхів їхнього синтезу, умов проведення хімічних реакцій, ступеня очищення кінцевого продукту (продуктів) і найбільше – від кваліфікації хіміка, що працює в підпільних умовах. Крім того, на різних рівнях ланцюга розповсюдження до наркотиків можуть додаватися різні домішки, наприклад, медичні препарати, з метою збільшення обсягу кінцевого продукту та, відповідно, доходу.

Загальну схему моделі системи профілювання наркотичних речовин наведено на рис. 1.

За цією моделлю з кожною наркотичною речовиною можуть бути асоційовані різноманітні поняття, які його характеризують. При цьому пропонується модель не обмежує дослідника жорсткою односпрямованістю дій, а може гнучко використовуватися як для виявлення причинних характеристик (наприклад, за відомим хімічним складом і місцем вилучення визначити лабораторію та можливі шляхи поставки), так і передбачати наслідки (наприклад, якщо було виявлено, що деяка лабораторія випустила партію речовини, то визначити, появи якого типу речовин і на якій території слід очікувати найближчим часом). Описані вище властивості моделі на рис. 1 проілюстровано стрілками, що показують рух інформації між виділеними блоками. Слід зазначити, що внаслідок специфіки роботи байєсівських мереж всі висновки за цією моделлю щодо шуканої інформації мають імовірнісний характер і наведені у вигляді рангового (за значеннями імовірності вірності того або іншого висновку) списку. Остаточне рішення приймає дослідник.

Для розв'язання поставленої задачі пропонується використовувати класифікувальну структуру байєсівської мережі, приклад якої зображено на рис. 2. Як видно з рисунка, така мережа враховує не тільки незалежні, але й пов'язані між собою атрибути. Це зручно, наприклад, у тому випадку, коли необхідно отримати інформацію про канал постачання. Структуру байєсівської мережі, орієнтованої на розв'язання задачі профілювання наркотичних речовин, наведено на рис. 3. Дилери 1,2 і 3 являють собою клас оптових дилерів, а 4,5 і 6 – роздрібні дилери.

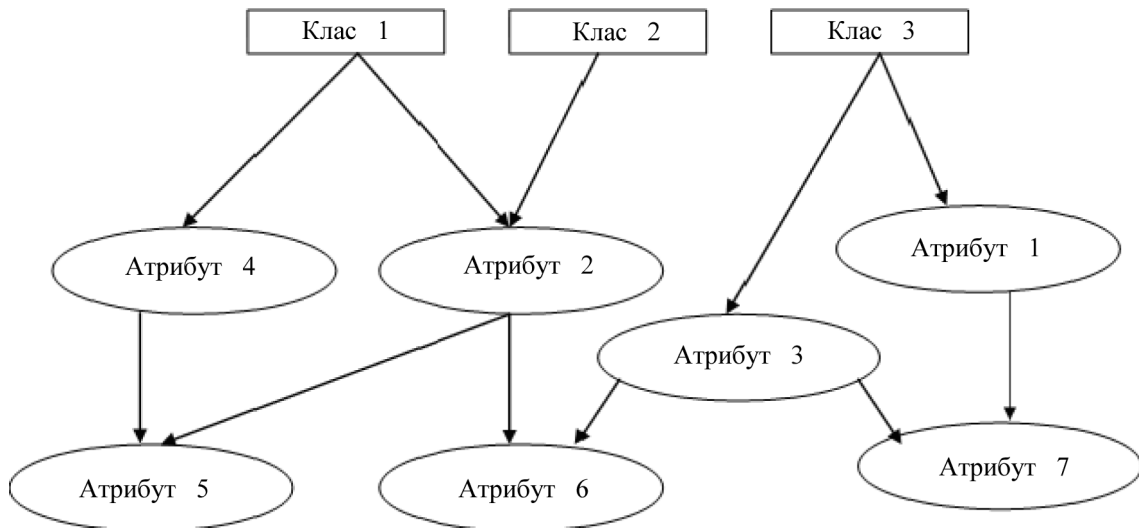


Рис. 2. Приклад структури байєсівської мережі для розв'язання задач класифікації

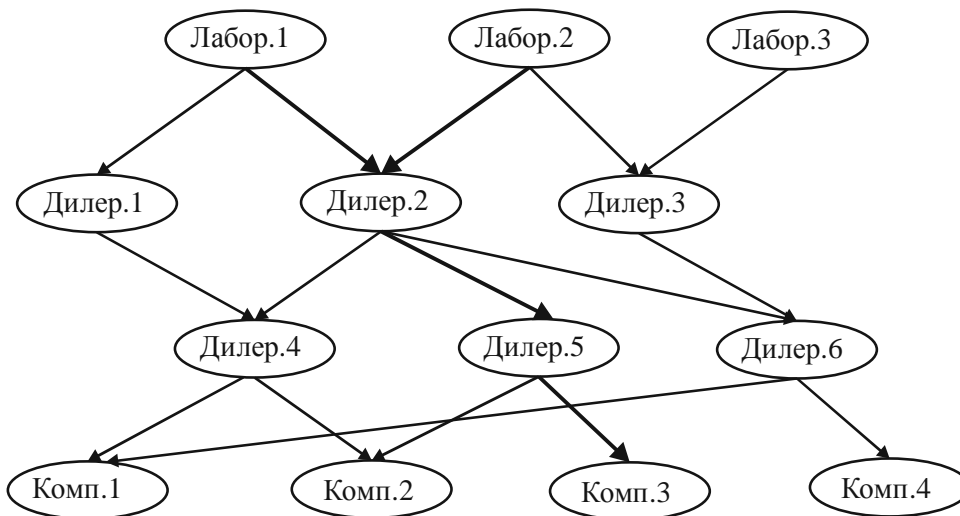


Рис. 3. Структура байєсівської мережі для розв'язання задачі профілювання наркотичних речовин

Так, наприклад, аналіз наркотичної речовини показав, що воно поряд з іншими компонентами містить компоненту 3. Як видно з рис. 3, цю компоненту було додано роздрібним дилером 5, який отримав товар від оптового дилера 2. Поставки цього дилера йдуть, своєю чергою, від лабораторій 1 і 2. Уточнити конкретну лабораторію, яка постачала товар, можна, аналізуючи інші компоненти наркотичної речовини. Отже, встановивши на підставі існуючої бази даних [1] характер взаємозв'язку між вузлами мережі, можна з високим ступенем ймовірності передбачити шлях поставки товару в пункт його збуту.

Розв'язували задачу побудови мережі Байєса, використовуючи програмне середовище GeNIe 4.0. Як цільовий вузол приймали вузол, який визначає лабораторію виробництва наркотиків. Модель байєсівської мережі для розв'язання задачі профілювання наркотичних речовин наведено на рис. 4 і 5.

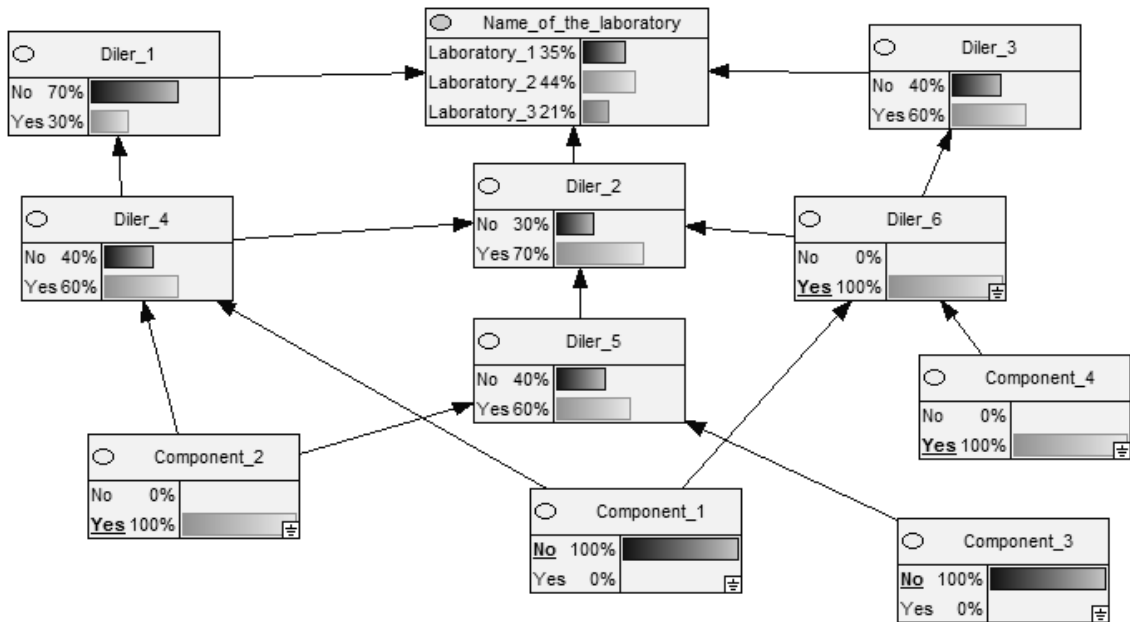


Рис. 4. Модель байєсівської мережі для розв'язання задачі профілювання наркотичних речовин за наявності компонент 2 і 4

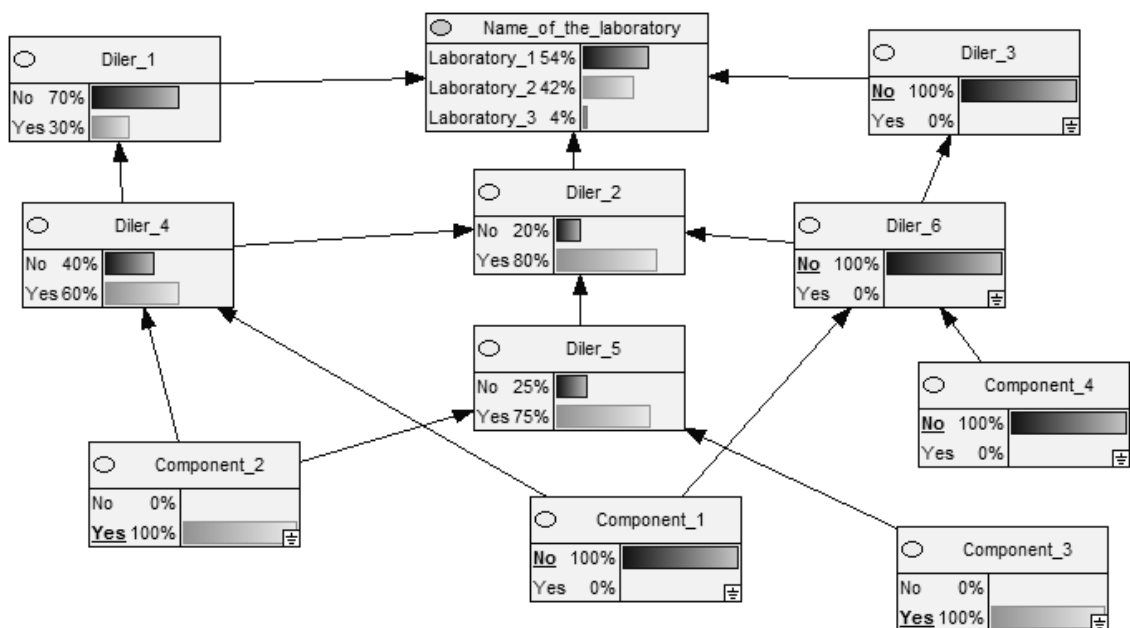


Рис. 5. Модель байєсівської мережі для розв'язання задачі профілювання наркотичних речовин за наявності компонент 2 і 3

Як видно з рис. 4, за наявності в наркотичній речовині компонент 2 і 4, з ймовірністю 44 % можна стверджувати, що виробництво здійснювалося в лабораторії 2, а поставку проводили через роздрібногo дилера 6, який, своєю чергою, отримав товар з ймовірністю 70 % від оптового дилера 2 і з ймовірністю 60 % від оптового дилера 3. Модель, яку наведено на рис. 5, дає змогу зробити висновок, що за наявності в речовині компонент 2 і 3 з високою ймовірністю можна стверджувати, що виробництво здійснювалося в лабораторії 1 і шлях поставки проходив через дилерів 2 і 5.

Висновок

У роботі розроблено модель байесівської мережі для розв'язання задачі профілювання наркотичних речовин. Запропонована модель дає змогу, аналізуючи наркотичну речовину, оцінити ймовірність місця виробництва цієї речовини і визначити можливі шляхи його транспортування до місця збуту. В процесі налагодження моделі враховували два стани компоненти наркотичної речовини: присутній або відсутній. При цьому в силу рівноцінності цих станів кожному з них було приписано ймовірність 50 %. Однак за наявності бази даних з точним складом компонентів цієї чи іншої речовини і відомою його передісторією можливе точніше налагодження моделі з урахуванням обсягу компоненти. Подальшим напрямом досліджень авторів є реалізація цієї моделі за реальними даними відповідно до [1].

1. Организация Объединённых Наций. Описание свойств наркотиков/составление профилей содержания примесей. Основные понятия и концепции. Руководство для национальных правоохранительных органов и лабораторий экспертизы наркотиков. ST/NAR/32/Rev.1 / Организация Объединённых Наций. – Нью-Йорк, 2004. – С. 1–18. 2. Бидюк П.І., Терентьев О.М., Коршевнюк Л.О. Интеллектуальный анализ слабоструктурированных данных за помощью байесовых сетей / Звіт по результатам виконання робіт за грантом НТУУ „КПІ” № 3/5-ГР, 2006, 2007р. – 85 с. 3. Згуровський М.З., Бидюк П.І., Терентьев О.М. Системна методика побудови байесових мереж // “Наукові вісті” НТУУ “КПІ”. – 2007. – №4. – С. 47–61. 4. Терентьев А.Н., Бидюк П.И., Коршевнюк Л.А. Байесовская сеть – инструмент интеллектуального анализа данных // Проблемы управления и информатики. – К.: ИКИ НАНУ-НКАУ, 2007. – № 4. – С. 83–92. 4. Terentyev A.N., Gasanova L.T. Bayesian networks in credit scoring / The second international conference on control and optimization with industrial applications (COIA-1008), Baku, June 2-8, 2008. – Baku: Institute of applied mathematics BSU, 2008. – P. 171. 5. Терентьев А.Н., Бидюк П.И., Коршевнюк Л.А. Алгоритм вероятностного вывода в байесовских сетях / Материалы IX международной научно-технической конференции “Системный анализ и информационные технологии” (САИТ-2007), г. Киев, 15–19 мая 2007. – К.: Екмо, 2007. – С. 76. 6. Бидюк П.И., Терентьев А.Н. Метод вероятностного вывода в байесовских сетях по обучающим данным // Кибернетика и системный анализ. – 2007. – № 3. – С. 93–99.

МЕТОД ЗНАХОДЖЕННЯ ВІДПОВІДНИХ ТОЧОК НА КОНТУРАХ МІКРООБ'ЄКТІВ БІОМЕДИЧНОЇ ПРИРОДИ

© Мельник Г., 2012

Характерною властивістю цифрових біомедичних зображень є наявність повторюваних елементів з нечіткою геометричною структурою. Наближення форми клітин до форми круга або еліпса вимагає вдосконалення алгоритмів зіставлення областей. Для визначення найкращого афінного перетворення між областями використано симетричну різницю множин точок областей. Застосуванням дискретних центральних моментів області підвищено точність визначення відповідних точок на контурах при визначенні афінних перетворень між областями мікрооб'єктів.

Ключові слова: біомедичні зображення, структурний аналіз, зіставлення областей, центральні моменти, контур, відповідні точки.

The characteristic feature of digital biomedical images is presence of repeated elements with fuzzy geometry. Cell shape is approximately equals to shape of circle or ellipse. It demands an improved algorithms for region matching. The symmetric difference of regions point sets used to determine the best affine transformation between the regions. Application of region discrete central moments increases the accuracy of contours corresponding points during calculation of affine transformations between the regions of microobjects.

Key words: biomedical image, structural analysis, region matching, central moments, contour, correspondence points.

Вступ

Сьогодні опрацювання зображень є важливим напрямком застосування сучасної обчислювальної техніки. Задачами опрацювання зображень є опис, аналіз та оброблення зображень. Проблеми аналізу зображень, крім класичної задачі розпізнавання фігур заданої форми, ставлять нові завдання опису структури зображення.

Біомедичні зображення – це растрові зображення, отримані за допомогою будь-якої біомедичної техніки, що використовуються для візуального та автоматизованого аналізу у медицині та біології. Для діагностики одного з найпоширеніших захворювань – раку – використовується світлова мікроскопія. Цей метод дослідження є джерелом цифрових цитологічних і гістологічних зображень, які містять від десятків до сотень клітин одного або декількох типів, що дає змогу в контексті комп'ютерного зору розглядати їх як зображення повторюваних елементів. Зображення повторюваних елементів є предметом аналізу в системах штучного інтелекту та широко використовуються в комп'ютерній графіці. Ці зображення можна описати так:

- 1) існує фрагмент, який регулярно повторюється в межах області, яка є великою порівняно з його розміром;
- 2) цей фрагмент утворено елементарними (непохідними) частинами, що розміщені певним чином;
- 3) елементарні частини є приблизно однаковими в межах всього зображення.

Аналіз зображень повторюваних елементів застосовують [1, 2] для опрацювання дистанційно отриманих зображень (аерокосмічних знімків), контролю якості матеріалів і тканин, оброблення документів та ін.

Нижче розглянуто основні результати в області аналізу цифрових гістологічних зображень (ЦГЗ). Вибір методів аналізу ЦГЗ зумовлений насамперед варіабельністю ознак мікрооб'єктів – площі, периметра, округлості та ін.

В області сегментації кольорових цитологічних зображень автори [3] розробили алгоритм розподілу зображення на кольорові складові, що дало змогу виділити ядро та цитоплазму. Алгоритм полягає в обчисленні внеску кожного із застосовуваних фарбників за їхнім спектром поглинання світла. Для кожного типу фарбника наперед обчислено вектор кольорів у RGB просторі.

В області класифікації типу клітин задачу постановки діагнозу, а саме висновку про наявність злякисного новоутворення, вирішено для окремих видів пухлин [4, 5]. У роботі [4] детально показано кроки постановки діагнозу. Інтерпретація у виконанні спеціаліста-патологоанатома повинна привести до збирання морфологічних доказів (числових характеристик об'єктів), що дають змогу ідентифікувати патологічні процеси, які можуть виявитися доброякісними або злякисними. Цей крок передбачає вирішення трьох різних завдань: морфологічного аналізу, ідентифікації клітин, постановки та інтерпретації діагнозу.

Питання побудови нечітких правил для постановки діагнозу на основі якісних ознак описано в роботі [4]. Умовами правил є ознаки окремих мікрооб'єктів (клітин).

Із наведеного вище видно, що задачі діагностування розв'язані для частини патологічних процесів, що розглядають клітину як мікрооб'єкт. Проте існує багато задач діагностування патологічних процесів, що базуються на аналізі характеристик шарів тканин та інших складних об'єктів. Тому актуальною є задача групування клітин у складні об'єкти на основі їх взаємного просторового розміщення, тобто структурних залежностей.

1. Зіставлення областей під час структурного аналізу зображень

Відомо, що проблема відповідності (кореспонденції) форми областей є однією з найскладніших проблем в обробці зображень не тільки через ітеративний пошук, але й чутливість до шуму [6]. В загальному випадку відповідність форми областей I та J , представленими множинами точок, задається відображенням $\forall i \in I, \exists j \in J: T(i) = j$ [7].

Для пошуку повторюваних елементів використовують [8] кореляцію знайденого елемента з зображенням для пошуку параметрів перетворення. В роботі [9] для зіставлення повторюваних елементів застосовується кореляція попередньо перетвореного елемента з елементом-кандидатом. Такий підхід успішно використовується для зображень об'єктів штучної природи. Проте мікрооб'єкти одного класу (клітини, ворсинки) на цифрових гістологічних зображеннях відрізняються за формою, кольором, текстурою і розміром у певних межах. Тому кореляційний метод зіставлення та метод на основі характерних точок контуру дають неточний результат.

Для опису структури зображення використовують такі зв'язки між сегментами: сусідство, трансляція, дотик, включення, перекриття, просторові відношення. Однією з найчастіше застосовуваних структур є граф сусідства областей, кожна вершина якого представляє певну область зображення. Дві вершини зв'язані ребром тоді і лише тоді, коли відповідні області мають спільну границю. Граф сусідства створюється на основі розмітки шляхом прослідковування границі кожної області та зберігання ідентифікаційних міток сусідніх областей або у вигляді квадродерева.

У статті [10] запропоновано 8 елементарних топологічних відношень (рис. 1,а): роз'єднані, ззовні пов'язані, однакові, частково перекриваються, дотична власна частина, інверсна дотична власна частина, не дотична власна частина, інверсна не дотична власна частина.

Система прямокутних відношень головних напрямків використовує відношення між обмежувальними прямокутниками областей [11]. Положення областей одна відносно однієї визначається напрямками сторін світу, наприклад, "північний схід" (рис. 1, б).

Недоліками простих систем просторових відношень є відсутність врахування орієнтації області та її форми.

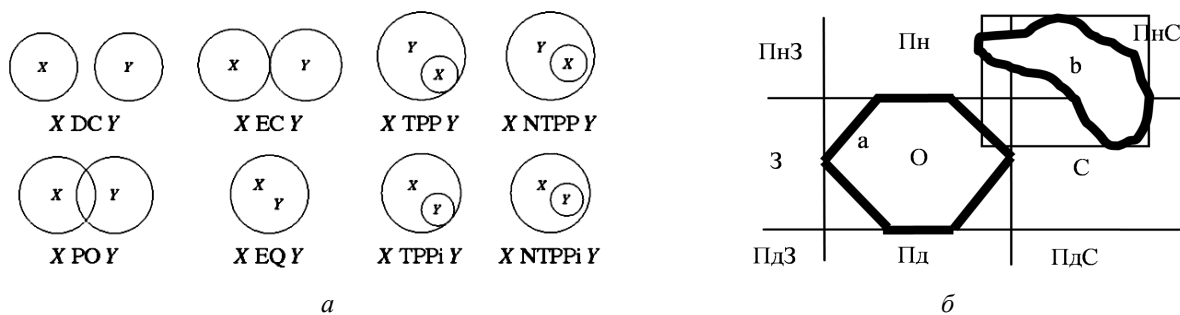


Рис. 1. Системи просторових відношень:
а – топологічні відношення між областями;
б – прямокутні відношення головних напрямків

Для усунення недоліків застосуємо підхід [12] до опису структурних залежностей на основі операції афінного перетворення T , а саме комбінації трансляції, повороту та масштабування (1)

$$[x' \ y' \ 1] = \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & 1 \end{bmatrix} [x \ y \ 1] = [T][x \ y \ 1], \quad (1)$$

де x, y – координати точки; x', y' – координати перетвореної точки; a, b, c, d – коефіцієнти зсуву і масштабування; m, n – коефіцієнти трансляції; p, q – коефіцієнти проєкцій.

Для знаходження перетворення g між двома зображеннями або областями достатньо взяти три точки, які не лежать на жодній прямій, на області A і знайти відповідні їм точки області B . Ця вимога справедлива для областей однакової форми (або для двох однакових зображень). У цьому випадку області пов'язані тільки поворотом і трансляцією.

Характерною властивістю цифрових біомедичних зображеннях є зміна форми об'єктів одного класу (клітин) та наближення їх форми до форми еліпса або до форми фігури з елементом n -кратної симетрії повороту. Тобто пряма відповідність між точками двох областей при афінному перетворенні неможлива, навіть враховуючи похибку дискретизації.

Для оцінювання відмінності областей одна від однієї використовується розмір області перекриття [13]. Для двох компактних множин A і B область перекриття M визначено як симетричну різницю

$$M = A \Delta B = (A \setminus B) \cup (B \setminus A), \quad (2)$$

де Δ – операція симетричної різниці.

Ця міра відмінності є метрикою. У випадку плоских областей одного класу з нечіткою геометричною структурою приймемо, що афінне перетворення g^* переводить область A у B так, щоб мінімізувати симетричну різницю відповідних множин точок:

$$g^* = \arg \min_g (A \Delta (g^{-1}B)). \quad (3)$$

Для знаходження трьох відповідних точок можна застосовувати простий алгоритм на основі серединного перпендикуляра [14]. Він полягає в знаходженні координат відрізка максимальної довжини. Третя точка знаходиться в місці перетину контуру і серединного перпендикуляра, побудованого на відріжку. Прикладом комбінаторного методу розв'язання задачі пошуку найкращого перетворення є метод січних прямих [6]. Він полягає у побудові прямих, які перетинають контур з деякої стартової точки через заданий кут α . Із множини точок перетину знаходять три точки, які мінімізують похибку перетворення між відповідними координатами на заданих контурах.

Вимогами до відповідних точок є стабільність їх положення на контурі – інваріантність вибору відносно повороту, трансляції, спотворень контуру тощо. Відповідні точки обираються з множини характерних точок контуру (ХТ), які відображають кривизну контуру, точки максимуму і мінімуму і т. д. Слабо виражені кутові ознаки контурів мікрооб'єктів та відмінність форми клітин вимагає додаткового застосування алгоритмів визначення найкращої відповідності між самими ХТ [7]. Повторюваність ХТ на контурі, за умов дії на область афінних перетворень, обчислюється як

відношення кількості спотворених зображень із однаково визначеними ХТ до всієї кількості тестових зображень. Для дослідження впливу ознак форми мікрооб'єктів на повторюваність ХТ використаємо ознаку округлості. Округлість області F_O розраховується так:

$$F_O = \frac{4\pi S}{P^2},$$

де S – площа області, P – периметр.

Проведено тестування алгоритму визначення точок максимальної кривизни [15] на областях різних класів. Як тестові взято зображення об'єктів із аерокосмічних зображень (АЗ), зображень документів (ЗД) та цитологічних зображень (ЦЗ1, ЦЗ2), які показані на рис. 2.

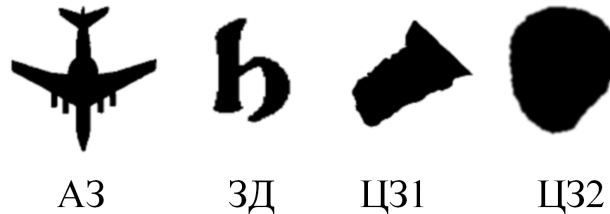


Рис. 2. Тестові зображення

Тестові зображення відрізняються кривизною контуру та ознакою округлості (табл. 1).

Таблиця 1

Повторюваність визначення ХТ

Клас зображення	АЗ	ЗД	ЦЗ1	ЦЗ2
Округлість	0,113	0,1705	0,51	0,8375
Рівень повторюваності ХТ	0,94	0,88	0,91	0,92

Відсутність кутових ознак приводить до неправильного визначення відповідних точок навіть на одному і тому самому контурі при дії повороту і трансляції на область. Для розв'язання задачі пошуку афінного перетворення можливим рішенням є обчислення моментів області, які характеризують область, проте обчислюються не на основі ознак контуру.

Метою статті є удосконалення методу визначення відповідних точок на контурах мікрооб'єктів біомедичної природи.

2. Визначення відповідних точок на основі центральних моментів

Розглянемо мікрооб'єкти на бінарному зображенні, заданому функцією

$$b(x, y) = \begin{cases} 1 & \text{об'єкт} \\ 0 & \text{фон} \end{cases}.$$

Наведемо визначення просторових моментів [16] для бінарних зображень:

$$m_{p,q} = \iint x^p y^q b(x, y) dx dy, \quad (4)$$

де p, q – індекси порядку моменту.

Моменти класифікують за їх порядком, тобто сумою $p + q$. Момент нульового порядку $((p, q) = (0,0))$ описує площу об'єкта. Моменти першого порядку $((p, q) = (1,0)$ або $(0,1))$ містять інформацію про центр мас об'єкта:

$$x_c = \frac{m_{1,0}}{m_{0,0}}, \quad y_c = \frac{m_{0,1}}{m_{0,0}}.$$

Визначення (4) описує загальні просторові моменти об'єкта. Із просторових моментів центральні моменти можна отримати скороченням за допомогою центру мас (x_c, y_c) області. Центральні моменти обчислюються так:

$$\mu_{p,q} = \iint (x - x_c)^p (y - y_c)^q b(x, y) dx dy.$$

Центральні моменти першого або вищого порядку можна безпосередньо отримати із просторових моментів:

$$\mu_{p,q} = \frac{m_{p,q}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}} \right)^p * \left(\frac{m_{0,1}}{m_{0,0}} \right)^q.$$

Центральні моменти другого порядку можна подати так:

$$\mu_{2,0} = \frac{m_{2,0}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}} \right)^2 = \frac{m_{2,0}}{m_{0,0}} - x_c^2,$$

$$\mu_{0,2} = \frac{m_{0,2}}{m_{0,0}} - \left(\frac{m_{0,1}}{m_{0,0}} \right)^2 = \frac{m_{0,2}}{m_{0,0}} - y_c^2.$$

На основі центральних моментів другого порядку обчислюється [16] значення головних осей інерції:

$$\lambda_{1,2} = \sqrt{\frac{1}{2} * (\mu_{2,0} + \mu_{0,2}) \pm \sqrt{4 * \mu_{1,1}^2 - (\mu_{2,0} - \mu_{0,2})^2}}.$$

Головні осі інерції об'єкта відповідають великій і малій осям a і b еліпса зображення, який можна використати як апроксимацію об'єкта. Головні осі інерції – це ті осі, навколо яких об'єкт може обертатися з мінімальною (велика вісь a) або максимальною (мала вісь b) інерцією. Осі a і b показано на рис. 3.

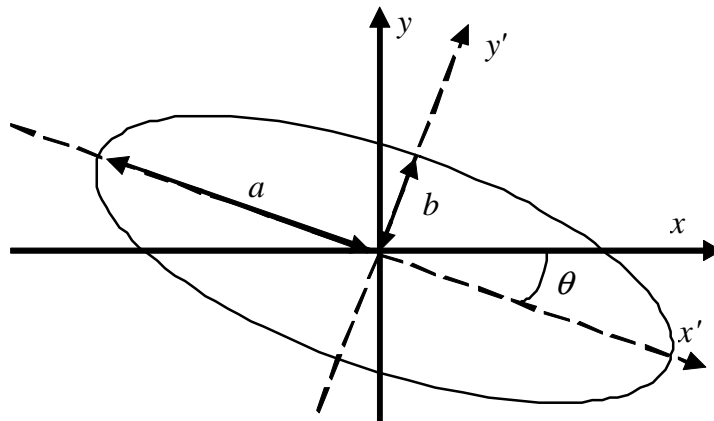


Рис. 3. Еліпс області

Орієнтація об'єкта визначається як кут нахилу осі, довкола якої об'єкт може обертатися з мінімальною інерцією, до осі абсцис (тобто кут нахилу великої осі). У цьому напрямі область має своє найбільше видовження. Орієнтація області обчислюється так:

$$\theta = \frac{1}{2} \arctg \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}.$$

Існує двозначність у визначенні кута нахилу θ положення об'єкта (рис. 3), яку вирішено шляхом вибору кута між віссю абсцис і великою віссю a . Кут нахилу θ набуває значень відповідно до табл. 2.

На основі обчислених ознак потрібно визначити три ХТ на контурі об'єкта. Ці точки повинні визначатися за правилами, що дають змогу знаходити відповідні їм точки на інших об'єктах.

Кут нахилу еліпса зображення

$\mu_{2,0} - \mu_{0,2}$	$\mu_{1,1}$	θ	
Нуль	Нуль	0°	
Нуль	Додатне	$+45^\circ$	
Нуль	Від'ємне	-45°	
Додатна	Нуль	0°	
Від'ємна	Нуль	-90°	
Додатна	Додатне	$\frac{2}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$	$0^\circ < \theta < 45^\circ$
Додатна	Від'ємне	$\frac{1}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$	$-45^\circ < \theta < 0^\circ$
Від'ємна	Додатне	$\frac{1}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} + 90^\circ$	$45^\circ < \theta < 90^\circ$
Від'ємна	Від'ємне	$\frac{1}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} - 90^\circ$	$-90^\circ < \theta < -45^\circ$

Метод визначення координат ключових точок на контурі складається з таких кроків (рис. 4).

1. Для об'єкта обчислюємо кут нахилу головної осі θ , координати центру мас об'єкта $M(x_c, y_c)$.
2. Будуємо пряму m , яка проходить через центр мас об'єкта і перетинає контур об'єкта. Точки $A(x_1, y_1)$ і $B(x_2, y_2)$ визначено так, щоб виконувалась умова $|AM| > |MB|$.
3. Для визначення третьої точки C через центр мас проводимо пряму n перпендикулярно до великої осі. У випадку простого опуклого контуру пряма n перетинає його в двох точках C і C' . Третю ключову точку C обираємо так, щоби кут між відрізком MB і MC дорівнював 90° .

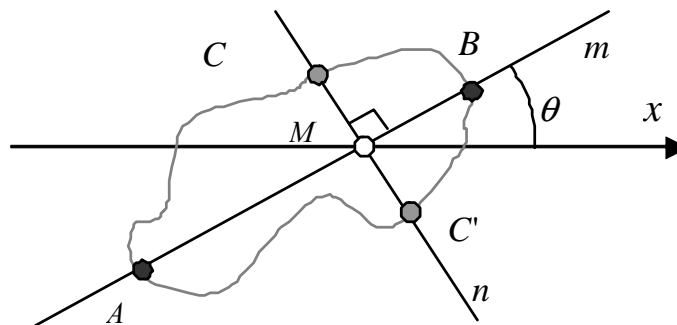


Рис. 4. Відповідні точки на контурі області

Отже, для об'єкта ідентифіковано три ХТ його контуру, на основі яких визначатимемо відповідність між об'єктами.

Схему алгоритму знаходження відповідних точок наведено на рис. 5.

Експериментальні дослідження

Загальний алгоритм аналізу ЦГЗ сформовано з таких кроків:

1. Формуємо тестові зображення із двох мікрооб'єктів шляхом афінного перетворення.
2. Кольорове зображення перетворюємо у градації сірого.
3. Розбиваємо зображення на фон та області за допомогою алгоритму порогової сегментації.
4. Виділяємо контури областей, обчислюємо координати центру мас, велику і малу вісь, кут орієнтації області.
5. Обчислюємо координати трьох відповідних точок на контурі розробленим алгоритмом, алгоритмом серединного перпендикуляру та методом січних.
6. Обчислюємо коефіцієнти афінного перетворення однієї області в іншу на основі координат відповідних точок.

7. Зіставляємо області на основі обчисленого афінного перетворення та обчислення симетричної різниці.

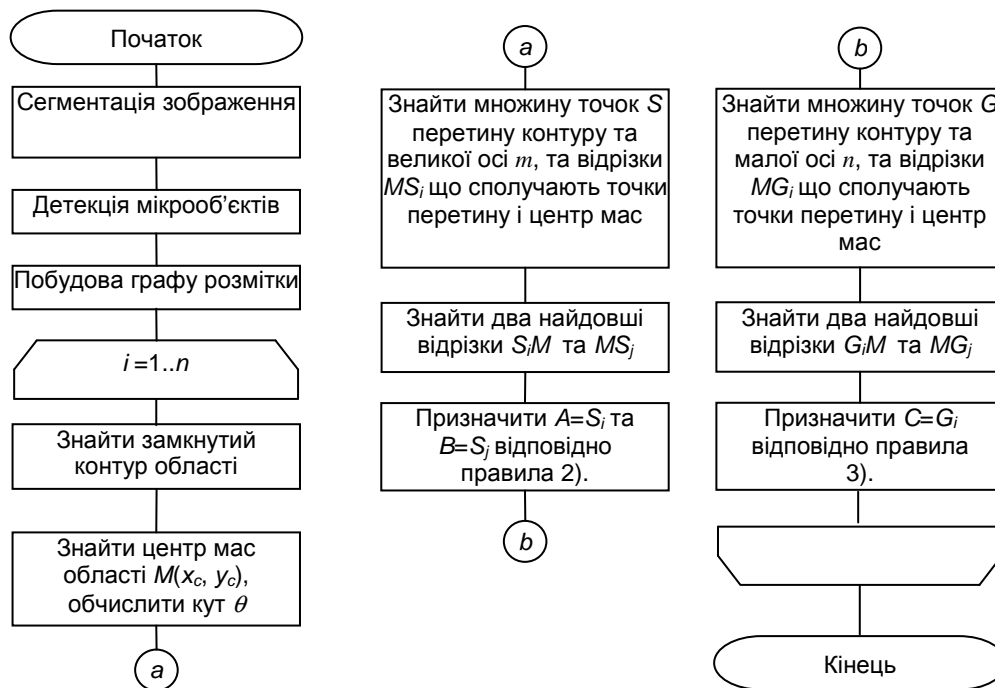


Рис. 5. Схема алгоритму знаходження відповідних точок

Параметри зображення, що впливають на результат виявлення характерних точок, такі: поворот зображення, розмиття, зміна масштабу, зменшення освітленості, збільшення коефіцієнта JPEG стиснення, збільшення піксельного шуму. За цими параметрами оцінюють алгоритми, що визначають ХТ на всьому зображенні. Цифрові гістологічні зображення обов'язково підлягають операції попереднього оброблення з вирівнювання нерівномірності фону, спричиненої нерівномірним освітленням. Тому ми не будемо оцінювати ефективність розробленого алгоритму за умови зменшення освітленості.

Для дослідження ефективності розробленого методу проведено його тестування на основі зображень окремих клітин, взятих з БД цитологічних і гістологічних зображень Breast Cancer Dataset. Зіставлено області об'єктів одного класу між собою, обчислено найкраще афінне перетворення та симетричну різницю (2). Площу перетину нормовано відносно площі одного із об'єктів. Результати тестування за дії різних комбінацій перетворень наведено на рис. 6.

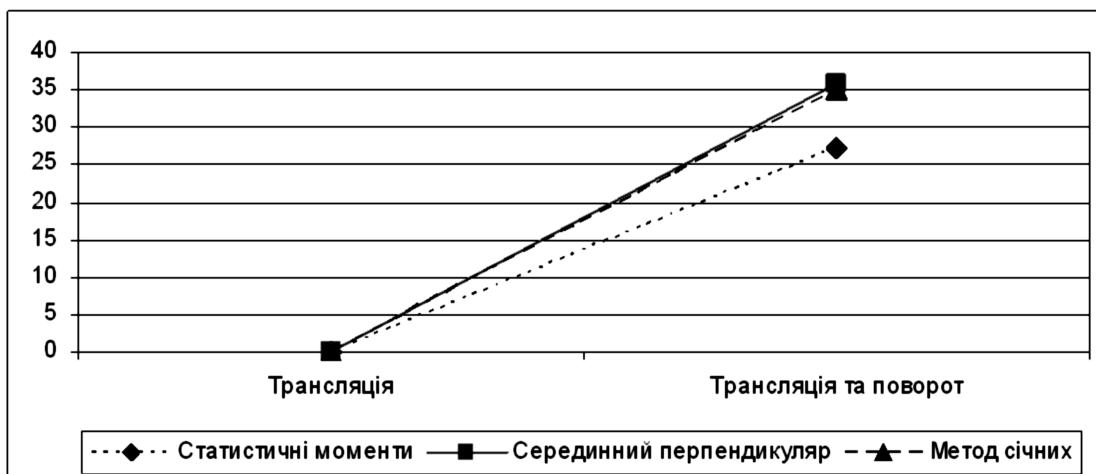


Рис. 6. Величина симетричної різниці для різних алгоритмів

Метод січних з кроком збільшення кута побудови на 10° потребує на порядок більше часу на перебір всіх можливих матриць перетворень, ніж розроблений метод. Розроблений метод визначення відповідних точок на основі центральних моментів виявляється кращим при дії цілочисельної трансляції та повороту.

Висновки

Результати експериментів підтверджують доцільність застосування методу визначення відповідних точок на основі дискретних центральних моментів під час розв'язання задачі відповідності двох областей одного класу з різною формою. Основна перевага центральних моментів, а саме їх інваріантність до трансляції і повороту області, дала змогу зменшити величину перекриття зіставлених областей за допомогою розробленого методу на 8 % порівняно із методом січних.

1. *Handbook of Texture Analysis* / Ed. by Majid Mirmehdi, Xianghua Xie, Jasjit Suri. – London, UK: Imperial College Press, 2008. – 413. P. 2. Tuceryan M. *Handbook of Pattern Recognition and Computer Vision* / M. Tuceryan, A.K. Jain // *Handbook of Pattern Recognition and Computer Vision (2 nd Edition)* / Ed. by C. H. Chen, L.F. Pau, P.S.P. Wang. – World Scientific Publishing Co., 1998. – P. 207–249. 3. Ruifrok A. C. *Quantification of histochemical staining by color deconvolution* / A. C. Ruifrok, D. A. Johnston // *Analytical and quantitative cytology and histology. The International Academy of Cytology and American Society of Cytology*. – 2001. – Vol. 23, no. 4. – P. 291–299. 4. Ovalle A. *Kids: A distributed expert system for biomedical image interpretation*. / A. Ovalle, Catherine Garbay // *Information Processing in Medical Imaging* / Ed. by Alan C. F. Colchester, David J. Hawkes. – Vol. 511 – 1991. – P. 419–433. 5. *Characterization of chromatin texture by contour complexity for cancer cell classification* / T. Kiyuna, A. Saito, E. Kerr, W. Bickmore // *8th IEEE International Conference on BioInformatics and BioEngineering*, 2008. – Athens, 2008. – P. 1–6. 6. Путятин Е.П. *Обработка изображений в робототехнике* / Е. П. Путятин, С. И. Аверин. – М.: Машиностроение, 1990. – 320 с. 7. *Contour correspondence via ant colony optimization* / O. van Kaick, G. Hamarneh, H. Zhang, P. Wighton // *Proc. 15th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2007)* – Maui, Hawaii, United States, 2007. – P. 271–280. 8. Schaffalitzky F. *Geometric grouping of repeated elements within images* / F. Schaffalitzky, A. Zisserman // *In Proc. 9th British Machine Vision Conference (BMVC), Southampton, September 1998*. – Southampton, UK: Springer-Verlag, 1998. – September. – P. 13–22. 9. Liu Y. *A computational model for periodic pattern perception based on frieze and wallpaper groups* / Yanxi Liu, Robert T. Collins, Yanghai Tsin // *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. – 2004. – Vol. 26, No. 1. – P. 354–371. 10. Wolter F. *Spatio-temporal representation and reasoning based on rcc-8* / Frank Wolter, Michael Zakharyashev // *In Proceedings of the seventh Conference on Principles of Knowledge Representation and Reasoning (KR2000)*. – Breckenridge, Colorado, USA: Morgan Kaufmann, 2000. – P. 3–14. 11. Navarrete I. *Consistency checking of basic cardinal constraints over connected regions* / Isabel Navarrete, Antonio Morales, Guido Sciavicco // *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. – 2007. – P. 495–500. 12. Березский О.Н. *Теоретико-групповой подход к анализу симметричных изображений* / О. Н. Березский // *Управляющие машины и системы*. – 2010. – Т. 2. – С. 16–24. 13. Veltkamp R. C. *State of the art in shape matching* / Remco C. Veltkamp, Michiel Hagedoorn // *Principles of visual information retrieval*. – 2001. – P. 87–119. 14. Березский О.Н. *Алгоритмы анализа и синтеза биомедицинских изображений* / О. Н. Березский // *Проблемы информатики и управления*. – 2007. – № 2. – P. 134–144. 15. Chetverikov D. *A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves* / Dmitry Chetverikov / Ed. by *Computer Analysis of Images, Patterns*. – 2004. – Vol. 2756/2003 – P. 746–753. 16. Kilian J. *Simple image analysis by moments*. *Opencv library documentation [Електронний ресурс]* / J. Kilian – 2001. – Режим доступу до статті.: <http://public.cranfield.ac.uk/~c5354/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf>

О. Овсяк*, В. Овсяк, Ю. Петрушка

*Київський національний університет культури і мистецтв, Львівська філія,
Українська академія друкарства**НЕСУПЕРЕЧНІСТЬ АЛГЕБРИ СЕКВЕНЦІЙНИХ АЛГОРИТМІВ**

© Овсяк О., Овсяк В., Петрушка Ю., 2012

Несуперечність алгебри секвенційних алгоритмів, за умов логічних значень змінних і функціональних змінних та наявності тільки одного індексу порядку, доведена зведенням операцій алгебри секвенційних алгоритмів до операцій несуперечної логіки предикатів.

Ключові слова: алгебра алгоритмів, несуперечність, логіка предикатів, операція, предикат.

Sequential consistency algebra algorithms under logical variables and functional variables and there is only one index procedure proved mixing operations algebra sequential algorithms for operations consistent logic.

Key words: algebra algorithms, consistency, logic predicates, operation predicate.

Вступ і формулювання задачі

Аксиоматичний метод [1] використовується для дефініції наукових теорій запровадженням їх початкових положень, які утворюють систему аксіом. Дуже важливим питанням, яке ставиться до побудованих аксиоматичним методом теорій, є питання їх несуперечності [2–4]. Система аксіом вважається суперечливою, якщо з неї виводиться формула та її заперечення [2, 3] або більш загально – вивідність кожної формули цієї системи [4]. Відомі методи доведення несуперечності побудованою моделі її або безпосереднього доведення несуперечності з використанням теорії доведень [5]. Доведення несуперечності теорії будь-яким з цих методів, тією чи іншою мірою, використовує засоби іншої завідомо несуперечної теорії [4]. Власне тому несуперечність досліджуваної теорії може бути доведена зведенням її до відомої несуперечної [4].

Відомі алгебра секвенційних алгоритмів [6] та її модифікація [7–9]. Означені вони з використанням аксиоматичного методу. У зв'язку з цим для алгебри секвенційних алгоритмів та її модифікації постає питання дослідження їхньої несуперечності. Розв'язання цієї задачі і є предметом статті.

Дослідження несуперечності алгебри секвенційних алгоритмів

Алгебра секвенційних алгоритмів утворена операціями секвентування, елімінування, паралелення, реверсування та циклічних секвентування, елімінування і паралелення [6].

Операції виконуються над знаками, якими можуть бути цифри, числа, логічні значення, змінні, функції абстрактні і предметні знаки і послідовності знаків алфавітів мов людського спілкування і технічних мов тощо.

Обов'язковими знаками є логічні значення і логічні змінні, які використовуються в операції елімінування. У зв'язку з цим, що теорія предикатів оперує логічними значеннями, змінними і предикатами, вона є несуперечною [2, 3]. Доцільно у частковому випадку звести алгебру секвенційних алгоритмів до теорії предикатів, що і буде зроблено.

Теорема. Алгебра секвенційних алгоритмів несуперечлива.

Доведення. Відомо [2, 3], що логіка предикатів є несуперечною. Для зведення алгебри секвенційних алгоритмів до логіки предикатів необхідно зробити два обмеження. По-перше,

допустити у ролі знаків, над якими будуть виконуватися операції алгебри алгоритмів, тільки логічні знаки 0 та 1, логічні змінні та предикати. По-друге, увести обмеження на кількість індексів порядку α та β логічних значень, логічних змінних і предикатів. Допускаємо тільки індекс порядку α . З метою спрощення отримуваних виразів індекс порядку α опускаємо.

Для двох індексів порядку α та β і двох змінних означення операцій секвентування, паралелення й реверсування над логічними змінними x і y матиме вигляд, наведений у табл. 1 (Зм. – змінні та індекси і № – порядкові номери комбінацій значень змінних).

Таблиця 1

Істинносне означення операцій логічного секвентування, паралелення і реверсування для двох змінних і двох індексів порядку

Зм. №	Індекс α Змінні		Індекс β Змінні		$\overline{x_\alpha y_\alpha}$	$\overline{x_\alpha y_\beta}$	$\overline{x_\alpha y_\alpha}$	$\overline{x_\alpha y_\beta}$	$\overline{\overline{x_\alpha y_\beta}}$	$\overline{\overline{x_\alpha y_\alpha}}$	$\overline{y_\alpha x_\beta}$
	x_α	y_α	x_β	y_β							
0	0	0	0	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	1	0
2	0	0	1	0	0	0	0	0	1	1	0
3	0	0	1	1	0	0	0	1	0	1	0
4	0	1	0	0	0	0	1	0	1	0	0
5	0	1	0	1	0	0	1	1	0	0	0
6	0	1	1	0	0	0	1	0	1	0	1
7	0	1	1	1	0	0	1	1	0	0	1
8	1	0	0	0	0	0	1	1	0	0	0
9	1	0	0	1	0	1	1	1	0	0	0
10	1	0	1	0	0	0	1	1	0	0	0
11	1	0	1	1	0	1	1	1	0	0	0
12	1	1	0	0	1	0	1	1	0	0	0
13	1	1	0	1	1	1	1	1	0	0	0
14	1	1	1	0	1	0	1	1	0	0	1
15	1	1	1	1	1	1	1	1	0	0	1

З уведеним спрощенням, яке полягає у наявності одного індексу порядку (тільки індекс порядку α), табл. 1 трансформується до табл. 2.

Таблиця 2

Істинносне означення операцій логічного секвентування, паралелення і реверсування для двох змінних і одного індексу порядку

Зм. №	Індекс α Змінні		$\overline{x_\alpha y_\alpha}$	$\overline{x_\alpha y_\alpha}$	$\overline{\overline{x_\alpha y_\alpha}}$	$\overline{\overline{y_\alpha x_\alpha}}$
	x_α	y_α				
0	0	0	0	0	1	1
1	0	1	0	1	0	1
2	1	0	0	1	0	1
3	1	1	1	1	0	0

У табл. 3 наведено істинносне означення логічних операцій кон'юнкції, диз'юнкції та інвертування.

Таблиця 3

Істинносне означення логічних операцій

	x	y	$x \& y$	$x \vee y$	$\overline{x \vee y}$	$\overline{x \& y}$
0	0	0	0	0	1	1
1	0	1	0	1	0	1
2	1	0	0	1	0	1
3	1	1	1	1	0	0

Порівнюючи табл. 2 і 3, бачимо, що

$$\overline{x_a y_a} = x \& y, \text{ а } \overline{x_w y_a} = x V y \text{ та } \overline{\overline{x_w y_a}} = \overline{x V y} \text{ і } \overline{\overline{x_a y_a}} = \overline{x \& y}.$$

Отже, для заданих умов операція секвентування алгебри алгоритмів є операцією кон'юнкції логіки предикатів, а операція паралелення – операцією диз'юнкції і операція реверсування – операцією інвертування.

Означення операції елімінування над логічними змінними наведено у табл. 4.

Таблиця 4

Операція логічного елімінування над істинносними змінними

Зм. №	x	y	u	$\overline{x; y; u-?}$	$\overline{x; y; 0-?}$	$\overline{x; y; 1-?}$	$\overline{x; y; *-?}$
0	0	0	0	0	0	0	$\overline{0,0}$
1	0	0	1	0	0	0	$\overline{0,0}$
2	0	1	0	1	1	0	$\overline{0,1}$
3	0	1	1	0	1	0	$\overline{0,1}$
4	1	0	0	0	0	1	$\overline{1,0}$
5	1	0	1	1	0	1	$\overline{1,0}$
6	1	1	0	1	1	1	$\overline{1,1}$
7	1	1	1	1	1	1	$\overline{1,1}$

Тепер, дамо істинносне означення формул логіки предикатів у вигляді табл. 5.

Таблиця 5

Формули логіки предикатів

Зм. №	x	y	u	$x \& u V y \& \bar{u}$	$x \& 0 V y \& \bar{0}$	$x \& 1 V y \& \bar{1}$	$x V y$
0	0	0	0	0	0	0	$0V0$
1	0	0	1	0	0	0	$0V0$
2	0	1	0	1	1	0	$0V1$
3	0	1	1	0	1	0	$0V1$
4	1	0	0	0	0	1	$1V0$
5	1	0	1	1	0	1	$1V0$
6	1	1	0	1	1	1	$1V1$
7	1	1	1	1	1	1	$1V1$

Порівнюючи табл. 4 і 5, бачимо, що справедливі такі рівності:

$$\overline{x; y; u-?} = x \& u V y \& \bar{u}, \overline{x; y; 0-?} = x \& 0 V y \& \bar{0}, \overline{x; y; 1-?} = x \& 1 V y \& \bar{1} \text{ і } \overline{x; y; *-?} = x V y.$$

З цих рівностей випливає, що для заданих умов операція елімінування алгебри алгоритмів є замінимою формулою логіки предикатів.

Для заданих умов порівнюємо операції циклічних секвентування, паралелення і елімінування алгебри алгоритмів із кванторними операціями загальності та існування логіки предикатів.

На підставі властивостей операції циклічного секвентування [6–9] маємо

$$\overline{\overline{\overline{x F(x) = F(i): F(j): F(k): \dots}}}, \text{ для } x \in Q = \overline{i: j: k: \dots}$$

Отже, ця операція є узагальненням операції секвентування на нескінченну кількість значень предметної змінної функціонального унітерму.

Відомо [2], що кванторна операція для усіх є узагальненням операції кон'юнкції на нескінченну кількість значень предметної змінної предиката.

Оскільки для заданих умов операція секвентування алгебри алгоритмів є операцією кон'юнкції, то узагальнення операції секвентування дорівнюватиме узагальненню операції кон'юнкції на нескінченну кількість значень предметної змінної. Отже, для заданих умов операцію циклічного секвентування алгебри алгоритмів можна замінити операцією загальності логіки предикатів.

Аналогічно замінимості операцій циклічного секвентування і кванторної операції загальності встановлюється замінимість операції циклічного паралелення алгебри алгоритмів і кванторної операції існування логіки предикатів.

Операція циклічного елімінування є узагальненням на нескінченну кількість значень предметної змінної операції елімінування, що видно на підставі властивості

$$\exists u_x F(x) = \overbrace{F(i); F(j); F(k); \dots u_k-?; u_j-?; u_i-?}^{\dots}, \text{ для } x \in Q = \overbrace{i; j; k; \dots}^{\dots}$$

Крім того, операцію циклічного елімінування можна замінити формулою

$$\exists u_x F(x) = \overbrace{\overbrace{\Phi u_x u_x; F(x)}^{\dots}, \overbrace{u_x; F^1(x)}^{\dots}}$$

у якій функціональний унітерм $F^1(x)$ є формулою паралелення, яка розкривається з розкриттям циклу операції циклічного паралелення. Операцію циклічного паралелення алгебри алгоритмів, як це було показано вище, можна замінити кванторною операцією існування логіки предикатів, а операції секвентування і паралелення – операціями кон'юнкції і диз'юнкції відповідно. Отож, операцію циклічного елімінування для заданих умов можна замінити операціями логіки предикатів.

На підставі виконаного аналізу встановлено, що для логічних значень змінних і функціональних унітермів та одного індексу порядку можна замінити усі операції алгебри алгоритмів операціями логіки предикатів. Оскільки логіка предикатів є несуперечливою, то несуперечливою є і алгебра алгоритмів. Теорему доведено.

Теорема. Модифікована алгебра секвенційних алгоритмів несуперечлива.

Доведення несуперечливості модифікованої алгебри алгоритмів аналогічне доведенню несуперечливості алгебри алгоритмів.

Висновок

Алгебра секвенційних алгоритмів та модифікована алгебра секвенційних алгоритмів є несуперечливими.

1. Математическая энциклопедия. – Т. 1. – М.: Советская энциклопедия, 1977. – 1152 с.
2. Гильберт Д., Бернайс П. Основания математики. Логические исчисления и формализация арифметики. – М.: Мир, 1982. – 556 с.
3. Новиков П.С. Элементы математической логики. – М.: Наука, 1973.
4. Математическая энциклопедия. – Т.3. – М.: Советская энциклопедия, 1982. – 1184 с.
5. Такеути Г. Теория доказательств. – М.: Мир, 1978. – 412 с.
6. Овсяк В.К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем // Доповіді Національної академії наук України, № 9, 1996. – С. 83–89.
7. Owsiak W., Owsiak A. Rozszerzenie algebry algorytmów // Pomiar, automatyka, kontrola. – 2010. – № 2. – S. 184–188.
8. Овсяк О.В. Мінімізація формули алгоритму транслятора електромеханічних схем друкарських машин // Збірник праць наук.-техн. конф. “Обчислювальні методи і системи перетворення інформації”, 7–8 жовтня 2010. – Львів: ФМІ, 2010. – С. 256–259.
9. Ovsyak A., Ovsyak V. The extended algebra of algorithms with additional cycle elimination axioms // Conference "Intelligent Information and Engineering Systems" (INFOS 2011), September 19–23, 2011, Polańczyk, Poland. – P. 23–34.

А. Ковальчук, Д. Пелешко, *Ю. Борзов
Національний університет “Львівська політехніка”,
кафедра ІТВС,

*Львівський державний університет безпеки життєдіяльності,

ВИКОРИСТАННЯ ПОБІТОВИХ ОПЕРАЦІЙ І ДОДАТКОВОГО ЗАШУМЛЕННЯ В АЛГОРИТМІ RSA ПРИ ШИФРУВАННІ–ДЕШИФРУВАННІ ЗОБРАЖЕНЬ

© Ковальчук А., Пелешко Д., Борзов Ю, 2012

Вступ

Алгоритм RSA є одним із найуживаніших промислових стандартів шифрування сигналів. Відносно зображення існують певні проблеми його шифрування, а саме частково зберігаються контури на різко флюктуаційних зображеннях [4, 5].

Проблема несанкціонованого використання зображень на найнижчому рівні вирішується положеннями про авторське право, а на найвищому – методами криптографії і стеганографії, поліграфічними сітками тощо.

Оскільки зображення є одними із найуживаніших видів інформації в сучасному інформаційному суспільстві, то актуальним завданням є захист зображень від несанкціонованого доступу та використання.

Зображення є сигналом, який володіє не лише типовою інформативністю даних, а й візуальною інформативністю. А остання привносить в питання захисту нові задачі.

Така інформативність з розвинутими сучасними методами обробки зображень дає можливість для організації несанкціонованого доступу. Фактично організація атаки на зашифроване зображення можлива у двох варіантах: традиційним зламом методів шифрування або за допомогою методів візуальної обробки зображень (методи фільтрації, виділення контурів тощо). У зв'язку з цим до методів шифрування у випадку їх використання стосовно зображень висувається ще одне завдання – повна зашумленість зашифрованого зображення. Це потрібно для того, щоб унеможливити використання методів візуальної обробки зображень.

Мета роботи

Стосовно зображень актуальною задачею є розроблення модифікації методу RSA для забезпечення повної зашумленості зображення з метою унеможливлення використання методів візуальної обробки зображень і збереження криптографічної стійкості.

Одним із шляхів розв'язання цієї задачі є використання побітових операцій в алгоритмі RSA з додатковим зашумленням в програмній реалізації.

Характеристики зображення

Цифрове зображення – масив даних, отриманий шляхом дискретизації (аналоого-цифрового перетворення) оригіналу. Будучи закодованим за допомогою особливого алгоритму і записаним на носій, цей масив даних стає файлом.

У сучасному процесі поліграфічного виробництва всі ілюстрації й елементи оформлення представлені цифровими зображеннями різних типів. Цифрові зображення за способом дискретизації оригіналу поділяються на растрові, векторні та змішаного типу.

До растрових зображень належать двовимірні масиви даних (матриці пікселів), кожен елемент яких представляє ділянку оригіналу з усередненим колірним показником.

Найменшими елементами векторного зображення є вектор і крива Безьє. Вектор у комп'ютерній графіці – це відрізок, що з'єднує дві точки з заданими координатами. Основним керівним елементом кривої Безьє є вузол (node), також званий контрольною точкою (CP, control point) або контрольною вершиною (CV, control vertex). Ступінь кривизни лінії визначається координатами вузла і двох керівних точок.

Цифрові зображення змішаного типу являють собою масиви даних, що містять інформацію як у вигляді матриці пікселів, так і у вигляді опису векторів, кривих Безьє, примітивів і текстових блоків.

Нехай задано рисунок P з ширини l і висоти h . Його можна розглядати як матрицю інтенсивностей пікселів

$$C = \begin{pmatrix} c_{1,1} & \dots & c_{1,m} \\ \dots & \dots & \dots \\ c_{n,1} & \dots & c_{n,m} \end{pmatrix} \quad (1)$$

розмірності

$$n = n(l) \text{ і } m = m(h), \quad (2)$$

де c_{ij} – значення інтенсивності у напівтонових зображень піксела dtp_{ij} . Тобто існує відповідність [1]

$$P = \mathbf{P}_{l,h} = [pxl_{ij}]_{1 \leq i \leq n(l), 1 \leq j \leq m(h)} \rightarrow \mathbf{C} = [c_{ij}]_{1 \leq i \leq n(l), 1 \leq j \leq m(h)}. \quad (3)$$

Під градацію яскравості зазвичай надається 1 байт, причому 0 – чорний колір, а 255 – білий (максимальна інтенсивність).

Важливою характеристикою зображення є наявність у зображенні контурів [1]. Задача виділення контура вимагає використання операцій над сусідніми елементами, які є чутливими до змін і затемнюють області постійних рівнів яскравості, тобто контури – це ті області, де виникають зміни, стаючи світлими, тоді як інші частини зображення залишаються темними [2].

Математично ідеальний контур – це розрив просторової функції рівнів яскравості в площині зображення. Тому виокремлення контуру означає пошук найбільш різких змін, тобто максимумів модуля вектора градієнта [2]. Це є однією з причин того, що контури залишаються в зображенні при шифруванні в системі RSA, оскільки шифрування тут ґрунтується на піднесенні до степеня за модулем деякого натурального числа. При цьому на контурі і на сусідніх до контуру пікселях піднесення до степеня значення яскравостей дає ще більший розрив.

Опис модифікації

Шифрування і дешифрування за одним рядком матриці зображення

Нехай P, Q – пара довільних простих чисел і $N = P * Q$. Шифрування відбувається поелементно з використанням такого перетворення елементів матриці зображення C :

1. Випадково вибирається натуральне число $e < \varphi(N)$ і знаходиться таке натуральне d , що виконується конгруенція $ed \equiv 1 \pmod{\varphi(N)}$.

2. Будується число $A = (e \ll k) + (d \ll l) + (e \ll l) + (d \ll k)$, де $k < 16$, $l < 16$ – натуральні числа, $k \neq l$, \ll – операція логічного зсуву ліворуч.

3. У кожному рядку виконується логічний зсув ліворуч значення інтенсивності i -го пікселя, $i = 1, 2, \dots, m$, m – число елементів у рядку за таким правилом: виконується логічний зсув ліворуч значення інтенсивності пікселя на величину $i \bmod n$, $n < 16$.

4. Будується число B відніманням від отриманого значення інтенсивності пікселя числа $(A + e)$.

5. Зашифрованим значенням інтенсивності i -го пікселя, $i = 1, 2, \dots, m$, m – число елементів у рядку, вибирається число $C \equiv B^e \pmod{N} + f(i^2)$.

Дешифрування проводиться в порядку, протилежному до шифрування після отримання числа $(C - f(i^2))^d \equiv (B^e)^d \pmod{N}$, виконанням протилежних до змісту пунктів 4, 1 операцій.

Результати наведені на рис. 1–3.



Рис. 1. Початкове зображення

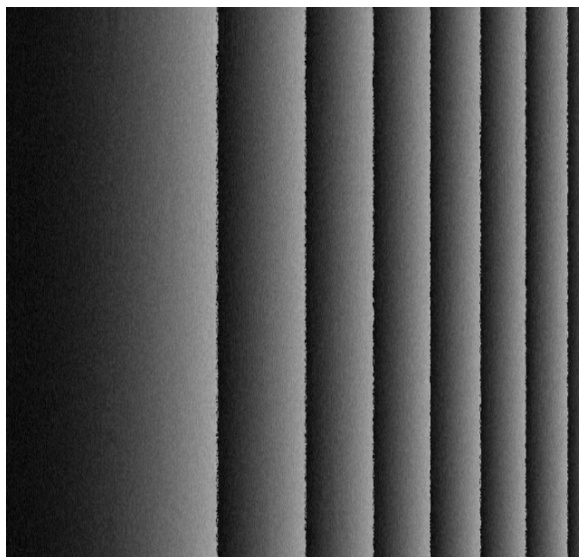


Рис. 2. Зашифроване зображення



Рис. 3. Дешифроване зображення

Шифрування за двома рядками матриці

Шифрування відбувається з використанням елементів двох рядків за алгоритмом, який описано вище, для шифрування елементів одного рядка інтенсивностей, за винятком п.5, причому кожний рядок з вибраних двох рядків шифрується незалежно за своїм алгоритмом, для нього модифікованим п.5.

Пункт 5 має вигляд:

5.1. Для першого рядка зашифрованим значенням інтенсивності i -го пікселя, $i = 1, 2, \dots, m$, m – число елементів у рядку, вибирається число $C \equiv B^e \pmod{N} + g(i^2)$.

5.2. Для другого рядка зашифрованим значенням інтенсивності i -го пікселя, $i = 1, 2, \dots, m$, m – число елементів у рядку, вибирається число $C \equiv B^e \pmod{N} - g(i^2)$.

Дешифрування відбувається в протилежному порядку з урахуванням п.п. 5.1, 5.2.

Результати наведено на рис. 4–6.



Рис. 4. Початкове зображення

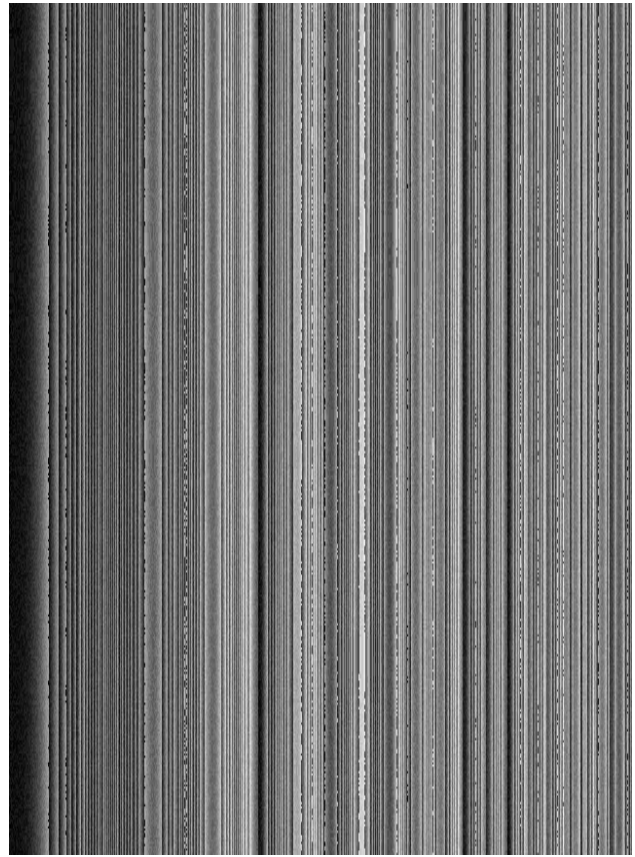


Рис. 5. Зашифроване зображення



Рис. 6. Дешифроване зображення

Порівнюючи рис. 2 і 5, бачимо, що шифрування за одним рядком матриці (3) істотно не відрізняється від шифрування за двома рядками цієї матриці. Контури в обох зашифрованих зображеннях відсутні. Початкові і дешифровані зображення тільки незначно відрізняються за рівнем яскравості

Висновки

1. Запропоновані модифікації шифрування призначені для шифрування зображень в градаціях сірого і ґрунтуються на використанні ідей базового алгоритму RSA.

2. Запропоновані модифікації можна використовувати стосовно будь-якого типу зображень, але найбільших переваг досягають у випадку використання зображень, які дають змогу чітко виділяти контури.

3. Обидва типи модифікацій без жодних застережень можна використати і стосовно кольорових зображень. Однак, незалежно від типу зображення, пропорційно до розмірності вхідного зображення може зрости розмір шифрованого зображення.

4. Стійкість до несанкціонованого дешифрування запропонованою потоковою модифікацією забезпечує алгоритм RSA.

1. Никулин Е.А. *Компьютерная геометрия и алгоритмы машинной графики.* – СПб.: БХВ-Петербург, 2005. – 560 с. 2. Яне Б. *Цифровая обработка изображений.* – М.: Техносфера, 2007. – 583 с. 3. Шнайер Б. *Прикладная криптография.* – М.: Триумф, 2003. – 815 с. 4. Рашкевич Ю.М., Пелешко Д.Д., Ковальчук А.М., Пелешко М.З. *Модифікація алгоритму RSA для деяких класів зображень // Технічні вісті.* – 2008/1(27). – 2(28). – С. 59–62. 5. Rashkevych Y., Kovalchuk A., Peleshko D., Kupchak M. *Stream Modification of RSA Algorithm For Image Coding with precize contour extraction // Proceedings of the X-th International Conference CADSM 2009. 24–28 February 2009, Lviv–Polyana, Ukraine.* – P. 469–473.

В. Литвин, М. Бойчук

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ПОСТАНОВКА ЗАДАЧІ ОЦІНЮВАННЯ НОВИЗНИ ОНТОЛОГІЧНИХ ЗНАНЬ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

© Литвин В., Бойчук М., 2012

Розглянуто формалізацію процесу оцінювання новизни онтологічних знань з метою побудови ефективних процедур розв’язування задач інтелектуальною системою.

Ключові слова: онтологія, оцінка новизни знань, інтелектуальна система, база знань.

In the paper the formalization of the evaluation process innovation ontological knowledge to build effective procedures for solving problems of intelligent systems..

Keywords: ontology, knowledge innovation, intelligent system, knowledge base.

Постановка проблеми у загальному вигляді

Основною компонентою інтелектуальних систем (ІнтС) є база знань (БЗ), що формується відповідно до предметної області (ПО), на яку зорієнтоване функціонування цієї системи [1]. Призначення цієї компоненти полягає у зберіганні, впорядкуванні та керуванні інформацією про ПО та задачі, які в ній виникають. Найважливіший параметр БЗ – якість та повнота знань про ПО, яку вона задає. Якість БЗ залежить від структури та формату знань, способу їх подання. Для широкого впровадження будь-якої технології чи методики необхідний чіткий і аргументований стандарт. Традиційні методи інженерії знань (отримання знань від експерта, інтелектуальний аналіз даних, машинне навчання тощо) не ґрунтуються на системі вивіренених та загальноприйнятих стандартів, тому побудовані на їхній основі бази знань з часом втрачають свою функціональність через низьку ефективність їх функціонування. Як стандарт інженерії знань використовують онтологічний інжиніринг, у результаті застосування якого отримують онтологію бази знань. Онтологія – це детальна формалізація деякої області знань, подана за допомогою концептуальної схеми. Така схема складається з ієрархічної структури понять, зв’язків між ними, теорем та обмежень, які є прийняті у певній ПО.

Використання онтологій у складі БЗ ІнтС допомагає вирішити низку проблем методологічного та технологічного характеру, які виникають під час розроблення таких систем. Зокрема для України характерні проблеми відсутності концептуальної цілісності й узгодженості окремих прийомів та методів інженерії знань; нестачі кваліфікованих фахівців у цій галузі; жорсткості розроблених програмних засобів та їх низькій адаптивній здатності; складності впровадження ІнтС, що зумовлено психологічними аспектами.

Ефективність онтології на пряму залежить від новизни знань, які до неї додаються. Постає завдання оцінювання знань, які пропонується додавати в онтологію БЗ ІнтС, тобто побудови деякої міри знань, подібної до відомої міри інформації, яку ввів Шеннон [2]. Однак, на відміну від інформації, для знань необхідно визначити ефект від її використання. Отже, метою цієї роботи є розроблення формальної постановки задачі оцінювання ефективності нових знань.

Аналіз останніх досліджень та публікацій

Види кількісного оцінювання інформації можна поділити на три групи: синтаксичні, семантичні та прагматичні.

1. *Синтаксичне вимірювання кількості інформації* ґрунтується на взаємозв’язку між інформацією і ентропією – мірою невизначеності. Якщо апріорно ситуація характеризується ентропією H_0 ,

а після одержання повідомлення S ентропія зменшилася до H_1 , то кількість інформації, що міститься у повідомленні S (позначення I_S), визначається різницею

$$I_S = H_0 - H_1. \quad (1)$$

Отже, кількість інформації вимірюється зменшенням невизначеності ситуації у результаті отримання повідомлення S . Що більшим є число можливих повідомлень, то більшою була апіорна невизначеність H_0 і то більшу кількість інформації одержує адресат, коли ця невизначеність знижується.

Величина ентропії на один елемент повідомлення визначається за формулою К. Шеннона:

$$H = -\sum_{i=1}^m p_i \log p_i, \quad (2)$$

де m – кількість можливих станів елемента повідомлення; p_i – ймовірність того, що елемент перебуває в i -му стані.

2. Семантичні міри інформації. Ця категорія мір призначена для вимірювання змісту I_{sem} повідомлень, які одержує адресат інформації. Оскільки зміст повідомлень кожний конкретний їх адресат сприймає по-різному (залежно від об'єму інформації, якою він володіє), то саме “запас знань” адресата і покладено в основу семантичних мір інформації. Отже, семантичні міри враховують не тільки зміст повідомлень, але і те, який об'єм нової інформації несе повідомлення його адресату порівняно з тим, що він вже знав.

Методи кількісного вимірювання змісту інформації ще достатньо не розроблено. Найбільшого визнання здобула запропонована Ю.І. Шнейдером тезаурусна міра, у якій семантична ємність I_{sem} інформації пов'язується із здатністю її одержувача сприймати (асимілювати) відомості, що надходять, а це залежить від об'єму знань адресата інформації, або, як прийнято говорити, від його тезауруса.

Максимального значення семантичність інформації набуває за погодженості тезаурусів, коли інформація, що надходить, по-перше, є зрозумілою адресату і, по-друге, несе йому невідомі раніше (відсутні у його тезаурусі) відомості.

Отже, кількість семантичної інформації у відомостях (тобто кількість нових знань, що їх отримує адресат) є величиною відносною: одне й те саме повідомлення може мати зміст для компетентного і бути беззмістовним (семантичним шумом) для некомпетентного адресата. Водночас інформація, яка хоча є зрозумілою, але відомою компетентному адресатові, є для нього також семантичним шумом. Іншими словами, відомості, які не є новими, несуть нульову кількість інформації.

Отже, зміст інформації можна вимірювати ступенем зміни тезауруса адресата інформації під впливом отриманого повідомлення.

У зв'язку з цим дуже важливими є задачі моделювання тезаурусів різних інформаційних систем, розроблення взаємоузгоджених тезаурусів у системі “людина-машина”, вирішення питань про те, яким повинен бути тезаурус у людей, що займаються тією чи іншою проблемною сферою (економікою, фінансами, математикою, програмуванням, інженерним проектуванням, медициною, біологією тощо).

3. Прагматичні міри інформації. Ці міри інформації відображають цінність інформації, її корисність і доцільність для досягнення поставленої мети.

Прагматичні міри, як і семантичні, є відносними, і вибір міри зумовлюється особливостями використання певної інформації у певній системі.

Як правило, цінність інформації залежить від того, ким і з якою метою вона використовується. Наприклад, повідомлення, що завтра буде значна хмарність, має різну цінність для торговельної організації і для авіації.

Відомі два основні методи визначення кількісної міри цінності інформації. Якщо мета напевне є досяжною і до того ж декількома можливими шляхами, то зручною є міра цінності, запропонована Р.Л. Стратоновичем. Вона полягає в оцінюванні умовних “штрафів” (або витрат часу, засобів, грошей та ін.) і вимірюється зменшенням витрат у результаті досягнення цілей. Так,

наприклад, кількісною мірою цінності інформації, яка стосується поїздки у місто (розклад руху транспорту, вартість квитків тощо), може слугувати зекономлений час і (або) гроші.

Якщо, навпаки, досягнення мети малоімовірне, то зручніше користуватися критерієм, який запропонували Н.М. Бонгарт і А.А. Харкевич. Мірою цінності при цьому є логарифм відношення ймовірностей досягнення мети до одержання інформації (P_{in}) і після цього (P_{fin}):

$$I_{ц} = \log \frac{P_{fin}}{P_{in}}. \quad (3)$$

Що стосується оцінювання новизни знань, то у роботі [3] нами був запропонований підхід, що ґрунтується на використанні лексичного словника WordNet [4]. Однак існує проблема розроблення єдиного уніфікованого підходу до такого оцінювання.

Формування цілей

Здійснити формальну постановку задачі оцінювання ефективності нових знань.

Основний матеріал

Проблема автоматизованого формування онтологій БЗ супроводжується виникненням таких проблем:

- 1) зменшення якості нарощуваного обсягу інформації (релевантності до заданої ПО);
- 2) втратою монотонності, тобто появою внутрішніх конфліктів, які порушують її цілісність;
- 3) появою неконтрольованої надлишковості.

Як відзначено у [5], онтологію формують вручну, напівавтоматизовано та автоматизовано. Якщо цей процес здійснюється автоматизовано, то необхідно оптимізувати онтологію [6].

Напрями оптимізації онтології безпосередньо залежать від критерію якості ІнТС, оскільки якість таких систем залежить від БЗ [7], а в розглянутому у цій роботі підході ядром БЗ є онтологія. Для оцінювання якості ІнТС використаємо стандарт ISO 9126 [8] для інформаційних систем, оскільки спеціалізованого стандарту лише для інтелектуальних систем немає. Цей стандарт призначено або для стандартизації виробничого процесу (аналогія з життєвим циклом інформаційних систем), або для оцінювання якості програмних засобів.

Характеристики ISO 9126 визначають напрями оптимізації онтологій, які полягають в оптимізації її структури, що динамічно формується під час експлуатації системи та оптимізації змісту онтології БЗ. Під змістом розуміємо інформаційне наповнення, яке має бути гнучким, тобто налаштовуваним під конкретну ПО і потреби користувача. Іншим підходом до оптимізації онтології є оцінювання новизни знань, яку ми плануємо додавати в онтологію порівняно з тими знаннями, які вже зберігаються в ній. Саме формальна постановка цієї задачі розглядається у цій роботі.

Базовими характеристиками якості ІнТС за ISO 9126 є:

- функціональна придатність до використання;
- коректність або достовірність;
- ресурсна економічність;
- практичність;
- супроводжуваність;
- мобільність.

Оскільки ми розглядаємо клас інтелектуальних систем, призначених для підтримки прийняття рішень, ядром БЗ яких є онтології, то ці характеристики напряду залежать від якості онтологій, оскільки, своєю чергою, якість ІнТС залежить від її БЗ.

Функціональна придатність ІнТС залежить від повноти онтологій, наскільки вона точно описує специфіку ПО та задач, які у ній виникають. Своєю чергою, повнота онтологій залежить від вміння давати правильні відповіді на запити до неї. Це залежить від вміння системою оцінювати новизну знань, яку пропонується додавати до онтології. Тому надалі розглянемо метод оцінювання новизни знань, який ґрунтується на використанні лексичного словника WordNet. Мірою якості

функціональної придатності буде середній відсоток нетривіальних (ненульових), правильних відповідей на запити до онтології. Тобто

$$\chi_1 = \frac{N_q^p}{N_q} \cdot 100 \% ,$$

де N_q – кількість всіх запитів до онтології БЗ; N_q^p – кількість правильних відповідей на запити.

Визначення функціональної придатності є однією з базових характеристик ІнтС.

Коректність, або достовірність функціонування ІнтС – це відсоток достовірно розв’язаних інтелектуальною системою задач. Це основна характеристика якості ІнтС, яка залежить не лише від якості БЗ, але й від моделі функціонування таких систем, тобто від побудованої метрики. Отже

$$\chi_2 = \frac{N_z^p}{N_z} \cdot 100\% ,$$

де N_z – кількість задач, які розв’язала ІнтС; N_z^p – кількість правильно розв’язаних нею задач.

Використовуваність ресурсів (або *ресурсна економічність*) у стандартах відображається зайнятістю ресурсів центрального процесора, оперативної, зовнішньої та віртуальної пам’яті, каналів введення–виведення, терміналів і каналів зв’язку. Для покращення цієї характеристики розглянемо оптимізаційну задачу, критерієм якої буде фізичний обсяг пам’яті, яку займає онтологія. З іншого боку, очевидно, що онтологія займає найменший об’єм пам’яті, якщо в ній немає жодного поняття. Тому цей критерій необхідно скомбінувати з іншим критерієм, а саме з функціональною придатністю ІнтС. Отримуємо задачу оптимальної кількості понять онтології.

Практичність – важко формалізоване поняття, яке визначає функціональну придатність і корисність застосування ІнтС для певних користувачів. До цієї групи показників входять субхарактеристики, які з різних сторін відображають функціональну зрозумілість, зручність освоєння, системну ефективність і простоту використання ІнтС. Така придатність повинна ґрунтуватися на цілісності онтології, тобто відсутності в її тілі взаємозаперечувальних тверджень та дублювання, а також на збалансованості ПО, яка виражається у рівномірному поданні її окремих підрозділів в онтології.

Супроводжуваність ІнтС відображається зручністю і ефективною виправлення, удосконалення або адаптації структури та змісту онтології БЗ залежно від змін у зовнішньому середовищі застосування, а також у вимогах і функціональних специфікаціях замовника. Узагальнено якість супроводжуваності ІнтС можна оцінювати як потребу ресурсів для забезпечення її функціональності та реалізації. Сукупність субхарактеристик супроводжуваності програмної системи, подана в стандарті ISO 9126, цілком застосовна для описання цієї якості інтелектуальних систем, переважно тими самими організаційно-технологічними субхарактеристиками.

Мобільність характеризується тривалістю і трудомісткістю інсталяції інформаційних продуктів, адаптації та заміщення при перенесенні на інші апаратні та операційні платформи. Інформація про процеси, що відбуваються у зовнішньому середовищі, може мати великі обсяги і трудомісткість первинного накопичення та актуалізації, що визначає необхідність її ретельного зберігання та регламентованої зміни. Критерієм мобільності є швидкодія, яка виражається часом відгуку ІнтС на зовнішнє звернення (час реакції на зміну параметрів зовнішнього середовища, до яких чутлива система).

Отже, основною характеристикою якості будь-якої ІнтС є достовірність отриманого розв’язку цією системою χ_2 . Якщо ІнтС побудована на основі онтології, то вагомою є й перша характеристика χ_1 .

Для підвищення ефективності вищенаведених шести характеристик необхідно розв’язати задачу оцінювання новизни онтологічних знань.

Нехай Z – задача, яка розв’язується за допомогою знань поданих у вигляді онтології.

$Z_j \subset Z$ – деякий підклас задачі Z .

Для розв’язування задачі Z використовується множина методів $M^Z = \{M_1^Z, M_2^Z, \dots, M_n^Z\}$.

Якість методів, які використовуються для розв’язування задачі Z , оцінюється множиною параметрів $P^Z = \{p_1^Z, p_2^Z, \dots, p_m^Z\}$.

Новий метод розв'язування задачі $Z : M_{new}^Z$.

Виграш цього методу порівняно з деяким іншим i -м методом, який використовується для розв'язування підкласу Z_j задачі $Z : M_i^{Z_j}$, задається як відображення в підмножину елементів, які являють собою пару (параметр, перевага):

$$u(M_{new}^{Z_j}, M_i^{Z_j}) \rightarrow U_{i,new}^{Z_j} = \left\{ (p_i^{Z_j}, \alpha_i) \right\}_{i=1}^{k_i}, \alpha - \text{кількісний або якісний опис переваги.}$$

$$\text{Аналогічно програш } v(M_{new}^{Z_j}, M_i^{Z_j}) \rightarrow V_{i,new}^{Z_j} = \left\{ (p_i^{Z_j}, \beta_i) \right\}_{s=1}^{k_i}.$$

Функція виграшу нового методу порівняно з методом $M_i^{Z_j} : \varphi(U_{i,new}^{Z_j})$, функція програшу нового методу порівняно з методом $M_i^{Z_j} : \psi(V_{i,new}^{Z_j})$.

Читається як: Розроблено новий метод M_{new}^Z для розв'язування задачі Z , який на відміну від іншого методу M_i^Z дає змогу отримати виграш $\varphi(U_{i,new}^{Z_j})$ та програш $\psi(V_{i,new}^{Z_j})$.

Приклад. Нехай задача Z полягає у розв'язанні квадратного рівняння $ax^2 + bx + c = 0$.

Z^1 – підклас задачі, для якої $D = b^2 - 4ac > 0$, $a, b, c \in Z$, $a = 1$, $b \neq 0$, $c \neq 0$.

Z^2 – підклас задачі, для якої $D = b^2 - 4ac > 0$, $a, b, c \in Z$, $a \neq \{0, 1\}$, $b \neq 0$, $c \neq 0$.

$$M^Z = \{M_1^Z = \text{'дискримінант'}, \dots\}. P^Z = \{p_1^Z = \text{'час розв'язування'}, \dots\}.$$

$M_{new}^Z = \text{'теорема Вієта'}$. Тоді

$$u(M_{new}^{Z_1}, M_1^{Z_1}) \rightarrow U_{1,new}^{Z_1} = \{(p_1^{Z_1}, 30c)\}. v(M_{new}^{Z_2}, M_1^{Z_2}) \rightarrow V_{1,new}^{Z_2} = \{(p_1^{Z_2}, 40c)\}.$$

Тобто $\alpha_1 = 30c$, $\beta_1 = 40c$.

Функція виграшу: $\varphi = \alpha_1 = t_1 - t_{new}$ – різниця між часом розв'язування задач старим та новим методом.

Висновки

Здійснено формальну математичну постановку задачі оцінювання новизни онтологічних знань з погляду підвищення ефективності функціонування інтелектуальних систем, ядром баз знань яких є онтологія. Визначено основні характеристики інтелектуальних систем згідно із ISO 9126, що дало змогу обґрунтувати доцільність розв'язування такої задачі. Оцінювання новизни знань, на відміну від оцінювання новизни інформації, ґрунтується на функції виграшу, яку може отримати користувач системи, використовуючи ці знання. Наведено приклад, який ілюструє розроблену математичну модель задачі.

1. Литвин В.В. *Бази знань інтелектуальних систем підтримки прийняття рішень* / В.В. Литвин. – Львів: Видавництво Львівської політехніки, 2011. – 240 с.
2. Шеннон К. *Работы по теории информации и кибернетике* / К. Шеннон. – М.: 1963. – 830 с.
3. Литвин В.В. *Оцінка новизни знань під час автоматичної розбудови онтологій* / В.В. Литвин, А.С. Мельник, В.Я. Крайовський // *Вісн. Нац. ун-ту "Львівська політехніка". "Інформаційні системи та мережі"*. – 2011. – № 699. – С. 343–353.
4. Miller G.A. *WORDNET: A lexical database for English* / G.A. Miller // *Communications of ACM* (11). – 1995. – Р. 39–41.
5. *Інтелектуальні системи, базовані на онтологіях* // Д.Г. Досин, В.В. Литвин, Ю.В. Нікольський, В.В. Пасічник. – Львів: "Цивілізація", 2009. – 414 с.
6. Литвин В.В. *Задачі оптимізації структури та змісту онтології та методи їх розв'язування* / В.В. Литвин // *Вісн. Нац. ун-ту "Львівська політехніка". "Інформаційні системи та мережі"*. – 2011. – № 715. – С. 189–200.
7. Гаврилова Т.А. *Базы знаний интеллектуальных систем* / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 384 с.
8. ISO/IEC 9126:1991. *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*. – 1991. – 39 p.

В. Висоцька, Т. Шестакевич, Ю. Щербина*

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж,

*Львівський національний університет імені Івана Франка,
кафедра дискретного аналізу та інтелектуальних систем

УТВОРЕННЯ РЕЧЕНЬ АНГЛІЙСЬКОЮ ТА НІМЕЦЬКОЮ МОВАМИ ЗА ДОПОМОГОЮ ПОРОДЖУВАЛЬНИХ ГРАМАТИК

© *Висоцька В., Шестакевич Т., Щербина Ю., 2012*

Описано застосування породжувальних граматики у лінгвістичному моделюванні. Подані в роботі моделі синтаксису речення застосовують для автоматизації процесів аналізу та синтезу природномовних текстів.

Ключові слова: породжувальні граматики, структурна схема речення, комп'ютерна лінгвістична система.

This paper describes the generative grammar application in linguistic modelling. Presented in work of syntax sentence models is applied to automate the processes of analysis and synthesis of texts in natural language.

Key words: Generative grammar, structured scheme sentences, computer linguistic system.

Вступ. Загальна постановка проблеми

Швидкий темп розвитку Інтернету сприяв створенню різноманітних лінгвістичних ресурсів. Потреба в автоматизації процесів аналізу та синтезу природномовних текстів зумовила появу відповідних лінгвістичних моделей процесів їх опрацювання. Сьогодні важливо забезпечити автоматичне опрацювання природномовної інформації в інформаційних засобах як системах машинного перекладу, інформаційно-пошукових, семантичного, статистичного, оптичного та акустичного аналізу і синтезу мови, автоматизованого редагування, екстракції знань з природномовних текстів, реферування, анотування, індексування текстів, навчально-дидактичних, менеджменту лінгвістичних корпусів, інструментальних засобів укладання словників різних типів тощо. Фахівці інтенсивно шукають нових моделей та способів їх опису, методів автоматичного опрацювання природномовної інформації. Одним із таких способів є розроблення загальних принципів побудови лексикографічних систем синтаксичного типу та побудови за цими принципами зазначених систем для конкретних мов. У зв'язку з цим в умовах постійного зростання інформаційних потоків тема статті є актуальною. Необхідним є розвиток багатьох мовознавчих дисциплін для потреб інформаційних наук. Інтеграційні процеси в більшості галузей життя сучасного світу привертають особливу увагу до розроблення та створення автоматизованих систем опрацювання багатомовної інформації.

Зв'язок висвітленої проблеми із важливими науковими та практичними завданнями

Розглянувши відомі способи та підходи до вирішення проблеми автоматичного опрацювання природномовної інформації [2–13, 15, 18–21, 23–24] та виділивши недоліки й переваги існуючих підходів та результатів [6, 24], отримані у галузі синтаксичних аспектів комп'ютерної лінгвістики, сформулюємо загальні концептуальні принципи моделювання словозмінних процесів при утворенні текстових масивів на прикладі англійських та німецьких речень. Далі з використанням запропонованих синтаксичних моделей та словозмінних класифікацій лексичного складу англійських та німецьких речень розробимо лексикографічні правила синтаксичного типу для автоматизованого опрацювання цих речень. Застосовуючи запропоновану методику, можна досягти вищих показників

надійності порівняно з відомими аналогами та високої ефективності у прикладних застосуваннях при побудові нових інформаційних технологій лексикографування та дослідження словозмінних ефектів природних мов. Робота має практичну цінність, оскільки запропоновані моделі та правила дають змогу ефективно організувати процес створення лексикографічних систем синтаксичного типу.

Аналіз останніх досліджень та публікацій

Традиційно аналіз природномовних текстів складається з трьох послідовних процесів: морфологічного, синтаксичного та семантичного аналізу. Для кожного з цих етапів було створено відповідні моделі та алгоритми [6, 24]. Так, теорія породжувальних граматики, початок якої заклав американський лінгвіст Н. Хомські [7–10, 18–21], є ефективним інструментом лінгвістичного моделювання на синтаксичному рівні мови. Він використав прийом формального аналізу граматичної структури фраз, який дає змогу виділити синтаксичні структури (складові), що є основною схемою фрази незалежно від її значення. Ідеї Н. Хомські розвинув радянський лінгвіст А.В. Гладкий, який застосовував поняття дерев залежності та систем складових для моделювання синтаксичного рівня мови [2, 8–10, 13, 23]. Він запропонував спосіб моделювання синтаксису за допомогою синтаксичних груп, що виділяють складові словосполучень як одиниці побудови дерева залежностей, – таке подання дало змогу об'єднати переваги методу безпосередніх складових і дерев залежностей [4, 5, 23]. Напрацювання Н. Хомські та А.В. Гладкого застосовано до розроблення таких засобів опрацювання природної мови, як інформаційно-пошукові системи, системи машинного перекладу, анування текстів, морфологічний, синтаксичний та семантичний аналіз текстів, навчально-дидактичні системи, лінгвістичне забезпечення спеціалізованих програмних систем тощо [12]. На сучасному етапі потреба в розробленні загальних та спеціалізованих лінгвістичних систем змушує прикладну та комп'ютерну лінгвістику виходити далеко за власні межі – в галузь інформаційних технологій [8–11, 15, 18–21]. Розроблення дієвих моделей мовлення для забезпечення комп'ютерних лінгвістичних систем дасть змогу виконати такі завдання прикладної лінгвістики, як аналіз та синтез усних і писемних текстів, описування та індексування документів, перекладання текстів, створення лексикографічних баз даних тощо. Перевагами моделювання за допомогою породжувальних граматики є можливість успішно описувати синтаксичний рівень мови (правила утворення речень зі словоформ) [8–10].

Формальна породжувальна граMATика G – це четвірка $G = (V, T, S, P)$, де V – скінченна непорожня множина, алфавіт (словник); T – її підмножина, елементи якої є термінальними (основними) символами, терміналами; S – початковий символ ($S \in V$); P – скінченна множина продукцій (правил перетворення) вигляду $\xi \rightarrow \eta$, де ξ та η – ланцюжки над V . Множину $V \setminus T$ позначають N , її елементи є нетермінальними (допоміжними) символами, не терміналами [8–10]. Граматики класифікують за типами продукцій, на які накладено певні обмеження (табл. 1) [18–21].

Таблиця 1

Класифікація граматики за типами продукцій

Тип	Опис
Довільна	Тут ξ – довільний ланцюжок, що містить хоча б один нетермінальний символ, η – довільний ланцюжок над V .
Контекстно-залежна	В множині продукцій P є продукція вигляду $\gamma\xi\delta \rightarrow \gamma\eta\delta$ $ \xi \leq \eta $ (але не у формі $\xi \rightarrow \eta$), то ξ можна замінити на η лише в оточенні ланцюжків $\gamma\dots\delta$, тобто у відповідному контексті.
Контекстно-вільна	Нетермінал A у лівій частині продукції $A \rightarrow \eta$ може бути замінений ланцюжком η у довільному оточенні щоразу, коли він зустрічається, тобто незалежно від контексту.
Регулярна	Можуть бути лише продукції $A \rightarrow aB$, $A \rightarrow a$, $S \rightarrow \lambda$, де A, B – нетермінали, a – термінал, λ – порожній ланцюжок.

Формулювання мети

Лінгвістичне забезпечення використовують в усіх без винятку комп'ютерних системах. Удосконалення спілкування типу машина–людина є важливим актуальним завданням, вирішити яке можна удосконаленням процесу синтезу текстів на синтаксичному рівні. З цією метою розглянемо процес лінгвістичного моделювання синтезу речень англійською та німецькою мовами за допомогою породжувальних граматик. Для цього здійснимо належний опис словозмінних систем: на основі синтаксичного аналізу визначимо перелік відповідних слів, а також сформуємо систему правил, за допомогою яких можна одержати будь-які правильні форми речень, не отримавши при цьому жодного неправильного. Подання синтаксичної структури в термінах безпосередніх складових широко використовується в лінгвістиці, багато досліджувалося в різних аспектах і завоювало право на існування в теоретичному та експериментальному аспектах (автоматичний переклад або реферування тощо). Тому той факт, що граматики безпосередніх складових при породженні термінальних ланцюжків, наприклад, фраз природної мови, одночасно дають структуру їх безпосередніх складових (виведення фрази фактично і є структурою безпосередніх складових), робить їх особливо привабливими з лінгвістичного погляду. Довільна граMATика (без вимоги заміни відразу лише одного символу) не володіє властивістю заміняти фрази структури їх безпосередніми складовими. Оскільки в такій граMATиці кожного разу замінюється не один символ, а ціла група символів, у виведенні неможливо однозначно вказати для кожного символу його предка, і тому виведення не може бути перетворене в структуру безпосередніх складових. У межах статті покажемо способи застосування апарату породжувальних граматик до моделювання синтаксису речень англійською та німецькою. Для цього розберемо синтаксичну структуру речень, продемонструємо особливості процесу синтезу речень зазначених мов. Розглянемо вплив норм та правил мови на хід побудови граматик [6, 24].

Аналіз отриманих наукових результатів

Тлумачитимемо термінальні символи як словоформи деякої природної мови, нетермінальні символи – синтаксичні категорії, а термінальні ланцюжки, що виводяться, – правильні речення цієї мови [6–11, 18–21, 24]. Тоді виведення речення природно інтерпретується як його синтаксична структура, подана в термінах безпосередніх складових [6, 24]. Пояснимо сказане прикладами.

Приклад 1. Розглянемо процес виведення ланцюжків слів в англійській мові, де речення мають строго визначений, особливий порядок слів. Наприклад, для простого розповідного речення англійської мови основними синтаксичними категоріями є іменна та дієслівна групи [1, 25].

Іменна група виражається *займенником* або має таку структурну схему (тут і надалі у квадратних дужках зазначено необов'язкові елементи, у фігурних дужках – елементи, які можуть повторюватись):

$$[\text{Визначник}] \{ \{ \text{Прикметникова група} \} \} \text{Іменник} [\text{Прийменникова група}]. \quad (1)$$

Визначник – це граматичний клас (не частина мови), до якого належать слова, що уточнюють іменну групу з погляду визначеності, кількості тощо. *Визначником* у англійській мові слугують артиклі, присвійні та вказівні займенники, кількісні прикметники та іменники у присвійному відмінку.

Прикметникова група має таку структурну схему:

$$\{ \{ \text{Прислівник} \} \} \text{Прикметник} [\text{Прийменникова група}]. \quad (2)$$

Прийменникова група має таку структурну схему:

$$\text{Прийменник} \text{Іменна група}. \quad (3)$$

Основними граматичними характеристиками складових іменної групи в англійській мові є рід, число, відмінок іменника, відмінок займенника, ступінь порівняння, особа [1, 25].

Дієслівна група має таку структурну схему:

$$\text{Дієслівний вираз} [\text{Прислівникова група}]. \quad (4)$$

Дієслівний вираз складається з лексичних, допоміжних та модальних дієслів (тут і надалі Д. – скорочення для дієслова):

$$[\text{Модальне Д.}] [\text{Д. допоміжне}] [\text{Д. допоміжне}] [\text{Д. допоміжне}] \text{Д. лексичне} . \quad (5)$$

Прислівникова група має таку структурну схему:

$$\{\text{Прислівник}\} \{[\text{Прийменникова група}]\} . \quad (6)$$

Основними граматичними характеристиками складових дієслівної групи в англійській мові є особа, число, час, вид, спосіб, стан. Граматичними категоріями іменної та дієслівної груп є особа та число, бо саме за ними в реченні відбувається узгодження.

Розглянемо породжувальну граматику для моделювання синтаксису речення англійською мовою описаної структурної схеми. Алфавітом є іменникова група, дієслівна група та їх складові (нетермінальні символи), а також лексичний запас англійської мови (відповідні термінальні символи). За вимогами та правилами англійської мови в реченні на першому місці розташовується іменна група, після неї – дієслівна (це правило є основним). Усі можливі перетворення термінальних символів у нетермінальні складають множину правил. Під час виведення отримують незліченну кількість термінальних ланцюжків (речень англійською відповідної структурної схеми), тому така граMATика буде необмеженою і через свою складність застосування не матиме. Для введення контекстно-залежної граматики змодельємо процес побудови простого розповідного речення англійською мовою. Введемо такі обмеження на структуру речення. Розглядатимемо речення з іменною групою, вираженою не займенником, визначником-артиклем, спрощеною прикметниковою групою та без прийменникової групи; дієслівна група виражена лише лексичним дієсловом (вид – простий, спосіб – вказівний, стан – активний), у прислівниковій групі опущено прийменникову групу. Тобто розглядатимемо речення спрощеної структурної схеми. Спрощена іменна група матиме таку структурну схему:

$$\text{Визначник} \{\text{Прикметник}\} \text{Іменник} , \quad (7)$$

а спрощена дієслівна група:

$$\text{Дієслівний вираз} \{\text{Прислівник}\} . \quad (8)$$

Тепер основними граматичними характеристиками іменної групи є число, відмінок іменника, ступінь порівняння, особа. Прикметники об'єднуються сполучником. Основними граматичними характеристиками дієслівної групи будуть особа, число, час. Прислівники можуть об'єднуватися сполучником, допускається моделювання прислівника прислівником. Уведемо позначення складових спрощених іменної та дієслівної груп та їх граматичних категорій (у дужках – еквівалент англійською).

Складові іменної групи (після «/» тут і надалі вказано використовувані позначення)/ \tilde{N} : визначник/ D , прикметник/ A , іменник/ N . Граматичні категорії іменної групи та її складових: число (Number)/ NR : одина (Singular)/ sg , множина (Plural)/ pl ; відмінок (Case)/ CS : загальний (Common Case)/ cc , присвійний (Possessive Case)/ pc ; ступінь порівняння (Comparison Degree)/ CD : основна форма (Positive Degree)/ pd , вищий ступінь (Comparative Degree)/ cd , найвищий ступінь (Superlative Degree)/ sd ; особа (Person)/ PR : 1-ша (First)/ 1 , 2-га (Second)/ 2 , 3-тя (Third)/ 3 . Відповідно, скороченими позначеннями іменної групи буде $\tilde{N}_{NR,PR}$, а їх складових – $N_{NR,CS}$, A_{CD} .

Зауваження. Категорія особи – це граматична категорія особового займенника, який опущено у спрощеній формі речення. Проте цю категорію необхідно визначати для іменної групи з метою узгодження вживання дієслівної групи.

Складові дієслівної групи/ \tilde{R} : дієслово/ R , прислівник/ B . Граматичні категорії складових дієслівної групи: число (Number)/ NR : одина (Singular)/ sg , множина (Plural)/ pl ; особа (Person)/ PR : 1-ша (First)/ 1 , 2-га (Second)/ 2 , 3-тя (Third)/ 3 ; час (Time)/ TM : теперішній (Present)/ pr , минулий (Past)/ ps , майбутній (Future)/ ft . Скороченими позначеннями дієслівної групи буде $\tilde{R}_{NR,PR}$, а їх складових – $R_{NR,PR,TM}$, B .

Спосіб задання контекстно-залежної граматики, що виводить речення розробленої структурної схеми англійською мовою, наведемо на прикладі речення *The Ukrainian and Polish fans cheered emotionally and very actively.* (Українські та польські вболівальники вболівали емоційно та дуже активно.) Розглянемо граматику $G_1=(V, T, S, P)$. Алфавіт (позначення синтаксичних категорій

подамо без індексів – для зручності) $V=(S, \tilde{N}, \tilde{R}, D, A, N, R, Q, B, \text{Ukrainian}, \text{Polish}, \text{fan}, \text{cheer}, \text{the}, \text{and}, \text{emotionally}, \text{actively}, \text{very})$, $T=(\text{Ukrainian}, \text{Polish}, \text{fan}, \text{cheer}, \text{the}, \text{and}, \text{emotionally}, \text{actively}, \text{very})$, S – початковий символ. Множину правил P подамо у вигляді табл. 2.

Таблиця 2

Правила формування речення англійською мовою для Прикладу 1

№	Група	Правила
I.	Структура	$S \rightarrow \# \tilde{N}_{NR,PR} \tilde{R}_{NR,PR} \#$.
II.	Іменна група	1) $\tilde{N}_{NR,PR} \rightarrow DA_{CD} N_{NR,CS}$; 2) $A_{CD} \rightarrow A_{CD} QA_{CD}$.
III.	Дієслівна група	1) $\tilde{R}_{NR,PR} \rightarrow R_{NR,PR,TM} B$; 2) $B \rightarrow BQB$; 3) $B \rightarrow BB$; 4) $BB \rightarrow B^{\text{ступінь}} B$, тобто сполучуваність прислівника залежить від контексту. Змістовний сенс такого обмеження полягає у тому, що згідно з виділеними моделями сполучуваності прислівників, прислівники міри та ступеня, в разі сполучення їх з іншим прислівником, виступають здебільшого як препозитивний елемент (<i>so well, almost far</i>) [16].
IV.	Синтаксичні категорії слівформ	1) $A_{CD} \rightarrow \text{Ukrainian}_{CD}$ (український); 2) $A_{CD} \rightarrow \text{Polish}_{CD}$ (польський); 3) $N_{NR,CS} \rightarrow \text{fan}_{NR,CS}$ (вболівальник); 4) $R_{NR,PR,TM} \rightarrow \text{cheer}_{NR,PR,TM}$ (вболівати); 5) $D \rightarrow \text{the}$ (означений артикль); 6) $Q \rightarrow \text{and}$ (і); 7) $B \rightarrow \text{emotionally}$ (емоційно); 8) $B \rightarrow \text{actively}$ (активно); 9) $B^{\text{ступінь}} \rightarrow \text{very}$ (дуже).

Графічно виведення в граматиці G_1 для генерування англійського речення – на рис. 1 (тут і надалі $\#$ – символ межі речення, який вважається термінальним).

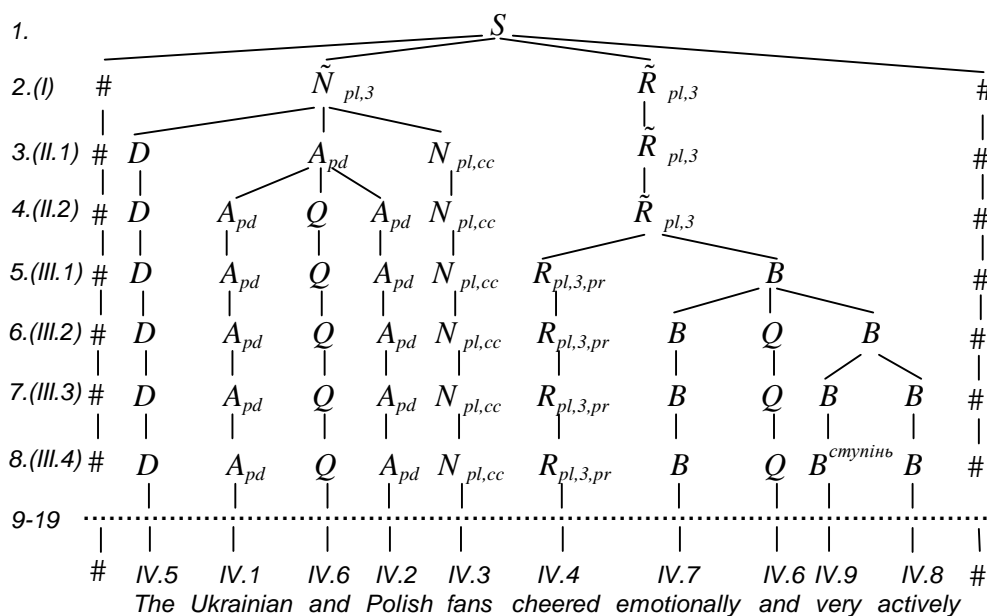


Рис. 1. Виведення в контекстно-залежній граматиці G_1 для генерування речення англійською

Виведення речення заданої структурної схеми подано нижче.

1. S
2. (I) $\# \tilde{N}_{pl,3} \tilde{R}_{pl,3} \#$
3. (II.1) $\# DA_{pd} N_{pl,cc} \tilde{R}_{pl,3} \#$

4. (II.2) # $D A_{pd} Q A_{pd} N_{pl,cc} \tilde{R}_{pl,3} \#$
5. (III.1) # $D A_{pd} Q A_{pd} N_{pl,cc} R_{pl,3,ps} B \#$
6. (III.2) # $D A_{pd} Q A_{pd} N_{pl,cc} R_{pl,3,ps} BQB \#$
7. (III.3) # $D A_{pd} Q A_{pd} N_{pl,cc} R_{pl,3,ps} BQBB \#$
8. (III.4) # $D A_{pd} Q A_{pd} N_{pl,cc} R_{pl,3,ps} BQB^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
9. (IV.5) # $The A_{pd} Q A_{pd} N_{pl,cc} R_{pl,3,ps} BQB^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
10. (IV.1) # $The Ukrainian Q A_{pd} N_{pl,cc} R_{pl,3,ps} BQB^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
11. (IV.2) # $The Ukrainian Q Polish N_{pl,cc} R_{pl,3,ps} BQB^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
12. (IV.6) # $The Ukrainian and Polish N_{pl,cc} R_{pl,3,ps} BQB^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
13. (IV.6) # $The Ukrainian and Polish N_{pl,cc} R_{pl,3,ps} B \text{ and } B^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
14. (IV.3) # $The Ukrainian and Polish fans R_{pl,3,ps} B \text{ and } B^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
15. (IV.4) # $The Ukrainian and Polish fans cheered B \text{ and } B^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
16. (IV.7) # $The Ukrainian and Polish fans cheered emotionally \text{ and } B^{cmyu\text{ni}\tilde{n}\tilde{b}} B \#$
17. (IV.8) # $The Ukrainian and Polish fans cheered emotionally \text{ and } B^{cmyu\text{ni}\tilde{n}\tilde{b}} \text{ actively} \#$
18. (IV.9) # $The Ukrainian and Polish fans cheered emotionally \text{ and very actively} \#$

Кожен крок виведення полягає або в розгортанні одного з символів попереднього ланцюжка (так, при переході від ланцюжка 2 до ланцюжка 3 символ $\tilde{N}_{pl,3}$ розгортається в три символи – $D, A_{pd}, N_{pl,cc}$), або в заміні його іншим (наприклад, при переході від ланцюжка 7 до ланцюжка 8 символ B замінюється на $B^{cmyu\text{ni}\tilde{n}\tilde{b}}$), інші ж символи переписуються без зміни. Граматика G_1 здатна породжувати й інші фрази (які не обов'язково мають бути змістовними), наприклад: # *The Ukrainian and Ukrainian and Ukrainian and Ukrainian fans cheered actively and very actively* # тощо. Більше того, граматика G_1 породжує нескінченно багато різних фраз, бо до її складу входять циклічні правила (наприклад, II.2). Особливість такого правила полягає в тому, що результат його застосування містить входження його лівої частини, так що воно завжди може бути застосоване до свого власного результату, що і приводить до нескінченної кількості фраз, і множина всіх цих фраз нескінченна.

Приклад 2. Розглянемо процес виведення речення німецькою мовою. Для простого розповідного речення німецькою мовою основними синтаксичними категоріями є іменна та дієслівна групи [14]. Як і у попередньому прикладі, алфавітом є іменникова група, дієслівна група та їхні складові (нетермінальні символи), а також лексичний запас німецької мови (відповідні термінальні символи). Усі можливі перетворення термінальних символів у нетермінальні є множиною правил. Під час виведення отримаємо незліченну кількість термінальних ланцюжків. Така граматика буде необмеженою, і через свою складність застосування не матиме. Щоб отримати контекстно-залежну граматику, введемо певні обмеження. Система порядку слів у німецькій мові не є абсолютно зв'язною, фіксоване розташування слів стосується лише позиції дієслівної групи. Розташування іменної групи та інших синтаксичних форм є відносно нефіксованим [22], а тому допускає і більше структурних видозмін схеми речення. З огляду на це, введемо структурні схеми іменної та дієслівної груп одразу з певними обмеженнями (спрощеннями). Наприклад, **іменна група** має таку структуру:

$$[\text{Визначник}][\{\text{Прикметникова група}\}][\text{Іменник}][\text{Прийменникова група}] \text{ або } \text{Займенник} . \quad (9)$$

Визначник – це граматичний клас (строго кажучи, не частина мови), до якого належать слова, що уточнюють іменну групу з погляду визначеності, кількості тощо. *Визначником* у німецькій мові можуть бути артиклі, присвійні та вказівні займенники, кількісні прикметники та іменники у присвійному відмінку. *Прикметникова група* має таку структуру:

$$\{\{\text{Прислівник}\}\}[\text{Прикметник}][\text{Прийменникова група}] . \quad (10)$$

Прийменникова група має таку структуру:

$$\text{Прийменник Іменна група.} \quad (11)$$

Іменна група володіє сукупністю граматичних ознак її елементів, тобто основними граматичними характеристиками іменної групи в німецькій мові є рід, число, відмінок іменника, відмінок займенника, ступінь порівняння, особа [14]. Узгоджуються дієслівна та іменникова групи за особою та числом. Стійкою ознакою синтаксичної будови німецького речення є його рамкова структура [17].

Дієслівна група має таку структуру:

$$\text{Дієслівний вираз [Прислівникова група].} \quad (12)$$

Розглядатимемо наступну спрощену структурну схему іменної групи:

$$\{\{\text{Прикметник}\}\}\text{Іменник}\{\{\text{Іменна група}\}\}\text{або Займенник,} \quad (13)$$

а спрощена структурна схема дієслівної групи:

$$\text{Дієслово}\{\{\text{Іменна група}\}\}. \quad (14)$$

Дієслівна група в німецькій мові є більш місткою порівняно з дієслівною групою української мови внаслідок формального введення до неї модальних членів, часток, заперечень та охоплення групою дієслова всіх частин речення, за винятком першого місця [14]. Нормативність вербальної рамки є стійкою ознакою синтаксичної структури німецького речення. Існує три види вербальних рамок: аналітичні форми дієслова (допоміжне дієслово + основне дієслово, основне дієслово + інфінітив іншого дієслова, дієслово + відокремлюваний префікс), основне дієслово + іменник, основне дієслово + прислівник [17]. Дієслівний вираз складається з лексичних, допоміжних та модальних дієслів:

$$\{\{\text{Модальне Д.}\}\}\{\{\text{Д. допоміжне}\}\}\{\{\text{Заперечення}\}\}\{\{\text{Основне Д.}\}\}\{\{\text{Інфінітив Д.}\}\}\{\{\text{Префікс}\}\}. \quad (15)$$

Прислівникова група має таку структуру:

$$\{\{\text{Прислівник}\}\}\{\{\text{Прийменникова група}\}\}. \quad (16)$$

Уведемо позначення спрощених граматичних категорій для іменної та дієслівної груп.

Складові іменної групи (після «/» тут і надалі вказано використовувани позначення)/ \tilde{N} : займенник (das Pronomen)/ N^{pron} , прикметник (Adjektiv)/A, іменник (Substantive)/N. Граматичні категорії іменної групи та її складових: число (Numerus)/NR: однина (Einzahl, Singular)/sg, множина (Plural)/pl; рід (Genus)/GN: чоловічий (Maskulinum)/mg, середній (Neutrum)/ng, жіночий (Femininum)/fg; відмінок (Kasus)/KS: називний (der Nominativ)/nk, родовий (der Genitiv)/gk, давальний (der Dativ)/dk, знахідний (der Akkusativ)/ak; особа (Person)/PR: 1-ша (erste Person)/1, 2-га (zweite Person)/2, 3-тя (dritte Person)/3. Відповідно, скороченим позначенням іменної групи буде $\tilde{N}_{GN,NR,KS,PR}$, а її складових – $A_{GN,NR,KS}$, $\tilde{N}_{GN,NR,KS,PR}$, $N_{GN,NR,KS,PR}^{pron}$, тобто

$$P = \{ \tilde{N}_{GN,NR,KS,PR} \rightarrow A_{GN,NR,KS} \tilde{N}_{GN,NR,KS,PR}, \tilde{N}_{GN,NR,KS,PR} \rightarrow N_{GN,NR,KS,PR}, \tilde{N}_{GN,NR,KS,PR} \rightarrow N_{GN,NR,KS,PR}^{pron} \}.$$

Складові дієслівної групи/ \tilde{R} : дієслово (Verben)/R, іменник (Substantive)/N. Граматичні категорії дієслівної групи: особа (Person)/PR: 1-ша (erste Person)/1, 2-га (zweite Person)/2, 3-тя (dritte Person)/3, число (Numerus)/NR: однина (Einzahl, Singular)/sg, множина (Plural)/pl; час (Verbformen)/VF: простий дійсний (Präsens)/pr, простий минулий (Präteritum)/pm, складний минулий (Perfekt)/pt, складний минулий (Plusquamperfekt)/pl, майбутній (Futur I)/ft, складний майбутній (Futur II)/fr. Скороченим позначенням дієслівної групи буде $\tilde{R}_{NR,VF,PR}$, а її складової – $R_{NR,VF,P}$, тобто

$$P = \{ \tilde{R}_{NR,VF,PR} \rightarrow \tilde{R}_{NR,VF,PR} \tilde{N}_{GN',NR',KS,PR'}, \tilde{R}_{NR,VF,PR} \rightarrow R_{NR,VF,PR}, \tilde{N}_{GN,NR,KS,PR} \rightarrow A_{GN,NR,KS} \tilde{N}_{GN,NR,KS,PR}, \tilde{N}_{GN,NR,KS,PR} \rightarrow N_{GN,NR,KS,PR}, \tilde{N}_{GN,NR,KS,PR} \rightarrow N_{GN,NR,KS,PR}^{pron} \}.$$

Спосіб задання граматики, що виводить речення розробленої структурної схеми німецькою мовою, наведемо на прикладі речення *In seinem bedeutendsten Werk, in der Gedichtsammlung "Kobsar", zeigt er, Taras Schewtschenko, die bunte Welt des ukrainischen Dorfes in ihrem einmaligen Reiz* [14]. (У своїй найважливішій роботі, в антології "Кобзар", Тарас Шевченко показує барвистий світ українського села в його неповторній привабливості). Для спрощення скоротимо фразу до вигляду: *In seinem bedeutendsten Werk zeigt er die bunte Welt des ukrainischen Dorfes in ihrem einmaligen Reiz*. (У своїй найважливішій роботі він показує барвистий світ українського села в його

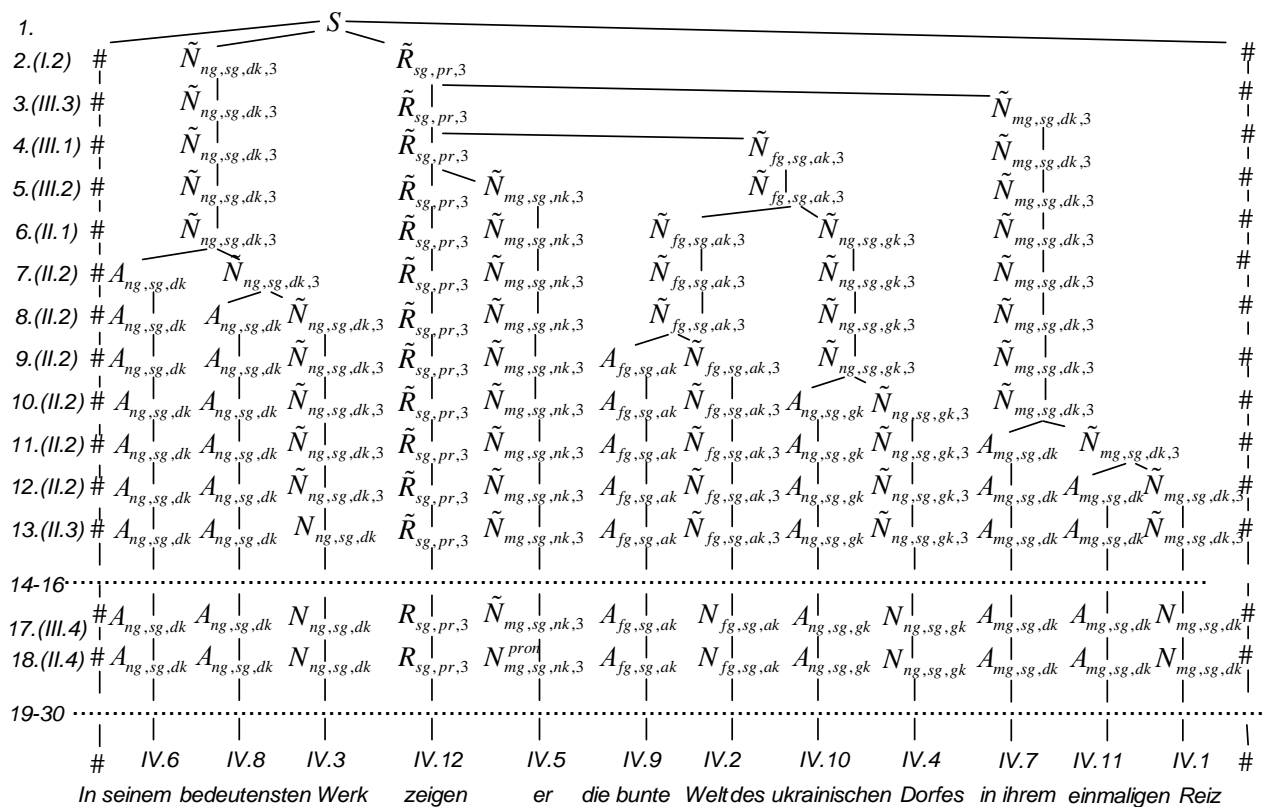


Рис. 2. Виведення в контекстно залежній граматиці G_2 для генерування речення німецькою

Кожен крок виведення полягає в розгортанні одного з символів попереднього ланцюжка (при переході від 2 до 3 ланцюжка $\tilde{R}_{sg,pr,3}$ розгортається в $\tilde{R}_{sg,pr,3}$ та $\tilde{N}_{mg,sg,dk,3}$) або в заміні його іншим (при переході від 16 до 17 ланцюжка $\tilde{R}_{sg,pr,3}$ замінюється на $R_{sg,pr,3}$), а інші символи переписуються без зміни. Граматика G_2 здатна породжувати й інші фрази, не обов'язково змістовні, наприклад: # *In seinem seinem seinem bedeutendsten bedeutendsten Werk zeigt er die bunte bunte Welt des ukrainischen Dorfes in ihrem ihrem ihrem ihrem einmaligen Reiz* #.

Висновки і перспективи подальших наукових розвідок

Дослідження застосування математичних методів для аналізу та синтезу текстів природною мовою беруть свій початок ще з середини 50-х років минулого століття, коли на розроблення математичних алгоритмів та комп'ютерних програм опрацювання природномовних текстів було спрямовано значні зусилля науковців. Апарат породжувальних граматики, запропонований Н. Хомські, моделює процеси на синтаксичному рівні мови – виділені структурні елементи речення дають змогу описувати синтаксичні конструкції незалежно від їх змісту. У статті показано особливості процесу синтезу речень різних мов із застосуванням породжувальних граматики, розглянуто вплив норм та правил мови на процес побудови граматики. Застосування породжувальних граматики має широкі можливості в розробленні та створенні автоматизованих систем опрацювання багатомовної інформації та лінгвістичного забезпечення комп'ютерних лінгвістичних систем.

1. Английская грамматика в доступном изложении // Режим доступа: <http://real-english.ru/crash/lesson3.htm>. 2. Анисимов А.В. Алгоритмична модель асоціативно-семантичного контекстного аналізу текстів природною мовою / А.В. Анисимов, О.О. Марченко, А.О. Никоненко // Пробл. програмув. – 2008. – № 2, 3. – С. 379–384. 3. Анисимов А.В. Компьютерная лингвистика для всех: мифы, алгоритмы, язык / А.В. Анисимов. – К.: Думка, 1991. – 208 с. 4. Апресян Ю.Д. Идеи и методы современной структурной лингвистики / Ю.Д. Апресян. – М.: Просвещение, 1966. – 305 с.

5. Апресян Ю.Д. Непосредственно составляющих метод / Ю.Д. Апресян // Лингвистический энциклопедический словарь под ред. В.Н. Ярцевой. – М.: Советская энциклопедия, 1990. – Режим доступа: <http://tapemark.narod.ru/les/332a.html>. 6. Висоцька В.А. Генерування речень українською за допомогою породжувальних граматики / В.А. Висоцька, Т.В. Шестакевич // ISDMIT'2012, Євпаторія. – 27–31 травня 2012. – С. 48–50. 7. Гакман О.В. Генеративно-трансформаційна лінгвістика Н. Хомського як вираження його лінгвістичної філософії / О.В. Гакман // Мультиверсум. Філософський альманах. – К.: Центр духовної культури, 2005. – № 45. – С. 98–114. 8. Гладкий А.В. Синтаксические структуры естественного языка в автоматизированных системах общения / А.В. Гладкий. – М.: Наука, 1985. – 144 с. 9. Гладкий А.В. Элементы математической лингвистики / А.В. Гладкий, И.А. Мельчук. – М.: Наука, 1969. – 192 с. 10. Гладкий А.В. Формальные грамматики и языки / А.В. Гладкий. – М.: Наука, 1973. – 368 с. 11. Дарчук Н.П. Комп'ютерна лінгвістика (автоматичне опрацювання тексту): підручник / Н.П. Дарчук. – К.: ВПЦ “Київський університет”, 2008. – 351 с. 12. Любченко Т.П. Лексикографічні системи граматичного типу та їх застосування в засобах автоматизованого опрацювання мови: автореф. дис. канд. техн. наук: спец. 10.02.21 / Т.П. Любченко. – Київ, 2011. – 19 с. 13. Марченко О.О. Алгоритми семантичного аналізу природномовних текстів: автореф. дис. ... канд. фіз.-мат. наук: спец. 01.05.01 // О.О. Марченко. – К., 2005. – 15 с. 14. Носков С.А. Самоучитель немецкого языка. Deutsch für sie / С.А. Носков. – К.: Наука, 1999. – 400 с. 15. Партико З.В. Прикладна і комп'ютерна лінгвістика. Вступ до спеціальності: навч. посіб. / З.В. Партико. – Л.: Афіша, 2008. – 224 с. 16. Торосян О.М. Функціональні характеристики прислівників міри та ступеня в сучасній англійській мові: автореф. дис. ... канд. філол. наук / О.М. Торосян. – Режим доступу: <http://disser.com.ua/contents/6712.html>. 17. Туришева О.О. Порушення рамкової конструкції в сучасній німецькій мові: функціональний аспект, нормативний статус: автореф. дис. ... канд. філол. наук: спец. 10.02.04 / Туришева О.О. – Одеса, 2012. – 20 с. 18. Хомський Н. О некоторых формальных свойствах грамматик / Н. Хомский // Кибернетический сборник. – М.: Мир, 1962. – № 5. – С. 279–311. 19. Хомський Н. Формальный анализ естественных языков / Н. Хомский, Дж. Миллер // Кибернетический сборник. – М.: Мир, 1965. – № 1. – С. 231–290. 20. Хомський Н. Язык и мышление / Н. Хомский // Публикации ОСиПЛ. Серия монографий. – М.: Издательство Московского университета, 1972. – № 2. – 122 с. 21. Хомський Н. Синтаксические структуры / Н. Хомский // Сборник «Новое в лингвистике». – М.: ИЛ, 1962. – № 2. – С. 412–527. 22. Чепурна З.В. Трансформація порядку слів у простому реченні при перекладі з німецької мови українською / З.В. Чепурна // Наукові записки, серія «Філологічні науки (мовознавство)»: у 5 ч. – Кіровоград: РВВ КДПУ ім. В. Винниченка, 2010. – Вип. 89 (1). – С. 232–236. 23. Шаров С.А. Средства компьютерного представления лингвистической информации / С.А. Шаров. – Режим доступу : <http://www.ksu.ru/eng/science/ittc/vol00>. 24. Шестакевич Т.В. Застосування породжувальних граматики для генерування речень українською мовою / Т.В. Шестакевич, В.А. Висоцька // Східно-Європейський журнал передових технологій. – Харків, 2012. – № 3/2 (57). – С. 51–53. 25. English Verbs (Part 1) – Basic Terms. – Режим доступу: <http://sites.google.com/site/englishgrammarguide/Home/english-verbs--part-1---basic-terms>.

Н. Яворський, І. Фармага, У. Марікуца
Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування.

РОЗРОБЛЕННЯ ДИСКРЕТНОЇ МОДЕЛІ ЗНАХОДЖЕННЯ ЕФЕКТИВНИХ ТЕПЛОФІЗИЧНИХ ХАРАКТЕРИСТИК КОМПОЗИТНИХ МАТЕРІАЛІВ ЗІ СКЛАДНОЮ СТРУКТУРОЮ

© Яворський Н., Фармага І., Марікуца У., 2012

Описано спосіб розрахунку теплофізичних характеристик композитних матеріалів зі складною внутрішньою структурою на основі дискретної моделі, що будується за методом теплоелектричних аналогій.

Ключові слова: МСК, задача теплопровідності, композит, дискретна модель, теплоелектричні аналогії, побудова сітки, триангуляція Делоне.

This paper consider the method of calculation the thermophysical characteristics of composite materials with complex internal structure based on the discrete model that is constructed by the method of thermal-electric analogies.

Key words: FEM, thermal analysis, composite, discrete model, thermal-electric analogies, mesh constructing, Delaunay triangulation.

Вступ

Вимогами до сучасних матеріалів є низька вартість і водночас унікальні фізико-хімічні властивості, тому зараз активно використовуються композитні матеріали. Теплофізичні характеристики композитів отримують експериментальним шляхом. Однак, важливе місце тут займають і модельні експерименти з використанням аналітичних, чисельних та чисельно-аналітичних методів [1].

Задачу розрахунку теплофізичних характеристик композитних матеріалів можна розглядати як обернену задачу теплопровідності, що є складовою частиною процесу теплового проектування технічних об'єктів, зокрема радіоелектронної апаратури [2].

Для розрахунку коефіцієнтів теплопровідності більшості композитних матеріалів різної структури виведено ряд емпіричних та напівемпіричних формул [3]. З іншого боку, ефективний коефіцієнт теплопровідності для композитів з регулярною структурою можна визначити на основі аналізу перенесення тепла в елементарній комірці. Для отримання простих наближених залежностей для $\lambda_{\text{еф}}$ часто проводять штучне дроблення комірки, в результаті чого вона набуває вигляду сукупності ділянок з паралельним і послідовним з'єднанням теплових опорів [1], [4]. Так можна уникнути необхідності проведення реальних експериментів та емпіричних розрахунків, а обмежитись лише процесами моделювання за допомогою ЕОМ.

Для синтезу ефективних теплофізичних параметрів композитних матеріалів складної структури ми пропонуємо використати аналогічну дискретну модель.

Постановка задачі, аналіз способів рішення

Стационарну задачу теплопровідності (закон Фур'є), за умови відсутності внутрішніх джерел тепла можна записати як (1).

$$-q = \lambda \nabla T = \lambda_x \frac{\partial^2 T}{\partial x^2} + \lambda_y \frac{\partial^2 T}{\partial y^2} + \lambda_z \frac{\partial^2 T}{\partial z^2}, \quad (1)$$

де q – густина теплового потоку; λ – коефіцієнт теплопровідності середовища; ∇T – градієнт утвореного температурного поля [5].

Крім самого рівняння, для коректності постановки задачі теплопровідності необхідно вказати крайові умови (постановка крайової задачі).

Ефективний коефіцієнт теплопровідності λ_{eff} , який відображає здатність матеріалу проводити тепло, можна визначити як (2).

$$\lambda_{eff} = \frac{d_m}{R_T}, \quad (2)$$

$$R_T = \frac{\Delta T}{q}, \quad (3)$$

де d_m – товщина матеріалу; R_T – термічний опір; ΔT – стала різниця температур між сторонами тіла [6].

Метод скінченних елементів дає змогу моделювати складні задачі математичної фізики, ідея методу полягає в побудові дискретної моделі що апроксимує складну невідому функцію за допомогою скінченної множини простіших [6].

Математично метод скінченних елементів полягає в наближеному розв'язанні відповідно варіаційної задачі [7]. З погляду варіаційного аналізу складну диференціальну функцію (1), що описує фізичний процес, що проходить у деякій області, можна замінити певним функціоналом (4).

$$I[T] = \iiint_V \frac{1}{2} \left[\lambda_x \left(\frac{\partial T}{\partial x} \right)^2 + \lambda_y \left(\frac{\partial T}{\partial y} \right)^2 + \lambda_z \left(\frac{\partial T}{\partial z} \right)^2 \right] dV + \iint_S q_s T dS. \quad (4)$$

Розбиваючи область на множину скінченних підобластей, функціонал подається сукупністю простих, переважно інтегральних функцій (5), кожна з яких визначено тільки в одній обмеженій підобласті, тобто функціонал, це так би мовити, функція від функції області.

$$I[T] = \sum_n I^{(n)}[T]. \quad (5)$$

Функцію області в загальному випадку можна подати в матричному вигляді (6)

$$\varphi^{(n)} = [N][\Phi], \quad (6)$$

де $[N]$ – це коефіцієнти впливу кожного з визначених вузлів підобласті, $[\Phi]$ – це шукані вузлові значення.

Варіаційна задача зводиться до мінімізації внутрішньої енергії функціонала (7)

$$\frac{\partial I[T]}{\partial [\Phi]} = \sum_n \frac{\partial I^{(n)}[T]}{\partial [\Phi]_{(n)}} = 0. \quad (7)$$

Для розв'язання такої задачі потрібно подати інтеграл (4) через вузлові значення функцій форми (6), для чого вводиться вектор часткових похідних (8)

$$[g^{(n)}] = \begin{Bmatrix} \frac{\partial \varphi^{(n)}}{\partial x} \\ \frac{\partial \varphi^{(n)}}{\partial y} \\ \frac{\partial \varphi^{(n)}}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1^{(n)}}{\partial x} & \frac{\partial N_2^{(n)}}{\partial x} & \dots & \frac{\partial N_p^{(n)}}{\partial x} \\ \frac{\partial N_1^{(n)}}{\partial y} & \frac{\partial N_2^{(n)}}{\partial y} & \dots & \frac{\partial N_p^{(n)}}{\partial y} \\ \frac{\partial N_1^{(n)}}{\partial z} & \frac{\partial N_2^{(n)}}{\partial z} & \dots & \frac{\partial N_p^{(n)}}{\partial z} \end{bmatrix} \begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_p \end{Bmatrix} \quad (8)$$

або

$$[g^{(n)}] = [B^{(n)}][\Phi]. \quad (9)$$

Для зручності також введемо матрицю коефіцієнтів теплопровідності (10)

$$[D^{(n)}] = \begin{bmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & \lambda_z \end{bmatrix}. \quad (10)$$

Використовуючи (5–10), задачу мінімізації функціоналу (4) можна записати у вигляді системи лінійних алгебраїчних рівнянь (11)

$$[B^T][D][B][\Phi] = [F] \quad (11)$$

або

$$[K][\Phi] = [F], \quad (12)$$

де $[K]$ – це коефіцієнти впливу кожного з визначених вузлів підобласті, разом із внеском граничних умов; $[\Phi]$ – це шукані вузлові значення температур; $[F]$ – це коефіцієнти навантажень, що також зумовлені наявністю граничних умов.

Отже, задача математичної фізики (1) зводиться до розв'язання системи лінійних алгебраїчних рівнянь (12), що є тривіальною задачею, яка легко обчислюється за допомогою комп'ютера.

Іншим, не менш ефективним способом розв'язання задачі теплопровідності, є метод електричних аналогій [4], [5]. Електрична аналогія оснований на формальній подібності диференціальних рівнянь теплопровідності, з одного боку, та рівнянь електропровідності – з іншого.

За законом Ома, густина потоку електричного струму пов'язана з напруженістю електричного поля, що, своєю чергою, пов'язана з потенціалом (13)

$$-J = -\frac{I}{A} = \sigma \nabla V = \sigma_x \frac{\partial^2 V}{\partial x^2} + \sigma_y \frac{\partial^2 V}{\partial y^2} + \sigma_z \frac{\partial^2 V}{\partial z^2}, \quad (13)$$

де I – це сила струму; A – площа поверхні, перпендикулярно якій протікає струм; σ – це коефіцієнт електропровідності; ∇V – це градієнт утвореного електричного поля [8]. Очевидно, що рівняння (1) та (13) є подібними.

Завдяки такому підходу задачу теплопровідності можна розв'язувати на основі аналізу теплового кола, що будується за допомогою аналогій відповідних теплових і електричних елементів (табл. 1) [4].

Таблиця 1

Елементи аналогій теплового та електричного кіл

Елемент теплової схеми	Електрична аналогія	Графічне позначення
Ізотермічна поверхня чи об'єм	Провідник	
Ідеальний тепловий зв'язок	Резистор	
Зосереджений тепловий опір (провідність)	Земля	
Джерело температурного напору (конвекція)	Джерело напруги	
Джерело теплового потоку (заданий тепловий потік)	Джерело струму	
Теплова ємність	Конденсатор	

Синтез дискретної моделі

Розглядаючи стаціонарну задачу теплопровідності для синтезу ефективного коефіцієнта теплопровідності, нас переважно цікавить тепловий опір. Згідно з [1, 4] тепловий опір вводиться за допомогою розбиття вже відомого температурного поля на сукупність елементарних комірок за допомогою ізотермічних та адіабатичних поверхонь, як це показано на рис. 1.

Такий підхід є актуальним, якщо можна передбачити температурне поле, наприклад, коли структура композиту є простою, багатошаровою з однорідними шарами. Якщо структура є складною, наприклад, із вкрапленнями різної форми та розмірів, то виникають проблеми у побудові теплового кола, оскільки температурне поле буде неоднорідним. У такому випадку в [1] пропонується вдаватися до спрощень структури та усереднень під час обчислень.

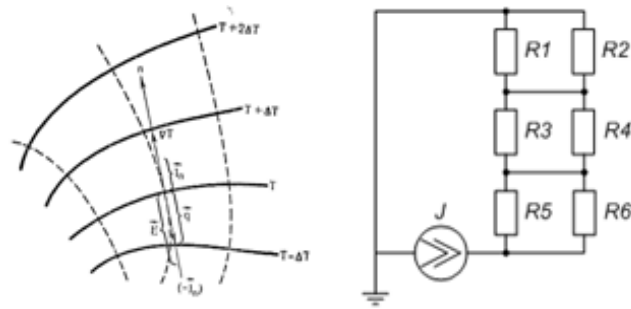


Рис. 1. Температурне поле, розбите на елементарні комірки, та відповідне теплове коло

Для синтезу ефективних теплофізичних параметрів композитних матеріалів складної структури ми пропонуємо використовувати дискретну модель у вигляді розбиття композита на скінченні симплекс-елементи, після чого вже на основі розбиття будувати теплове коло та його аналізувати.

На основі методу електричних аналогій кожен симплекс-елемент можна подати у вигляді набору опорів [9], наприклад, розглянемо двовимірний симплекс-елемент (трикутник) (рис. 2).

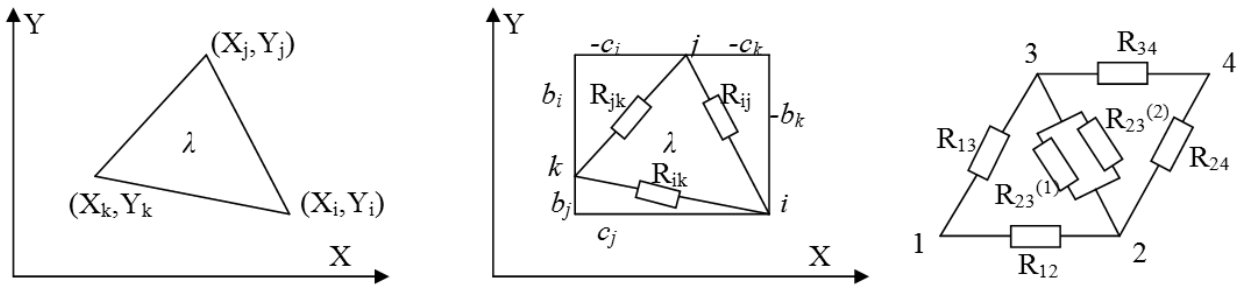


Рис. 2. Схема теплового кола на основі трикутного симплекс-елемента та приклад з'єднання двох трикутних елементів

Значення опорів на рис. 2 можна подати як:

$$\frac{1}{R_{ij}} = Y_{ij} = \frac{1}{2} \lambda \frac{b_i b_j + c_i c_j}{b_i c_j - c_i b_j}; \quad (14)$$

$$\frac{1}{R_{ik}} = Y_{ik} = \frac{1}{2} \lambda \frac{b_i b_k + c_i c_k}{b_i c_k - c_i b_k}; \quad (15)$$

$$\frac{1}{R_{jk}} = Y_{jk} = \frac{1}{2} \lambda \frac{b_j b_k + c_j c_k}{b_j c_k - c_j b_k}. \quad (16)$$

З іншого боку, функція форми для трикутного елемента має вигляд, як показано на рис. 3.

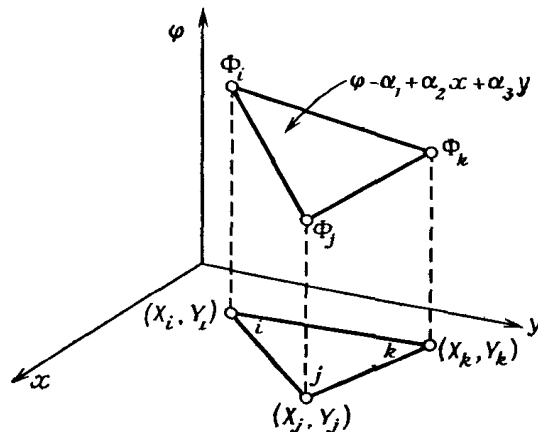


Рис. 3. Двовимірний симплекс-елемент та його функція форми

Або, аналогічно до (6), до функції форми можна записати в матричному вигляді (17).

$$\varphi = N_i \Phi_i + N_j \Phi_j + N_k \Phi_k = [N][\Phi] \quad (17)$$

де

$$N_i = \frac{1}{2A} [a_i + b_i x + c_i y] \text{ та } \begin{cases} a_i = X_j Y_k - X_k Y_j \\ b_i = Y_j - Y_k \\ c_i = X_k - X_j \end{cases} \quad (18)$$

$$N_j = \frac{1}{2A} [a_j + b_j x + c_j y] \text{ та } \begin{cases} a_j = X_k Y_i - X_i Y_k \\ b_j = Y_k - Y_i \\ c_j = X_i - X_k \end{cases} \quad (19)$$

$$N_k = \frac{1}{2A} [a_k + b_k x + c_k y] \text{ та } \begin{cases} a_k = X_i Y_j - X_j Y_i \\ b_k = Y_i - Y_j \\ c_k = X_j - X_i \end{cases} \quad (20)$$

$$2A = \begin{vmatrix} 1 & X_i & Y_i \\ 1 & X_j & Y_j \\ 1 & X_k & Y_k \end{vmatrix} \quad (21)$$

Тобто побудоване теплове коло симплекс-елемента відповідає його функції форми.

Для аналізу побудованого теплового кола розглянемо метод вузлових потенціалів [10]. Метод вузлових потенціалів – це метод розрахунку електричних кіл шляхом запису системи лінійних алгебраїчних рівнянь, в якій невідомими є потенціали у вузлах кола. У результаті застосування методу визначаються потенціали у всіх вузлах ланцюга, а також, за необхідності, струми у всіх контурах.

У матричному вигляді система рівнянь для методу вузлових потенціалів виглядає так (22):

$$[A][Y][A^T][U] = -A(J + YE), \quad (22)$$

де $[A]$ – це матриця з'єднань; $[Y]$ – діагональна матриця провідностей; $[U]$ – шукані вузлові потенціали; $[J]$ – джерела струму; $[E]$ – джерела напруги.

Враховуючи електричні аналогії (табл. 1) та те, що теплове коло, складене з опорів (14–16), відповідає функції форми симплекс-елемента (17–21), рівняння (22) можна записати у вигляді (23):

$$[K][U] = [F], \quad (23)$$

де $[K]$ – це коефіцієнти впливу кожного з визначених вузлів; $[F]$ – це коефіцієнти навантажень, що зумовлені наявністю джерел струму чи напруги (наявністю граничних умов).

Тобто система рівнянь (12) ідентична системі рівнянь (23).

Отже, будь-яку складну структуру композитних матеріалів, де розглядається процес теплопровідності, можна подати у вигляді сітки термічних опорів.

Аналіз методів побудови скінченно-елементних сіток

Слід зауважити, що скінченний елемент дає найкращу апроксимацію шуканої величини, коли значення у його вузлах не відрізняються [7]. Для уникнення зайвих обчислень область потрібно розбивати на скінченні елементи так, щоб їх кількість збільшувалася тільки з наближенням до зон, де шукані вузлові значення починають сильно відрізнятися в межах одного елемента. Такими зонами є границі переходів від одного матеріалу в інший.

Іншим критерієм, що впливає на точність апроксимації шуканої неперервної величини, є форма скінченного елемента. Для уникнення похибок апроксимації скінченний елемент має бути рівностороннім.

Існують два основні класи методів побудови сіток (триангуляції) – прямі та ітераційні [11]. За прямими методами сітка будується за один етап, причому її топологія (інакше кажучи, граф зв'язків між вузлами) і координати всіх вузлів відомі спочатку. За ітераційними методами сітка будується послідовно, на кожному кроці додається один або кілька елементів, причому спочатку не відомі ні координати вузлів, ні топологія сітки. Крім того, координати вузлів і топологія можуть змінюватися безпосередньо в процесі побудови.

Ітераційні методи на основі критерію Делоне є, мабуть, найпоширенішими методами триангуляції для методу скінченних елементів, які дозволяють повністю автоматизувати процес побудови скінченно-елементної сітки, що відповідає заданим критеріям.

Кажуть, що триангуляція задовольняє умову Делоне, якщо всередину кола, описаного навколо будь-якого побудованого трикутника, не потрапляє жодна із заданих точок триангуляції [12]. Триангуляція Делоне має низку цікавих властивостей, зокрема відомо, що триангуляція Делоне на заданій множині точок має максимальну суму мінімальних кутів елементів, тобто є оптимальною.

Проте, алгоритми побудови триангуляції на основі критерію Делоне не можна використовувати для методу скінченних елементів, без певної модифікації. Такою модифікацією є уточнення Раперта [13], що дає змогу створити такий набір точок, на якому мінімальні кути триангуляції не були б меншими за заданий поріг, тобто наблизити форми елементів до рівносторонніх (рис. 4).

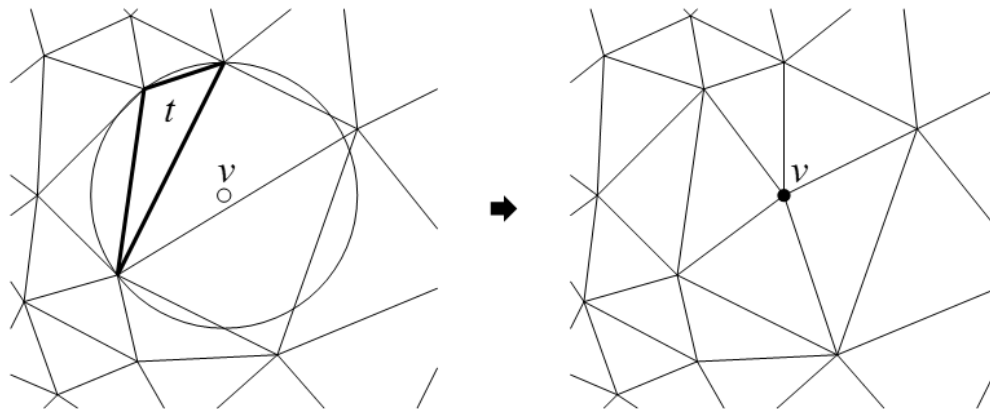


Рис. 4. Приклад роботи алгоритму Раперта:
t – нерівносторонній трикутник; *v* – центр описаного навколо *t* кола

Тому для побудови скінченно-елементної сітки ми використовуємо ітераційний алгоритм прямої побудови триангуляції Делоне (метод прогресивного фронту) з уточненням Раперта [14]. Подібні алгоритми можна застосовувати для задач з будь-якою розмірністю.

Висновки

Зусилля багатьох фізиків, хіміків та технологів напрямлені на розроблення матеріалів з наперед заданими властивостями. Вважають, що достатньо достовірні дані про фізичні властивості переважної більшості речовин може дати тільки досвід, а про властивості нових матеріалів слід судити тільки після того, як їх буде створено та досліджено [1].

Завдяки методу аналогій задачу теплопровідності можна розв'язувати на основі аналізу теплового кола, що будується за допомогою аналогій між тепловими і електричними елементами.

Запропонована дискретна модель дає змогу описувати будь-яку складну внутрішню структуру композитних матеріалів, де розглядається процес теплопровідності, і завдяки цьому знаходити ефективні теплофізичні характеристики (рис. 5) уникаючи необхідності проводити емпіричні дослідження.

Отже, за допомогою описаної дискретної моделі можна:

- Розв'язувати задачу теплопровідності відносно композитних матеріалів зі складною структурою;
- Враховувати нелінійності процесів теплопровідності, неоднорідності середовища;
- Знаходити ефективні усереднені фізичні властивості матеріалів;

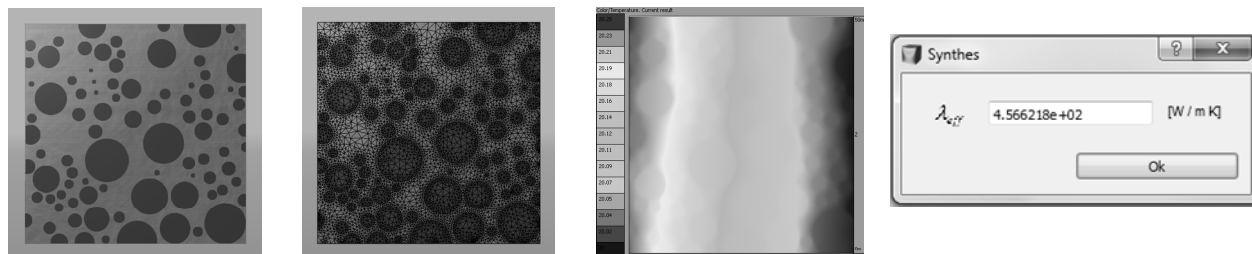


Рис. 5. Приклад розрахунку ефективного коефіцієнта теплопровідності для композита з сферичними вкрапленнями

1. Дульнев Г.Н., Заричняк Ю.П., Теплопроводность смесей и композиционных материалов: Справочная книга. – Л.: Энергия, 1976. 2. Lobur M., Farmaga I., Marikutsa U., Matviyiv O., Ciupinski L., Analysis and Problem Statement of the Optimal Thermal Design of Technical Objects // Proc. of the International Conference Microtechnology and Thermal, Lodz, Poland, 2011. 3. Шевченко В.Г., Основы физики полимерных композиционных материалов. – М., 2010. 4. Дульнев Г.Н. Тепло- и массообмен в радиоэлектронной аппаратуре. – М.: Высшая школа, 1984. 5. Лыков А.В., Теория теплопроводности. – М.: Высшая школа, 1967. 6. ДСТУ Б В.2.7-105-2000 (ГОСТ 7076-99) Матеріали і виробни будівельні. Метод визначення теплопровідності і термічного опору при стаціонарному тепловому режимі. – К.: Видання офіційне, 2001. 7. Segerlind L., Применение метода конечных элементов, пер. А.А. Шестакова, Б.Е. Победри. – М.: Мир, 1979. 8. Lienhard J.H. IV, Lienhard J.H. V, A Heat transfer textbook – 3rd ed. // Cambridge, Massachusetts, 2001. 9. Komarudin M. – Resistor network analogy of non-obtuse finite element model for electrical impedance tomography // Lampung Indonesia, 2010. 10. Нейман Л.Р., Демирчан К.С. Теоретические основы электротехники: Учебник для вузов. В 2-х т. Т. 1, – 3-е изд., перераб. и доп. – Л.: Энергоиздат, Ленингр. отд-ние, 1981. 11. Галанин М.П., Щеглов И.А., Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы. – М., 2006. 12. Скворцов А.В., Триангуляция Делоне и её применение. – Томск: Изд-во Том. ун-та, 2002. 13. Shewchuk J.R., Delaunay Refinement Mesh Generation // Pittsburgh, 1997. 14. Farmaga, Lobur M., Shmigelskiy P., Javorskiy N., Śpiewak P. – Regular and Adaptive Meshing Algorithms for Modeling of Spherical Inclusions by Finite Element Method.// TCSET'2012. Lviv–Slavske, 2012.

П. Кравець

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

МУЛЬТИАГЕНТНА ІГРОВА МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ З КОРЕЛЬОВАНИМИ СТРАТЕГІЯМИ

© Кравець П., 2012

Досліджено проблему кооперативного прийняття рішень у мультиагентних системах на основі моделі стохастичної гри з корельованими стратегіями. Сформульовано ігрову задачу, розроблено метод та алгоритм для її розв'язування. Наведено та проаналізовано результати комп'ютерного моделювання стохастичної гри з корельованими стратегіями.

Ключові слова: кооперативне прийняття рішень, стохастична гра, корельовані стратегії, мультиагентна система.

The problem of co-operative decision-making in multiagent systems on the basis of stochastic game model with the correlated strategies is investigated. The formulation of a game problem is executed, the method and algorithm are developed for its solving. Results of computer modelling of stochastic game with the correlated strategies are described and analysed.

Key words: co-operative decision-making, stochastic game, correlated strategies, multiagent system.

Вступ

Сучасний розвиток мережних інформаційних технологій спрямований у напрямку розроблення та впровадження мультиагентних систем. Мультиагентні системи використовуються для розв'язування складних проблем розподіленого штучного інтелекту для вироблення та прийняття рішень в умовах невизначеності [1]. Агент – це автономна система з елементами штучного інтелекту, яка функціонує згідно з цілями, закладеними його розробником або власником, для чого може взаємодіяти з іншими агентами та людиною, використовуючи ресурси інформаційної мережі. Мультиагентна система (МАС) – це система, утворена декількома взаємодіючими інтелектуальними агентами з властивостями: автономності – агенти повністю або частково незалежні; спеціалізації та обмеженості знань – кожен з агентів виконує вузькоспеціалізовані функції і не має цілісного уявлення про систему; децентралізації – агенти контролюють локальні ділянки розподіленої системи.

Прийняття рішень в МАС забезпечується координацією дій агентів при розподіленому розв'язуванні ними спільної задачі. Координація (узгодження) дій ґрунтується на взаємодії агентів, яка може бути змодельована за допомогою гри. Ігрова модель визначає абстрактне формулювання цільових функцій, стратегій та результатів взаємодії агентів. Агенти діють раціонально та стратегічно, вони обирають стратегії, які оптимізують їхні цільові функції, враховуючи знання про інших агентів. Стратегії агентів визначають план гри – дії або імовірності вибору дій гравців у часі.

Для дослідження МАС, що функціонують в умовах невизначеності, використовують моделі стохастичних ігор. На відміну від детермінованих ігор, у стохастичних іграх матриці виграшів апріорі не відомі. Стохастична гра задається множиною гравців, множинами чистих стратегій та матрицями розподілів випадкових виграшів (або програшів). Стохастична гра – це повторювальна математична гра в умовах невизначеності матриць виграшів, у якій гравці після незалежного випадкового вибору чистих стратегій на основі побудованого на динамічних змішаних стратегіях імовірнісного механізму отримують випадковий виграш (або програш), який використовують для адаптивної модифікації змішаних стратегій так, щоб максимізувати функції середніх виграшів (або мінімізувати функції середніх програшів) [2].

Багатоагентна стохастична гра є зручною моделлю колективного прийняття рішень в умовах невизначеності, оскільки дозволяє дослідити процеси конкуренції, кооперації, координації, самоорганізації, навчання та адаптивної поведінки гравців.

Серед двох великих класів ігор, кооперативних та некооперативних, з погляду прийняття рішень кооперативні ігри мають більшу теоретичну привабливість та практичну цінність, оскільки дають змогу враховувати коаліційну узгодженість стратегій поведінки агентів для досягнення результату за менший час та з меншими зусиллями. У кооперативних іграх гравці домовляються про стратегії поведінки у межах створених ними коаліцій. Агенти однієї коаліції мають спільну глобальну мету, якої досягають діями окремих гравців коаліції. Тим самим коаліція накладає обмеження на стратегії поведінки її учасників. Побудова кооперативної гри агентів ґрунтується на обміні інформацією, коли гравці узгоджують власні дії, обмінюються значеннями поточних вигравів, чистими або змішаними стратегіями. У некооперативних безкоаліційних іграх агенти діють незалежно і не можуть укласти домовленостей між собою.

Відомі методи ігрового вибору варіантів рішень переважно ґрунтуються на відсутності обміну інформацією між гравцями, що спрощує математичний аналіз їх збіжності до станів колективної оптимальності. Врахування обміну інформацією між гравцями дає змогу вивчати проблеми кооперативного вироблення та прийняття рішень за рахунок укладання договорів та переговорів, у результаті яких встановлюються значення функцій вигравів, визначаються сценарії кооперативної поведінки, враховуються психологічні фактори під час вибору варіантів рішень та ін. Домовленості між гравцями можуть бути обов'язковими або необов'язковими. При обов'язкових домовленостях гравці суворо дотримуються результатів переговорів, а при необов'язкових – свобода вибору гравців ніяк не обмежується. Стабільність коаліції гравців при необов'язкових домовленостях забезпечується визначенням умов, порушення яких стає не вигідним для гравців.

Під час некооперативних стохастичних ігор гравці вибирають стратегії випадково на основі незалежних розподілів імовірностей. У кооперативних іграх вибір стратегій є залежним у межах коаліції гравців. Така залежність може бути змодельована корельованими стратегіями – об'єднаним, спільним розподілом імовірностей вибору чистих стратегій гравців. Для цього до системи прийняття рішень вводиться арбітр, який на основі спільного розподілу знаходить рішення та пропонує його для виконання кожному гравцю. Для гравців це рішення не є обов'язковим – вони можуть генерувати власні незалежні від арбітра рішення. Відхилення стратегій гравців від рекомендованих арбітром значень оцінюється функцією штрафів.

Результати дослідження стохастичних ігор МАС з корельованими стратегіями не достатньо висвітлені у науковій літературі. Тому розроблення моделей стохастичних ігор для колективного вироблення і прийняття рішень, зокрема з корельованими стратегіями, є актуальною науковою та практичною проблемою, яка сьогодні привертає значну увагу дослідників МАС.

Метою роботи є побудова ігрового методу колективного прийняття рішень в умовах невизначеності з корельованими стратегіями гравців. Для досягнення мети побудовано модель стохастичної гри з корельованими стратегіями, розроблено метод, алгоритм та програмні засоби моделювання стохастичної гри для прийняття рішень в умовах невизначеності.

Модель мультиагентної стохастичної гри

Гра в умовах невизначеності задається кортежем $\Gamma_\zeta = (I, \{U^i\}_{i \in I}, \{\zeta^i(U)\}_{i \in I})$ на множині гравців $I \neq \emptyset$, які взаємодіють зі стохастичним середовищем за допомогою скінченного набору чистих стратегій $U^i = (u^i(1), u^i(2), \dots, u^i(N_i))$, $N_i \geq 2$. Вибирають чисті стратегії $u_n^i = u^i$ гравці у дискретні моменти часу $n = 1, 2, \dots$ випадково і незалежно на основі векторів змішаних стратегій $p_n^i \in S^{N_i}$, де S^{N_i} – N_i -вимірний одиничний симплекс з властивістю $\sum_{u^i \in U^i} p_n^i(u^i) = 1$, $p_n^i(u^i) \geq 0$ $\forall u^i \in U^i$. Значення комбінованої стратегії $u \in U = \times_{i \in I} U^i$ є колективним рішенням гравців, за яке кожен з них отримує випадковий програш $\xi_n^i(u_n)$.

Допустимо, що математичні сподівання програшів $M\{\xi_n^i(u)\} = v^i(u) = \text{const}$ не відомі та мають обмежений другий момент $\sup_n M\{[\xi_n^i(u)]^2\} = \sigma_i^2(u) < \infty$.

Нехай у системі колективного прийняття рішень присутній арбітр, який, діючи згідно з узагальненим розподілом $q_n \in S^N$, рекомендує гравцям обрати для реалізації дії, які утворюють комбінований варіант $a_n = (a_n^1, \dots, a_n^L) \in U$. Тут S^N – N -вимірний одиничний симплекс ($N = \prod_{i \in I} N_i$) із властивістю $\sum_{u \in U} q_n(u) = 1$, $q_n(u) \geq 0 \quad \forall u \in U$, а $L = |I|$ – кількість гравців. Гравець з номером i отримує інформацію тільки про компоненту a_n^i комбінованого варіанта $a \in U$. Цей сигнал сприймається i -м гравцем як необов'язкова пропозиція виконати дію a_n^i . Кожен гравець таємно і незалежно вибирає у момент часу n дійсний варіант дії u_n^i , за що отримує поточний програш:

$$\zeta_n^i(u_n^i, a_n^i) = \lambda \xi_n^i(u_n^i) + (1 - \lambda) \delta_n^i(u_n^i, a_n^i), \quad (1)$$

де $\xi_n^i(u_n^i)$ – поточний програш i -го гравця за вибір варіанта u_n^i , який залежить від колективного вибору гравців $u_n \in U$; $\delta_n^i(u_n^i, a_n^i)$ – штраф i -го гравця за відхилення вибору u_n^i від пропонованого арбітром рішення a_n^i ; $\lambda \in (0, 1)$ – ваговий коефіцієнт.

У [3] показано, що найкращою відповіддю u_n^i гравців є дотримання рекомендацій арбітра a_n^i , оскільки його стратегії охоплюють ігрове поле, задане на декартовому добутку чистих стратегій усіх гравців.

Поточний штраф гравця за відхилення від рішення арбітра визначимо так:

$$\delta_n^i(u_n^i, a_n^i) = |ORD(u_n^i) - ORD(a_n^i)| / N_i,$$

де $ORD(\cdot)$ – функція визначення порядкового номеру чистої стратегії; $\delta_n^i \in [0, 1)$.

Ефективність поточних ходів гри оцінюється функціями середніх програшів гравців:

$$Z_n^i = \frac{1}{n} \sum_{t=1}^n \zeta_t^i \quad \forall i \in I. \quad (2)$$

Метою кожного гравця є мінімізація функції середніх програшів (2):

$$\lim_{n \rightarrow \infty} Z_n^i \rightarrow \min \quad \forall i \in I. \quad (3)$$

Розв'язки гри шукатимемо у множині точок корельованої рівноваги (CE, Correlated Equilibrium), яка визначається об'єднаним розподілом стратегій гравців $q_n \in S^N$, коли кожен агент не має мотивації, щоб відхилитися від домовленостей в односторонньому порядку. В умовах повної інформації маємо:

$$\sum_{a_{-i} \in A_{-i}} q^{CE}(a_{-i} | a_i) v^i(a_{-i}, a_i) \leq \sum_{a_{-i} \in A_{-i}} q^{CE}(a_{-i} | a_i) v^i(a_{-i}, \tilde{a}_i), \quad \forall a_i, \tilde{a}_i \in U_i, \quad \forall i \in I, \quad (4)$$

де $U_{-i} = \prod_{j=1, j \neq i}^L U_j$, $U = U_{-i} \times U_i$, $a_{-i} \in U_{-i}$, $a = (a_{-i}, a_i) \in U$, $v^i(a) = M\{\zeta_n^i(a)\}$, $q^{CE}(a_i) = \sum_{a_{-i} \in A_{-i}} q^{CE}(a_{-i}, a_i)$, $q^{CE}(a_{-i} | a_i) = q^{CE}(a_{-i}, a_i) / q^{CE}(a_i)$, $q^{CE}(a_i) > 0$.

Домноживши обидві частини нерівності (4) на $q^{CE}(a_i)$, отримаємо еквівалентну нерівність:

$$\sum_{a_{-i} \in A_{-i}} q^{CE}(a_{-i}, a_i) v^i(a_{-i}, a_i) \leq \sum_{a_{-i} \in A_{-i}} q^{CE}(a_{-i}, a_i) v^i(a_{-i}, \tilde{a}_i). \quad (5)$$

При $q^{CE}(a_{-i}, a_i) = \prod_{\substack{j \in I \setminus \{i\}; \\ a_j \in a_{-i}}} p_j^{NE}(a_j)$ з (5) отримуємо умову рівноваги за Нешем (NE, Nash

Equilibrium). Усі точки рівноваги за Нешем є точками корельованої рівноваги [3]. Якщо $\forall i \in I$ $q^{CE}(a_{-i} | a_i) = q^{CE}(a_{-i} | \tilde{a}_i)$, $\forall a_i, \tilde{a}_i \in U_i$, $\forall a_{-i} \in U_{-i}$, $\forall q^{CE}(a_i), q^{CE}(\tilde{a}_i) > 0$, то корельована рівновага є також рівновагою за Нешем.

Множина точок СЕ-рівноваги є непорожньою, опуклою та компактною і в умовах повної інформації може бути обчислена за допомогою методів лінійного програмування. У випадку мінімізації сумарного програшу гравців задача лінійного програмування для знаходження q^{CE} може бути сформульована так:

$$\begin{aligned} \sum_{a \in U} \sigma(a) \sum_{i=1}^L v^i(a) &\rightarrow \min_{\sigma}, \\ \sum_{a_{-i} \in U_{-i}} q(a_{-i}, a_i) (v^i(a_{-i}, a_i) - v^i(a_{-i}, \tilde{a}_i)) &\leq 0, \\ \forall i \in I, \forall a_i \in U_i, \forall \tilde{a}_i \in U_i, \\ q(a) &> 0 \quad \forall a \in U, \\ \sum_{a \in U} q(a) &= 1. \end{aligned}$$

Значення середніх програшів $v^i(a) \quad \forall a \in U$ апіорі не відомі гравцям в умовах невизначеності. Тому корельована рівновага визначається для випадкових послідовностей $\{u_n\}$ асимптотичною умовою:

$$\forall i \in I \quad \overline{\lim}_{n \rightarrow \infty} [Z_n^i(\{u_n\}) - Z_n^i(\{\hat{u}_n\})] \leq 0, \quad (6)$$

де нерівності (6) виконуються з імовірністю 1, $\hat{u}_n = u_n \setminus u_n^i + \tilde{u}_n^i \in U$; $u_n^i, \tilde{u}_n^i \in U^i$.

Отже, в умовах невизначеності гравці повинні за спостереженнями власних поточних програшів ζ_n^i здійснювати незалежний вибір своїх чистих стратегій $u^i \in U^i$ так, щоб сформована послідовність варіантів рішень $\{u_n^i\} \quad \forall i \in I$ при $n \rightarrow \infty$ задовольнила умову асимптотичної оптимальності (6).

Метод розв'язування ігрової задачі

Сформуємо послідовності $\{u_n^i\}$ з потрібними властивостями на основі самонавчального рекурентного методу зміни змішаних стратегій гравців [4]:

$$p_{n+1}^i = \pi_{\varepsilon_{n+1}}^{N_i} \{p_n^i - \gamma_n R(\zeta_n^i, p_n^i, u_n^i)\} \quad \forall i \in I, \quad (7)$$

де $\pi_{\varepsilon_{n+1}}^{N_i}$ – проектор на одиничний ε -симплекс, який забезпечує нормалізацію вектора імовірностей вибору варіантів дій $p_{n+1}^i \in S_{\varepsilon_{n+1}}^{N_i} \subseteq S^{N_i}$; $\varepsilon_n > 0$ – монотонно спадна послідовність величин, яка регулює розширення ε -симплексу; $\gamma_n > 0$ – монотонно спадна послідовність величин, що регулює крок методу; $R(\zeta_n^i, p_n^i, u_n^i)$ – поточний крок методу; ζ_n^i – поточний програш i -го гравця.

Значення ζ_n^i обчислюється згідно з (1) із врахуванням поточної стратегії гравця та стратегії арбітра.

Вектор $R(\zeta_n^i, p_n^i, u_n^i)$ задає середній напрямок на оптимальний розв'язок стохастичної гри. В умовах невизначеності значення $R(\zeta_n^i, p_n^i, u_n^i)$ визначається на основі методу стохастичної апроксимації [4 – 6]. Так, при $R(\zeta_n^i, p_n^i, u_n^i) = \zeta_n^i / (e^T(u_n^i) p_n^i)$ з (7) отримуємо градієнтний метод пошуку стратегій, які задовольняють умову рівноваги за Нешем [4]. При $R(\zeta_n^i, p_n^i, u_n^i) = \zeta_n^i (e(u_n^i) - p_n^i)$ з (7) отримуємо метод доповняльної нежорсткості, призначений для пошуку вирівнювальних стратегій [7]. Тут $e(u_n^i)$ позначає орт, що відповідає вибраній чистій стратегії $u_n^i \in U^i$.

Змішана стратегія арбітра q_n може визначатися стратегіями гравців, що дає змогу знаходити розв'язки гри, які задовольняють умову рівноваги за Нешем. Для цього гравці повідомляють арбітру значення поточних змішаних стратегій $p_n^i \quad \forall i \in I$. Елементи вектора q_n є імовірностями

вибору комбінованих стратегій гравців $\forall u \in U$ і обчислюються добутком імовірностей відповідних чистих стратегій гравців $u_i \in u \quad \forall i \in I$:

$$q_n = \left(\prod_{i \in I} p_n^i(u_i) \mid u_i \in u, \forall u \in U \right),$$

де $U = \times_{i \in I} U^i$.

У загальному змішана стратегія q_n арбітра не залежить від стратегій гравців. Тоді в умовах невизначеності її можна задати подібно до (7):

$$q_{n+1} = \pi_{\varepsilon_{n+1}}^N \left\{ q_n - \gamma_n \Upsilon(\psi_n, q_n, u_n) \right\}, \quad (8)$$

де $N = \prod_{i \in I} N_i$; $\Upsilon(\psi_n, q_n, u_n)$ – поточний крок методу; ψ_n – отриманий від середовища поточний програш арбітра.

Середовище гри задається математичними сподіваннями програшів усіх гравців $v^i(a) \forall a \in U, \forall i \in I$.

Вважається, що арбітр генерує правильні (оптимальні у середньому) рекомендації щодо вибору стратегій гравців. Для цього використовується навчена змішана стратегія q^* або рекурентний метод (8) забезпечує більшу середньоквадратичну швидкість навчання арбітра, ніж метод навчання агентів (7).

Вибір чистої стратегії $a_n = (a_n^1, \dots, a_n^L)$ арбітра на основі розподілу q_n може здійснюватись детерміновано або випадково.

Детермінована процедура полягає у визначенні максимального значення розподілу q_n :

$$a_n = \left\{ U(k) \mid k = \arg \max_{j=1}^N q_n(j), N = \prod_{i \in I} N_i \right\}. \quad (9)$$

Випадкові реалізації чистих стратегій арбітра визначаються з імовірностями q_n :

$$a_n = \left\{ U(k) \mid k = \arg \left(\min_k \sum_{j=1}^k q_n(j) > \omega \right), k = 1..N, N = \prod_{i \in I} N_i \right\}, \quad (10)$$

де $\omega \in [0, 1]$ – випадкова величина з рівномірним розподілом.

Узагальнений розподіл q_n арбітра ґрунтується на знанні глобального стану гри і дає змогу згенерувати у середньому оптимальний комбінований сигнал a_n . На його основі визначаються рекомендовані для вибору гравцям чисті стратегії: $a_n = (a_n^1, \dots, a_n^L) \in U$. У результаті, надісланий i -му гравцю сигнал a_n^i несе інформацію про оптимальний (у середньому) вибір поточної дії з врахуванням рішень інших гравців. Однак воля гравців ніяк не обмежується, і вони мають право на автономний незалежний вибір стратегій.

Власні чисті стратегії гравців вибираються аналогічно до (9) або (10). Наприклад, випадковий вибір чистої стратегії здійснюється так:

$$u_n^i = \left\{ U^i(k) \mid k = \arg \left(\min_k \sum_{j=1}^k p_n^i(j) > \omega \right), k = 1..N_i \right\} \quad \forall i \in I. \quad (11)$$

Реалізоване колективне рішення $u_n = (u_n^1, \dots, u_n^L)$ визначає поточні програші гравців $\xi_n^i(u_n) \forall i \in I$. Реальне значення поточних програшів ζ_n^i додатково враховує стратегію арбітра і обчислюється згідно (1).

Алгоритм розв'язування стохастичної гри

1. Задати початковий момент часу $n=0$; множину I та кількість $L=|I|$ гравців; кількості чистих стратегій $N_i \quad \forall i \in I$; значення чистих стратегій $U^i = (u^i(1), \dots, u^i(N_i))$; початкові значення

матриць програвів $v^i(u) \quad \forall u \in U, \forall i \in I$; початкові значення змішаних стратегій $p_n^i(u_i) = 1/N_i \quad \forall i \in I$; точність обчислення $\varepsilon > 0$; максимальну кількість кроків повторювальної гри n_{\max} .

2. Обчислити значення вектора змішаної стратегії арбітра q_n згідно з (8).
3. На основі стратегії q_n виконати випадковий вибір дії арбітра $a_n = (a_n^1, \dots, a_n^L)$ згідно з (10).
4. Згідно з (11) виконати випадковий вибір дій гравців u_i на основі змішаних стратегій $p_n^i \quad \forall i \in I$.
5. Отримати поточні програти гравців $\zeta_n^i \quad \forall i \in I$ згідно з (1).
6. Модифікувати вектори змішаних стратегій p_{n+1}^i згідно з (7).
7. Якщо $\|p_{n+1}^i - p_n^i\| < \varepsilon \quad \forall i \in I$ (або $n < n_{\max}$), то задати $n := n + 1$ і перейти на крок 2.
8. Вивести розраховані значення змішаних стратегій гравців $p_n^i \quad \forall i \in I$. Кінець.

Контрольний приклад

Використовуючи розроблений алгоритм, виконаємо розв'язування стохастичної гри двох агентів з двома чистими стратегіями та матрицями програвів, заданими у таблиці.

Матриці програвів гравців

Стратегії	Перший гравець		Другий гравець	
	$p^2(a_2[1])$	$p^2(a_2[2])$	$p^2(a_2[1])$	$p^2(a_2[2])$
$p^1(a_1[1])$	0.5	0.1	0.9	0.1
$p^1(a_1[2])$	0.3	0.2	0.1	0.9

Елементи $v^i(u) \quad \forall u \in U$ матриць вигравів не відомі гравцям апіорі. Замість них гравці отримують нормально розподілені випадкові поточні програти

$$\xi^i = Normal(v^i(u), d^i(u))$$

з математичними сподіваннями $v^i(u)$ та дисперсіями $d^i(u) \quad \forall u \in U$.

Нормально розподілені випадкові величини отримано з допомогою суми дванадцяти рівномірно розподілених випадкових чисел $\omega \in [0, 1]$:

$$\xi_n^i(u_n, \omega) = v^i(u_n) + \sqrt{d(u_n)} \left(\sum_{j=1}^{12} \omega_j - 6 \right),$$

де $u_n \in U$.

Динаміка змішаних стратегій гравців визначається рекурентним перетворенням

$$p_{n+1}^i = \pi_{\varepsilon_{n+1}}^{N_i} \left\{ p_n^i - \gamma_n \zeta_n^i (e(u_n^i) - p_n^i) \right\} \quad \forall i \in I, \quad (12)$$

отриманим за методом стохастичної апроксимації умови доповняльної нежорсткості:

$$V^i(U^{i \setminus \{i\}}, u^i) = V^i, \quad \forall u^i \in U^i, \quad \forall i \in I.$$

де $V^i = \sum_{u^i \in U^i} p^i(u^i) V^i(U^{i \setminus \{i\}}, u^i)$ – функція середніх вигравів гравців.

Параметри γ_n та ε_n є дійсними монотонно спадними величинами і можуть бути обчислені так:

$$\gamma_n = \gamma n^{-\alpha}, \quad \varepsilon_n = \varepsilon n^{-\beta}, \quad (13)$$

де $\gamma > 0$; $\alpha \in (0, 1]$; $\varepsilon > 0$; $\beta > 0$.

Поточна змішана стратегія q_n арбітра визначається згідно з (8), або використовується попередньо навчена стратегія.

Збіжність стохастичної гри оцінюється такими параметрами:

1) евклідовою нормою вектора змішаних стратегій арбітра гри:

$$\|q_n\| = \sqrt{\sum_{j=1}^N (q_n[j])^2}; \quad (14)$$

2) усередненими за кількістю гравців середніми програшами:

$$Z_n = \frac{1}{|I|} \sum_{i=1}^{|I|} Z_n^i; \quad (15)$$

3) усередненою за кількістю гравців похибкою доповняльної нежорсткості:

$$\Delta_n = \frac{1}{|I|n} \sum_{t=1}^n \sum_{i \in I} \|p_t^i - \tilde{p}_t^i\|, \quad (16)$$

де $N = \prod_{i \in I} N_i$ – кількість комбінованих стратегій гравців; $|I|$ – кількість гравців;

$\tilde{p}_n^i = p_n^i V_n^i(U^{i(i)})/V_n^i$ – поточна оцінка вирівнювальної стратегії для умови доповняльної нежорсткості; V_n^i – поточне значення функції середніх виграшів i -го гравця.

Для визначення асимптотичного порядку швидкості збіжності використано метод моментів Чжуна [8]:

$$\overline{\lim}_{n \rightarrow \infty} n^\theta M\{\Delta_n\} \leq \vartheta, \quad (17)$$

де θ – параметр порядку; ϑ – величина швидкості збіжності. Більшому θ та меншому ϑ відповідає більша швидкість збіжності ігрового методу. Довжина досліджуваної вибірки становить 10 тис. кроків.

Враховуючи (17), поведінка процесу Δ_n у часі апроксимована залежністю $\Delta_n = \vartheta/n^\theta$, де $\vartheta > 0$, $\theta \in (0,1]$, $n = 1, 2, \dots$. Після логарифмування отримаємо лінійне співвідношення:

$$\lg \Delta_n = \lg \vartheta - \theta \lg n. \quad (18)$$

Параметр $\theta = \lg \Delta_n / \lg n$ вказує на порядок швидкості збіжності стохастичної гри. Для його експериментального обчислення виконано апроксимацію випадкового процесу $\lg \Delta_n$ лінійною залежністю (18) на відрізку $\lg n \in [3, 4]$ за методом найменших квадратів.

Згладжування випадкової складової швидкості збіжності та виділення порядку цієї швидкості виконано усередненням по реалізаціях випадкового процесу Δ_n .

Результати моделювання

Результати комп'ютерного моделювання стохастичної гри з корельованими стратегіями подано на рис. 1–5.

На рис. 1 у логарифмічному масштабі зображено графіки зміни у часі характеристик збіжності стохастичної гри з корельованими стратегіями. Графік 1 відповідає функції евклідової норми вектора змішаних стратегій арбітра гри $\|q_n\|$ (14), графік 2 – усередненої по кількості гравців функції середніх програшів Z_n (15), графік 3 – усередненої за гравцями функції похибки доповняльної нежорсткості Δ_n (16). Результати отримано для таких значень параметрів гри: $L = |I| = 2$, $N_i = 2 \forall i \in I$, $\lambda = 0.5$, $\alpha = 0.5$, $\beta = 2$, $\gamma = 1$, $\varepsilon = 0.999/N_i$, $d = 0.01$.

Спрямування логарифмічної функції норми змішаних стратегій арбітра $\|q_n\|$ до нуля свідчить про досягнення розв'язку гри у вершині одиничного симплексу. Динаміка функції Z_n демонструє мінімізацію середніх програшів гравців. Зменшення Δ_n ілюструє наближення змішаних стратегій гравців до оптимальних значень у межах одиничного симплексу.

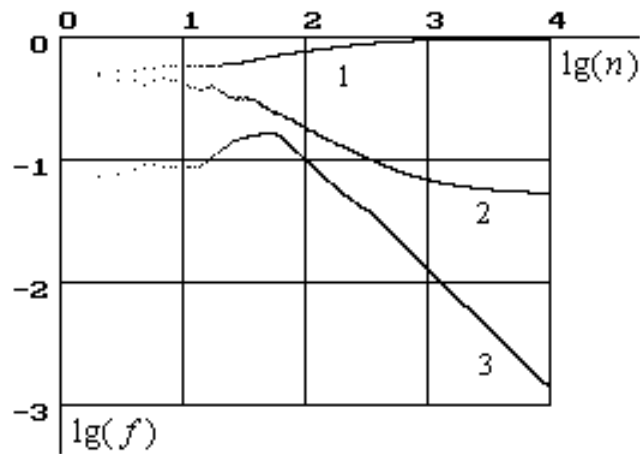


Рис. 1. Характеристики збіжності стохастичної гри з корельованими стратегіями

Загальний вигляд функцій середніх програшів гравців та отримана під час їх мінімізації траєкторія змішаних стратегій на одиничному симплексі зображені на рис. 2. Координатні осі позначають значення одного з елементів векторів змішаних стратегій $p^i = (p_i, 1 - p_i)$, $i=1..2$ для гри 2×2 . Метод (12) забезпечує досягнення оптимального розв'язку гри у точці $(1, 0)$, що відповідає рівновазі за Нешем у чистих стратегіях.

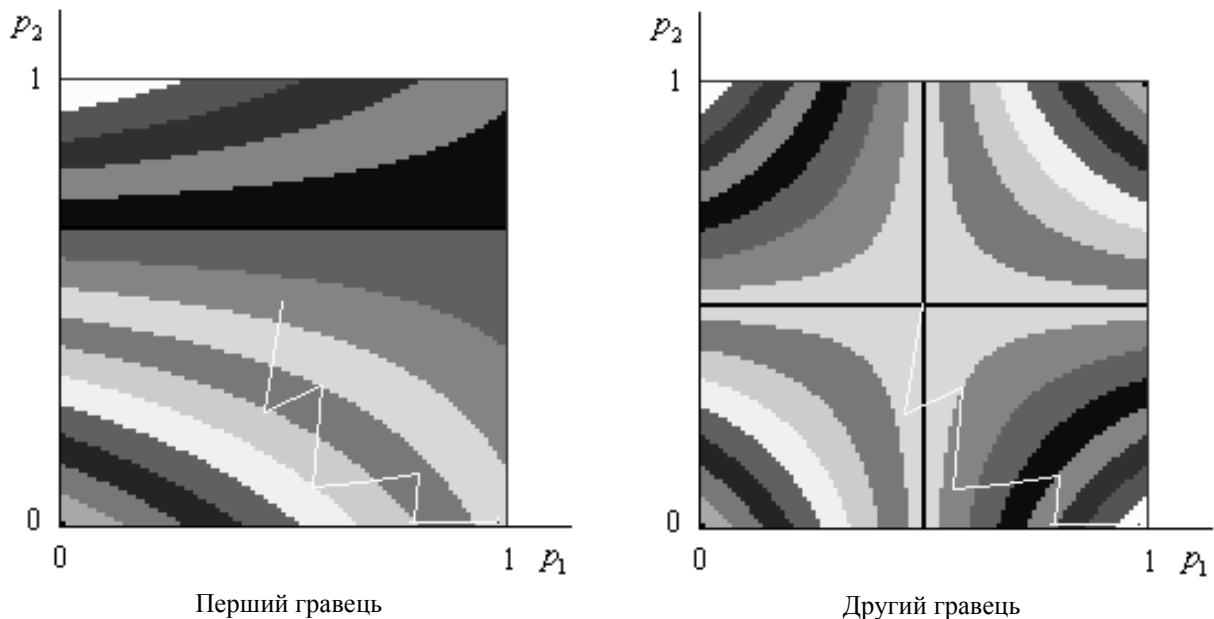


Рис. 2. Динаміка стратегій та функції середніх програшів гравців

Швидкість збіжності стохастичної гри значною мірою визначається параметрами середовища гри та параметрами методу навчання векторів змішаних стратегій гравців.

Залежність функції Δ_n від дисперсії програшів d зображено на рис. 3. Із зростанням дисперсії зменшується величина ϑ швидкості збіжності. Значення дисперсії програшів гравців практично не впливає на порядок θ швидкості збіжності стохастичної гри.

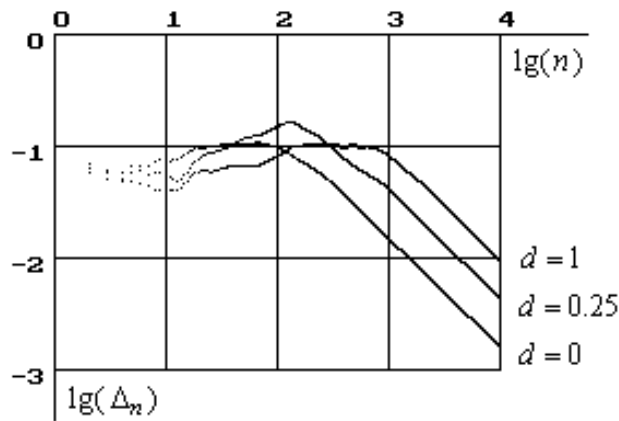


Рис. 3. Вплив дисперсії програвів d на збіжність стохастичної гри

Під час комп'ютерного моделювання виявлено нелінійну залежність швидкості збіжності стохастичної гри від параметра α методу (12). Відповідні графіки функції Δ_n для різних значень параметра α наведено на рис. 4.

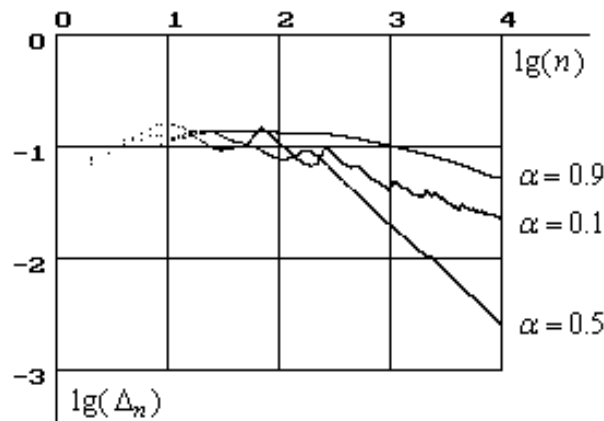


Рис. 4. Вплив параметра α на збіжність стохастичної гри

Встановлено, що для заданих матриць програвів найбільшого порядку швидкості збіжності ігрового методу з корельованими стратегіями досягається при $\alpha = 0.5$.

Швидкість збіжності стохастичної гри з навченим арбітром є більшою від гри без арбітра, що продемонстровано на рис. 5. Графік функції Δ_n , отриманий для значення вагового коефіцієнта $\lambda = 1$ поточних програвів (1), відповідає грі без арбітра, графік для коефіцієнта $\lambda = 0.5$ – гри з частковою довірою арбітру, а графік для коефіцієнта $\lambda = 0$ – гри з повною довірою арбітру.

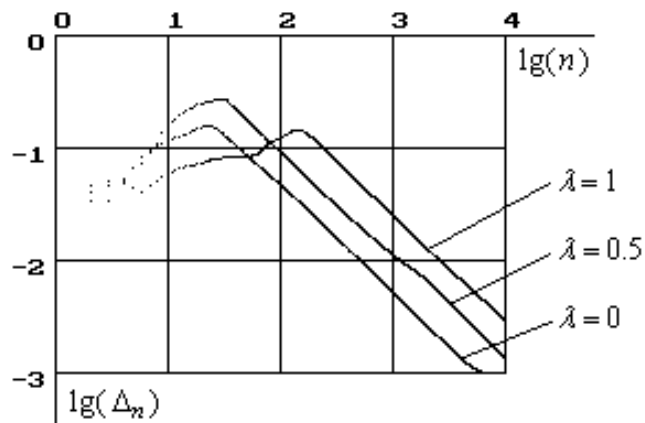


Рис. 5. Вплив параметра λ на збіжність стохастичної гри

Як видно на рис. 5, у грі з навченим арбітром у середньому забезпечується більша величина швидкості збіжності, ніж у грі без використання арбітра. Найбільша швидкість збіжності досягається у випадку, коли гравці абсолютно дотримуються рекомендацій навченого арбітра.

Висновки

Розглянута стохастична гра з корельованими стратегіями є продуктивною моделлю кооперативного прийняття рішень в МАС. Кооперація дій агентів задається узагальненим розподілом імовірностей, визначеним на декартовому добутку чистих стратегій коаліції гравців. Для цього у стохастичну гру введено додаткового гравця-арбітра, який виробляє статистично оптимальні рекомендаційні рішення, не обов'язкові для виконання коаліцією гравців. Зберігаючи незалежність, гравці отримують штраф за відхилення власного вибору від рекомендованого арбітром рішення. Дотримання гравцями визначених арбітром стратегій моделює коаліційну стійкість колективу гравців у процесі вироблення ними кооперативного рішення.

Розв'язування стохастичної гри в умовах невизначеності забезпечується динамічними змішаними стратегіями гравців, для цілеспрямованого формування яких запропоновано проєкційний рекурентний метод, отриманий на основі стохастичної апроксимації вектора, що задає напрямок на оптимальний колективний розв'язок у межах одиничного симплексу. Апроксимований вектор залежить від значень поточних програшів гравців, величина яких визначається випадковою реакцією середовища на реалізовані стратегії та відхиленням стратегій від рекомендованого арбітром значення. У зв'язку з цим апроксимований вектор у середньому вказує на оптимальний колективний розв'язок стохастичної гри з корельованими стратегіями.

У результаті комп'ютерного моделювання встановлено, що за однакових початкових умов кооперативна стохастична гра з корельованими стратегіями у середньому має більшу швидкість збіжності, ніж некооперативна гра за рахунок того, що арбітр генерує оптимальні у середньому рішення. Дотримуючись раціональних рекомендацій арбітра, що враховується штрафами за відмову від їх реалізації, гравцям потрібна у середньому менша кількість кроків для знаходження розв'язку гри в умовах невизначеності.

1. Weiss G. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* / G. Weiss, editor. – Springer Verlag, Berlin, 1996. – 643 pp. 2. Fudenberg D. *The Theory of Learning in Games* / D. Fudenberg, D.K. Levine. – Cambridge, MA: MIT Press, 1998. – 292 pp. 3. Greenwald A. *Correlated Q-learning* / A. Greenwald, K. Hall // *Proceedings of the Twentieth International Conference on Machine Learning*. – 2003. – P. 242–249. 4. Назин А.В. *Адаптивный выбор вариантов: Рекуррентные алгоритмы* / А.В. Назин, А.С. Позняк. – М.: Наука, 1986. – 288 с. 5. Граничин О.Н. *Введение в методы стохастической аппроксимации и оценивания: Учеб. пособие* / О.Н. Граничин. – СПб.: Изд-во СПб-го ун-та, 2003. – 131 с. 6. Вазан М. *Стохастическая аппроксимация* / М. Вазан. – М.: Мир, 1972. – 295 с. 7. Мулен Э. *Теория игр с примерами из математической экономики* / Э. Мулен. – М.: Мир, 1985. – 200 с. 8. Невельсон М.Б. *Стохастическая оптимизация и рекуррентное оценивание* / М.Б. Невельсон, Р.З. Хасьминский. – М.: Наука, 1972. – 304 с.

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПЕРЕХІДНИХ ПРОЦЕСІВ В СИСТЕМАХ ТРАНСПОРТУВАННЯ ГАЗУ

© Притула Н., 2012

Розглянуті задачі розрахунку параметрів нестационарних процесів у разі перемикань, які використовуються в управлінні газопотоками. Запропоновано алгоритми формування порядку перемикань. Запропоновано алгоритми, які забезпечують, за значних коливань витрати та тиску, стійку роботу методів розрахунку параметрів газопотоків.

Ключові слова: газотранспортна система, управління, нестационарний процес, стійкість методу, режим роботи.

In this paper, we consider problems of calculation of parameters of unsteady processes when switchings which are used in gas movement management. We have offered algorithms of formation of an order of switchings. We have also offered algorithms which provide, at considerable fluctuations of an expense and pressure, firmness of methods of calculation of parameters of gas of streams.

Key words: gas transport system, management, unsteady process, firmness of a method, operating mode.

Вступ

Газотранспортна система України є керованим об'єктом. Управління газопотоками здійснюється зміною: стану запірної арматури, продуктивності газоперекачувальних агрегатів, параметрів джерел та відборів газу тощо. Окремі зміни вимагають певної послідовності дій в часі, що визначається регламентом управління об'єктом. Перехідні процеси в системах в місцях проведених змін є значними як за амплітудою, так і швидкістю їх проходження. Це зумовлює будувати такі розрахункові методи, які б адаптувалися до параметрів перехідних процесів. На зміни, як її частоту, впливають також зміни параметрів відборів та джерел газу. Порядок проведення змін та їх частота належать до компетенції алгоритмів оптимального управління, які в цій роботі не розглядаються.

1. Регламент роботи технологічних об'єктів

Всі види управління роботою об'єктів є регламентованими як за операціями, так і часом їх проходження. Всі регламенти були відпрацьованими під час проведення числових експериментів. Прийнятий для використання регламент забезпечує максимально адекватну реальним об'єктам роботу і максимальну стійкість числових розрахункових методів. Розглянемо два основні об'єкти управління газопотоками в ГТС-крани та компресорні станції.

Регламент роботи крана складається із двох частин – закриття та відкриття:

- **закриття крана** відбувається миттєво (справа та зліва від крана, якщо його вважати ребром графа ставиться нульова гранична умова на витрату, тобто $Q = 0$).
- **відкриття крана** відбувається упродовж певного часу. Для стійкості роботи методу, на час моделювання відкриття крана, крок за часом зменшували (в проведених експериментах крок за часом зменшували в 10 разів). Під час цього перепад тисків e спрямовується до 1 ($e \rightarrow 1$) за таким алгоритмом:

- 1) об'єктом відношення тисків на крані e (який є наявним для певного часу t);

2) для певного часу t (для всіх ітерацій певного t) в рівняння входить степінь стиснення e^* , який знаходиться з однієї із двох формул: $e^* = e + \frac{1-e}{2^{9-n}}$, де n – число зменшення кроку за часом після відкриття крана, $n = (0,9)$, або $e^* = e + \frac{1-e}{2}$. Перша формула як функція – опуклий варіант, інший – вгнутий.

Регламент роботи компресорної станції складається з трьох частин:

- **вимкнення КС** (переведення КС з робочого режиму “на прохід”) відбувається миттєво, тобто наявний ступінь стиснення спрямовують до одиниці, тобто приймають $e = 1$;
- **увімкнення КС** здійснюється протягом N_{on} кроків, спрямуванням ступеня стиснення e від e_{min} до певного значення $e_{default}$ яке встановлюється заздалегідь. Після чого розраховуються параметри режиму роботи ГТС разом із увімкненою вказаним способом КС, зокрема, такий параметр, як потужність КС. Зауважимо, що в цей же момент розраховуються оптимальні параметри коридору зміни контрольованого параметра P_2 для тиску на виході КС. Якщо такі параметри існують, то їх вважаємо для P_2 за оптимальні;
- **зміна потужності КС** (збільшення, або зменшення) відбувається пропорційно для всіх цехів багатоцехових КС, які працюють, на певний відсоток k . У цьому варіанті k визначається як відсоток відхилення контрольованого параметра, який призводить до зміни роботи КС, від оптимального значення (якщо значення контрольованого параметра вийшло за технологічні/оптимальні допустимі межі).

2. Математична модель газових потоків

Моделлю руху газу на кожній з ділянок є система взаємопов’язаних диференціальних рівнянь у частинних похідних з початково-граничними умовами. Ця система містить такі рівняння [1]:

$$\begin{cases} \frac{\partial(rv)}{\partial t} + \frac{\partial P}{\partial x} + ar \frac{\partial}{\partial x} \left(\frac{v^2}{2} \right) + rg \frac{dh}{dx} + \frac{Irv^2}{2S} = 0, \\ \frac{\partial r}{\partial t} + \frac{\partial(rv)}{\partial x} = 0, \\ \frac{\partial}{\partial t} (rE) + \frac{\partial}{\partial x} rv \left(E + \frac{P}{r} \right) = \frac{4K(T_{ep} - T)}{D} - rvg \frac{dh}{dx}, \end{cases} \quad (1)$$

що замикається рівнянням стану газу $P = zrRT$.

Тут r, v, P невідомі густина, швидкість руху, температура і тиск газу; a – коефіцієнт Коріоліса, g – прискорення вільного падіння; h – висота залягання труби; I – коефіцієнт гідравлічного опору; K – коефіцієнт теплопередачі від труби до ґрунту; T_{ep} – температура ґрунту; h – висота залягання труби; E – повна енергія одиниці маси.

$$E = i - \frac{P}{r} + \frac{v^2}{2},$$

$$di = \frac{\partial i}{\partial T} dT + \frac{\partial i}{\partial P} dP = C_p dT + \left[\frac{1}{r} - T \left(\frac{\partial(1/r)}{\partial T} \right)_p \right] dP, \quad C_p = \left(\frac{\partial i}{\partial T} \right)_r.$$

Для обчислення коефіцієнта стисливості z , який описує відмінність реального газу від ідеального, застосовують емпіричну формулу

$$z = (1 + (24 - 0.21t^{\circ}C)p)^{-1} 10^{-4},$$

де $t^{\circ}C$ – температура газу за Цельсієм.

Для відшукування розв'язку системи (1) доповнюємо її відповідними крайовими умовами на швидкість і/або на тиск газу. Розв'язуємо методом скінченних елементів [2].

Для розрахунку режиму роботи КС використовуються напівемпіричні співвідношення [4], які пов'язують параметри газу на його входах (тиск, витрата, температура, густина газу) та виходах (тиск, температура). Результатом розрахунку є: кількість ГПА, схема їх з'єднання, гідравлічні втрати на вхідних та вихідних шлейфах та параметри роботи кожного ГПА (потужність приводу, об'єм споживання паливного газу, обороти відцентрового нагнітача, ступінь підвищення тиску). Так потужність приводу у разі політропних процесів розраховується за формулою [3]

$$N_{пол} = a \frac{m}{m-1} P_i Q_i \left(\left(\frac{P_j}{P_i} \right)^{(m-1)/m} - 1 \right), \quad (2)$$

де a – емпіричний коефіцієнт; m – показник політропи; Q_i – об'ємна витрата газу на вході.

3. Аналіз числових експериментів. Вплив параметрів алгоритмів змін на параметри нестационарних режимів

Розглянемо розрахунок ділянки газопроводу довжиною 100 км і з внутрішнім діаметром 1400 мм, потік газу в якій є стаціонарним. За граничні виберемо початкові умови. Розглянемо два випадки задання граничних умов – на тиски і на витрату (див. рис. 1). Зліва на рис. 1 бачимо зміну витрати (згори) та тиску (знизу) у випадку граничних умов на тиски. Справа на тому ж рисунку бачимо графіки, отримані у випадку задання граничних умов на витрату. Як бачимо, незначні коливання розрахованих параметрів в початковий момент часу, які найвірогідніше пов'язані з неузгодженістю початкових та граничних умов. Ці коливання є незначними (за межами точності вимірювальних приладів) і тому не впливають на розрахункові параметри.

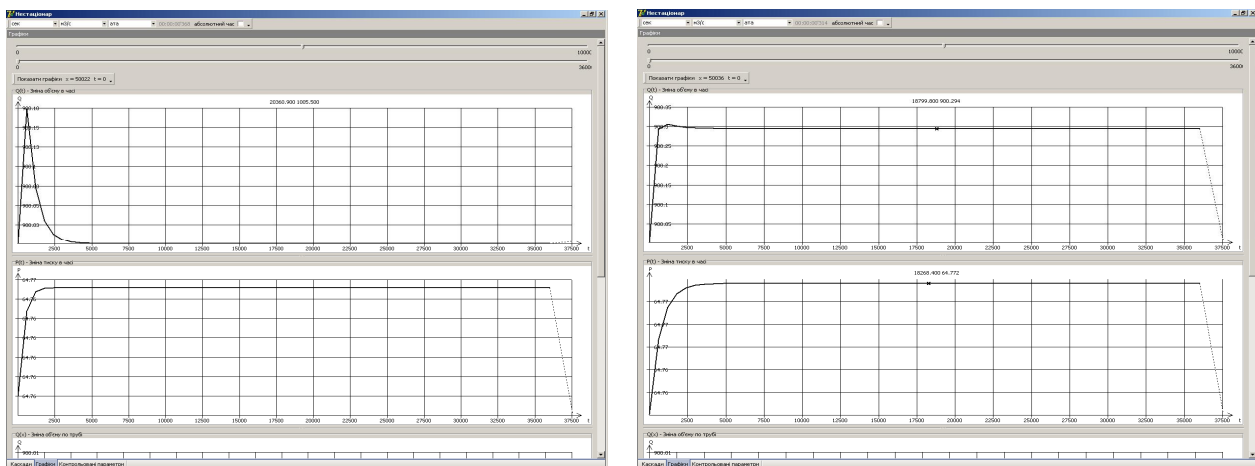


Рис. 1. Розрахунок стаціонарного режиму методом, розрахованим для нестационарних режимів

Відкриття та закриття кранів у газотранспортній системі відбувається доволі часто. Якщо відбувається зміна стану одночасно в декількох кранах, то відбуваються значні за амплітудою та швидкістю збурення параметрів потоків газу. Розрахунок у таких випадках може бути нестійким. В таких випадках потрібно відпрацювати такий регламент зміни стану, щоб і стійкість методу була достатньою, та розраховані параметри газодинамічних процесів були близькими до реальних.

Розглянемо приклад. Довжина ділянки газопроводу становить 100 км, а внутрішній діаметр – 1400 мм. Кран розміщений посеред ділянки. Закриття крана відбувається на 7200 с, а відкриття на 9000 с. Крок методу за часовою координатою – 600 с. В інтервалах часу 7200–7800 і 9000–9600 крок за часом – 60 с. Результати розрахунку подані на рис. 2, 3. На рис. 4, 5 представлені результати вимкнення КС.

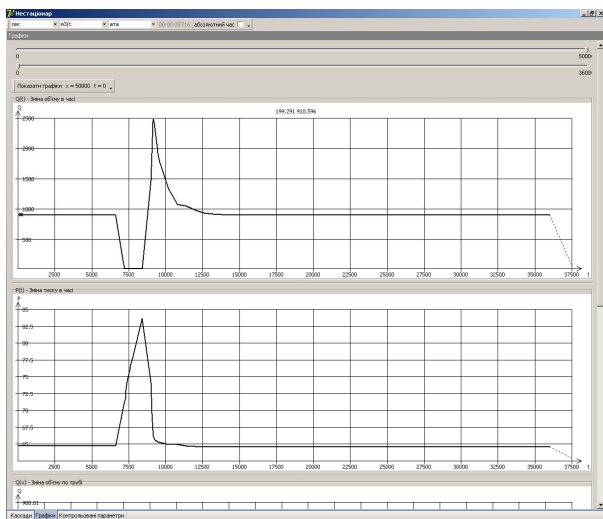


Рис. 2. Зліва від крана (з боку джерела)

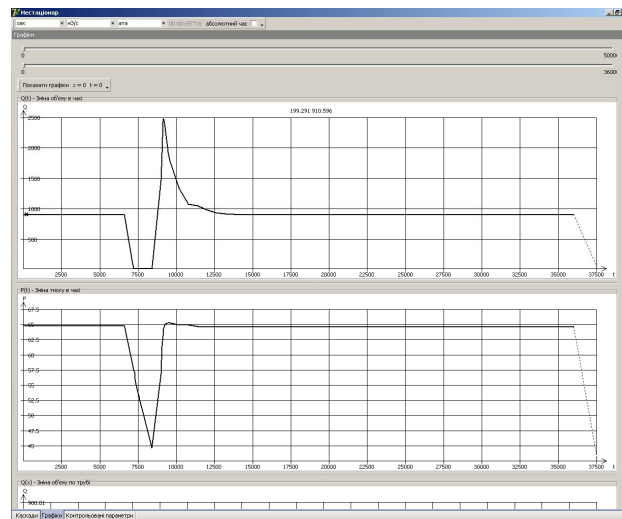


Рис. 3. Справа від крана (з боку відбирання)

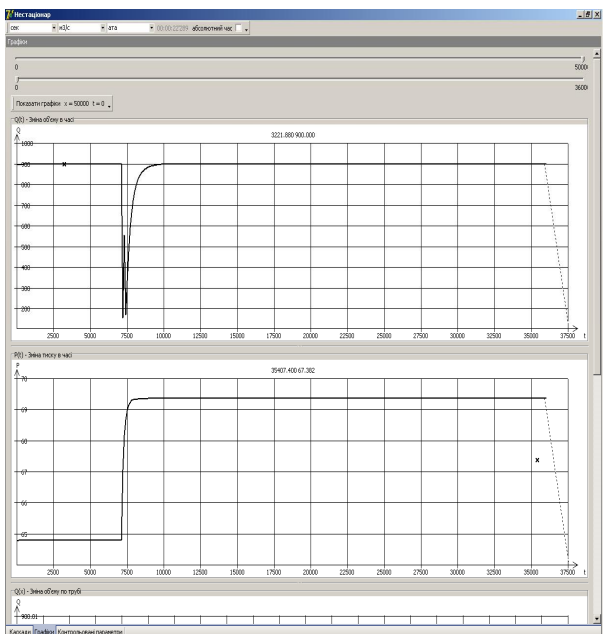


Рис. 4. Зміна витрати та тиску на вході КС

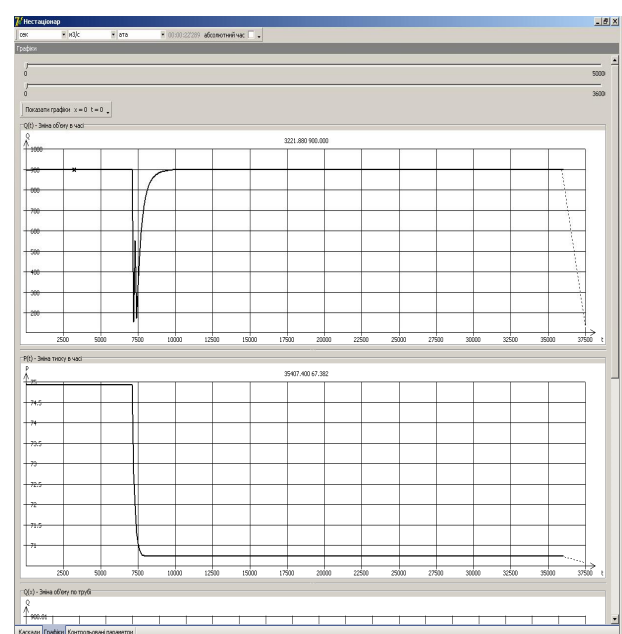


Рис. 5. Зміна витрати та тиску на виході КС

Висновки

Запропоновані регламенти робіт об'єктів та алгоритми їх реалізації були апробовані на багатьох прикладах. Результати числових експериментів продемонстрували, що запропоновані алгоритми забезпечують стійкість методів розрахунку параметрів газопотоків.

1. Селезнев В. Е., Алешин В. В., Клишин Г. С. Методы и технологии численного моделирования газопроводных систем. – М.: Едиториал УРСС, 2002. – 448 с. 2. Притула М., Лопух Н., П'янило Я., Савула Я. Алгоритми розрахунку параметрів течії газу в трубопроводах (2) // Вісник Нац. ун-ту "Львівська політехніка": Комп'ютерні науки та інформаційні технології. – Львів, 2009. – № 616. – С. 159–165. 3. Панкратов В.С., Дубинський Ф.В., Сиперштейн Б.И. Информационно-вычислительные системы. – Л.: Недра, 1988. – 246 с. 4. Сарданаивили С.А. Расчетные методы и алгоритмы (трубопроводный транспорт газа). – М.: ФГУП Изд-во РГУ нефти и газа им. И. М. Губкина, 2005. – 577 с.

С. Стрямець¹, О. Стрямець²

Національний університет “Львівська політехніка”,

¹кафедра автоматизованих систем управління²кафедра інформаційних систем та мереж

ГЕОІНФОРМАЦІЙНИЙ ПІДХІД ДО ОЦІНЮВАННЯ ЕМІСІЇ ТА ПОГЛИНАНЬ ПАРНИКОВИХ ГАЗІВ В ЛІСОВОМУ ГОСПОДАРСТВІ

© Стрямець С., Стрямець О., 2012

Проаналізовано цифрові карти лісів Львівської області (Україна) та Підкарпатського воєводства (Польща). Наведено порівняльний аналіз лісівничо-таксаційних показників лісів Львівської області (Україна) та Підкарпатського воєводства (Польща), які використовуються для обчислення емісій та поглинань парникових газів.

Ключові слова: геоінформаційні системи, цифрові карти, емісії та поглинання парникових газів

The digital maps of forest Lviv region (Ukraine) and the Subcarpathian Voivodeship (Poland) was analysed. Comparative analysis of forest taxation data in those study areas, which were used to calculate the emissions and absorption of greenhouse gases, was done.

Key words: GIS, digital maps, emission and absorption of greenhouse gases

Вступ

Україна приєдналася до міжнародного процесу запобігання змін клімату та зниження частки викидів парникових газів, підписавши Рамкову конвенцію ООН з питань змін клімату, яка має за кінцеву мету домогтися стабілізації концентрації парникових газів в атмосфері на такому рівні, який не допускав би небезпечного антропогенного впливу на кліматичну систему [19]. Функцію потужних поглиначів оксидів вуглецю з атмосфери виконують лісові екосистеми. Для оцінювання складних процесів поглинання і емісії парникових газів лісовими екосистемами традиційно використовується методика, розроблена Міжурядовою групою експертів зі змін клімату (Intergovernmental Panel on Climate Change – IPCC). Але передбачені в цій методиці методи обчислення емісій та поглинань парникових газів є доволі загальними, універсальними для різних регіонів світу і в остаточному підсумку вони не враховують особливостей цих регіонів [1]. Пропоновані цією методикою коефіцієнти для розрахунку емісій та поглинань коливаються іноді в доволі широкому діапазоні [2]. Тому завданням є їх уточнення та застосування з урахуванням специфіки конкретної аналізованої місцевості. Мета роботи – порівняльний аналіз лісівничо-таксаційних показників лісів Львівської області (Україна) та Підкарпатського воєводства (Польща), які використовуються для обчислення емісій та поглинань парникових газів.

Методика та методи

У керівних принципах національних інвентаризацій парникових газів запропоновано методи для оцінки викидів і поглинання парникових газів у зв'язку із змінами в біомасі, мертвій органічній речовині і органічному вуглеці ґрунтів на лісових площах і землях, відведених під лісові площі. Для оцінки джерел та поглиначів парникових газів враховуються всі ліси, що піддаються втручанню людини (зокрема природоохоронного призначення), зокрема всі етапи лісовирощування та лісокористування, виробництва комерційної деревини та некомерційної заготівлі паливної деревини, інші залишені без використання лісові землі (зміни в землекористуванні за останні 20 років). За

рекомендаціями IPCC для інвентаризації застосовують метод, який базується на визначенні нагромадження та втрат біомаси. Процес нагромадження об'єднує щорічний приріст надземної і підземної частин деревостану, процес втрати біомаси – це вилучення комерційної деревини (круглі лісоматеріали), заготівля дров, також враховується кількість деревини, пошкодженої шкідниками та знищеної пожежами [1].

Труднощі полягають у тому, що статистичні звіти з лісового господарства наводять дані про породний склад, вік, бонітет, запас насадження, де враховано лише об'єм товарної деревини, а для розрахунків поглинання вуглецю потрібні вхідні дані про кількість біомаси на одиниці площі, яка включає в лісових екосистемах підлісок, підріст, надгрунтовий покрив, всі ці яруси мають ще й підземну частину, яка теж акумулює вуглець. Крім того, існують інші неточності та незбіги у термінології, наприклад, в Україні діловою деревиною вважають деревину, яку використовують для будівництва, виготовлення меблів тощо і в звітах відокремлені дані про дров'яну деревину, яку використовують для опалення, у звітах Польщі під діловою деревиною розуміють всю товарну деревину.

Поглинання CO₂ відбувається через збільшення його запасу у фітомасі лісів при зміні лісокористування, залісненні та відновленні лісів. При веденні лісового господарства джерелами викидів CO₂ є:

- природний розпад відходів наземної фітомаси;
- спалювання відходів під час заготівлі лісу;
- використання деревини як палива;
- лісові пожежі і пошкодження деревини внаслідок епіфітотій і епізоотій.

Щоб обчислити поглинання вуглекислого газу, оцінюються дані про площу лісів, щорічний приріст фітомаси в лісах, на плантаціях та інших значущих резервуарах біомаси, а також дані про заготівлю деревини (дрова, комерційна деревина тощо).

Результати досліджень та їх обговорення

Для створення ГІС, призначеної для формування кадастру лісів Польщі, використані цифрові карти рослинності Польщі [3], в яких додатково враховані межі воєводств (адміністративна одиниця типу області) та дані про склад насаджень, вікову структуру, запас тощо [4]. У результаті отримана цифрова карта рослинності об'єднує 13518 елементарних ділянок хвойних лісів, 11043 ділянок листяних, 13926 ділянок мішаних лісів, які ростуть в 16 воєводствах Польщі (рис.1). Головними лісотвірними породами на рівнинних територіях є сосна, дуб, граб, береза, осика, для гірських лісів – ялиця, ялина, бук (рис. 2.). У Підкарпатському воєводстві виділено 1638 елементарних ділянок, частина лісів розташована в горах, частина – на рівнині. У Львівській області виділено 687 елементарних ділянок.

Використовуючи методику IPCC, визначено кількість біомаси на окремих елементарних ділянках, а потім – на території воєводств за допомогою відповідних формул.

Дані про господарську діяльність на n-й ділянці позначаємо як Δx_{nsm} з відповідним індексом, а результати інвентаризації, загалом, або в окремо взятому секторі для цієї ділянки, як відповідно ΔY_n і Δy_{ns} . У цьому випадку модель інвентаризації записуємо у вигляді

$$\Delta Y_n = \sum_{s=1}^S \Delta y_{ns} = \sum_{s=1}^S \sum_{m=1}^{M_s} a_{nsm} \Delta x_{nsm}, \quad n = 1, \dots, N, \quad (1)$$

де a_{nsm} – коефіцієнт для обчислення емісій від m-ї категорії господарської діяльності в s-му секторі n-го елементарної ділянки, N – загальне число елементарних ділянок. У моделі інвентаризації найнижчого рівня (рівня елементарних ділянок, якими можуть виступати як цілісні ділянки лісу, так і елементи сітки певного кроку) вхідні і вихідні дані представляють у вигляді георозподіленої бази даних. Підсумовування результатів інвентаризації по всіх елементарних об'єктах дає результати традиційної інвентаризації в межах країни або регіону.

Як приклад розглянуто лісові площі двох сусідніх областей Львівської в Україні і Підкарпатського воєводства Республіки Польщі. По цій території тягнуться суцільні лісові масиви Карпатського регіону і Розточчя. Лісовий фонд Львівської області характеризується високими таксаційними показниками. Зокрема, насадження II-го і вище бонітетів зростають на 95,8 % площі, зокрема високобонітетні деревостани займають 20,2 %, середньобонітетні – 75,6 %, низькобонітетні (IV-V) деревостани становлять лише 4,2 % вкритих лісом земель.

Вікова структура лісів області є значно кращою, ніж в середньому по Україні і близькою до оптимальних норм. Питома вага молодняків становить 23 % від загальної площі, вкритої лісовою рослинністю, що відповідає нормі, середньовікових насаджень – 49,5 % (при оптимальній нормі 30 %), пристигаючих – 16,8 % (норма 25 %); стиглих і перестійних – 10,7 % (при оптимальній нормі 22 %). Тобто спостерігається переважання середньовікових насаджень і до того ж площі пристигаючих, стиглих і перестійних лісів становлять дещо менший відсоток, ніж це передбачено за оптимальними нормами.

Загальний запас лісів Львівської області становить 153,6 млн. м³ або 252 м³ на 1 га вкритої лісовою рослинністю площі, що значно вище ніж в середньому по Україні, проте менше ніж в Закарпатській, Івано-Франківській та Чернівецькій областях. Із загального запасу лісів на стиглі і перестійні деревостани припадає 20,5 млн. м³ (або 13 %) і понад 6 млн. м³ (або понад 46 %) припадає на експлуатаційні ліси. Загальний середній приріст деревини щороку становить більше 2,5 млн. м³.

Поверхня, яка вкрита лісом у Республіці Польща, становить близько 9089 тис. га, лісистість становить 29,1 %, лісистість Підкарпатського воєводства – 37,2 %, що становить одну з найбільших лісистостей Польщі. Більшу лісистість має тільки Любуське воєводство (48,9 %). Площа, вкрита лісом в Підкарпатському воєводстві, становить 663797 га, запас ділової деревини становить 194,675 млн. м³, або 293,3 м³/га. За віковою структурою в Підкарпатському воєводстві переважаючими є середньовікові ліси III і IV класів віку відповідно 25 і 21 % (рис. 2).

Переважаючими породами в Підкарпатському воєводстві є сосна 35 %, бук 21 %, ялиця 16 % (рис. 3).

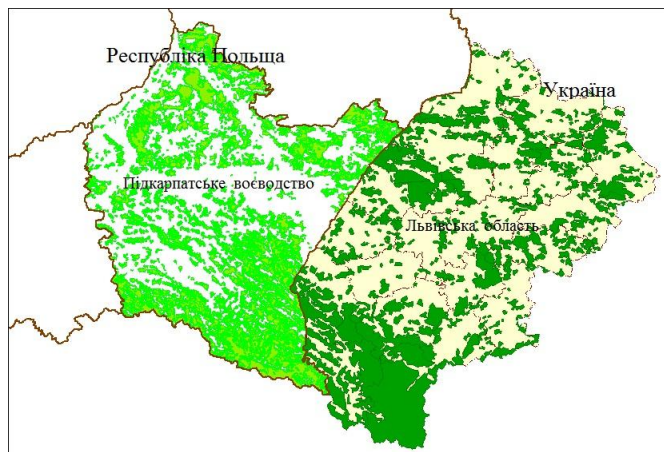


Рис. 1. Карта лісів Підкарпатського воєводства (Республіки Польща) і Львівської області (Україна)

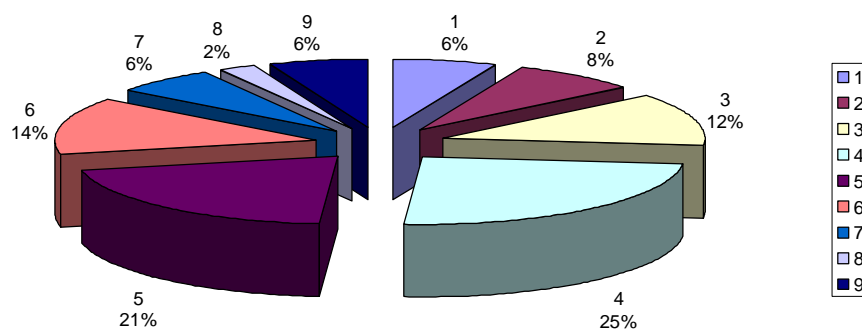


Рис. 2. Розподіл площі лісів Підкарпатського воєводства за класами віку:

- 1 – незаліснені землі; 2 – I класу віку 1 – 20 pp.; 3 – II клас віку 21 – 40 pp.; 4 – III клас віку 41–60 pp.; 5 – IV клас віку 61–80 pp.; 6 – V клас віку 81–100 pp.; 7 – VI клас віку 101–120 pp.; 8 – VII клас віку старше 120 p; 9 – ліси КО, КДО, ВР

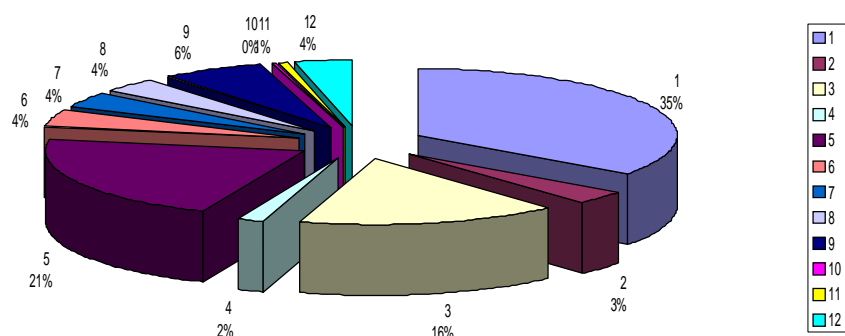


Рис. 3. Породний склад лісів Підкарпатського воєводства :
 1 – сосна; 2 – ялина (смерека); 3 – ялиця; 4 – інші хвойні; 5 – бук; 6 – дуб; 7 – граб;
 8 – береза; 9 – вільха; 10 – тополя; 11 – осика; 12 – інші листяні

Висновки

Можливості геоінформаційних систем дозволяють будувати цифрові карти лісів за різними лісівничо-таксаційними характеристиками, обчислювати загальний запас надземної та підземної частин деревостанів залежно від породного складу, вікової структури тощо, а в остаточному результаті оцінювати кількість депонованого вуглецю та емісії парникових газів під час ведення лісового господарства. За результатами досліджень найбільше нагромаджено вуглецю в Підкарпатському воєводстві та в Львівській області в гірських лісових екосистемах (Карпати) та на Розточчі.

Збереження вуглецевого запасу в лісових насадженнях може бути досягнуто за поліпшення методів ведення лісового господарства, що дасть можливість залучити додаткові кошти від продажу квот на викиди парникових газів.

1. IPCC 2006, 2006 IPCC Guidelines for National Greenhouse Gas Inventories, Prepared by the National Greenhouse Gas Inventories Programme, Eggleston HS, Buendia L., Miwa K., Ngara T. and Tanabe K. (Eds). Published: IGES, Japan. 2. The IPCC software for estimating greenhouse gas emissions. IPCC Version 1.1, 1998.-[http:// www.ipcc-nggip.iges.or.jp/public/gl/software.htm](http://www.ipcc-nggip.iges.or.jp/public/gl/software.htm). 3. Руководящие принципы национальных инвентаризаций парниковых газов, МГЭИК, 2006, Глава 4: Лесные площади. 4. Biuro Urządzenia Lasu i Geodezji Leśnej WIELKOBSZAROWA INWENTARYZACJA STANU LASÓW W POLSCE WYNIKI ZA OKRES 2006-2010 ETAP 2.2.1.b (praca wykonana na zamówienie Dyrekcji Generalnej Lasów Państwowych zgodnie z umową nr OP/2715-4/U/10 z dnia 22 lipca 2010 r.), Sękocin Stary, marzec 2011 r.

О. Різник, Б. Дубик, Н. Куліш, Т. Чабан
Національний університет “Львівська політехніка”,
кафедра інформаційних технологій видавничої справи

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ СТРУКТУР СИСТЕМ НА ОСНОВІ КОМБІНАТОРНИХ МЕТОДІВ

© Різник О., Дубик Б., Куліш Н., Чабан Т., 2012

Розглянуто інформаційні технології оптимізації структур технічних систем за числових в'язанок як зручних математичних моделей з нееквідистантною структурою на прикладі розробки моделей баркероподібних та псевдовипадкових послідовностей, розробки оптимальних схем відновлення для розподілених обчислень, розробки ефективних фільтрів зображень, розробки моделей розташування датчиків сенсорної мережі.

Ключові слова: ідеальна кільцева в'язанка, фільтрація зображення, розподілені обчислення, схема відновлення, сенсорна мережа.

In the article information technologies of optimization of structures of the technical systems are examined at numerical bundles as comfortable mathematical models with a non-uniform structure on the example of development of models of quasibarker and pseudocausal sequences, development of optimal charts of renewal for the up-diffused calculations, development of effective filters of images, development of models of location of sensors of sensory network.

Key words: idea ring bundle, filtration of image, up-diffused calculations, recovery scheme, sensory network.

Вступ

Сучасний розвиток інформаційної техніки значною мірою пов'язаний з дослідженням методів структурної оптимізації технічних систем. Одним з перспективних методів дослідження моделей систем, які використовуються в технічній кібернетиці, є побудова абстрактних нееквідистантних моделей дискретних систем із застосуванням апарата комбінаторного аналізу. Однак недостатнє вивчення наявних моделей та алгоритмів швидкого синтезу комбінаторних конфігурацій з нееквідистантною структурою ускладнює розробку ефективних методів синтезу технічних систем на основі такого підходу.

Тому вагомим значення набувають питання розробки інформаційних технологій оптимізації структур систем та алгоритмів ефективного синтезу комбінаторних конфігурацій з нееквідистантною структурою як моделей технічних систем, які дають змогу знаходити оптимальні рішення без перегляду великої кількості варіантів.

Постановка проблеми

Дослідження і синтез комбінаторних конфігурацій з нееквідистантною структурою як моделей для оптимізації структур систем, які наділені корисними для практичних цілей властивостями, що дозволяють значно поліпшувати основні показники технічних систем за надійністю, точністю, стабільністю характеристик, діапазоном роботи, функціональними можливостями, становить актуальну науково-технічну задачу.

Один з таких підходів полягає в створенні математичної моделі, яка б за своїми властивостями відповідала широкому класу технічних систем з різноманітними параметрами; щоб ця модель дозволяла здійснювати багатогранні дослідження над різними типами технічних систем, спрощувала методикку їх оптимізації та відкривала нові можливості для практичного застосування.

Розв'язання задачі

Практична реалізація задачі оптимізації структур систем у технічних системах та пристроях дає змогу розглядати в загальному випадку будь-які їх характеристики або параметри, які піддаються кількісній оцінці. Залежно від способу реалізації структур систем з нееквідистантною структурою можна виділити три основні підходи до оптимізації структур систем у технічних системах: геометричний, параметричний, змішаного типу. В основу вищезгаданих підходів покладено поняття комбінаторної конфігурації у вигляді числової в'язанки.

Числовою в'язанкою за деякою операцією називається така впорядкована послідовність скінченної кількості N елементів, де всі значення елементів та результати виконання операцій над послідовно розміщеними елементами утворюють нову множину взаємозв'язаних значень, які мають R повторюваних ознак [2]. Будь-яку числову в'язанку можна розглядати не лише як упорядковану послідовність елементів базової множини, але й як сукупність одержаних на ній результатів.

Для тих конструкцій, у яких елементами є числа або їх комбінації, постає можливість інтерпретації числової в'язанки у вигляді сукупності певним способом розподілених точок деякої геометричної фігури, що відповідають координатам просторового розміщення елементів результуючої множини. Це можуть бути точки на прямій, площині, об'ємній фігурі або поверхні багатовимірного простору.

Застосування для оптимізації числових в'язанок у технічних системах першого типу пов'язано з геометричними особливостями систем, зокрема, взаємним розміщенням окремих елементів на деякій ділянці виділеного простору. У системах другого типу завданням є вибір співвідношення значень параметрів, якими характеризуються властивості систем або процесів. Для третього типу систем характерні ознаки систем обох типів.

У технічних об'єктах часто поєднуються ознаки моделей числових в'язанок різних типів, наприклад, геометричних і параметричних систем, в яких виникає задача вибору, вигідного для практичної реалізації співвідношення параметрів, з'єднаних між собою в просторі елементів схеми за пропорціями числових в'язанок. Можна також конструктивні особливості технічних об'єктів встановлювати відповідно до значень параметрів. Отже, в геометрично-параметричних системах для забезпечення можливості реалізації відповідних значень параметрів необхідно розраховувати геометричні співвідношення розподілу елементів структури згідно з числовими в'язанками.

Розглянемо декілька практичних застосувань числових в'язанок для оптимізації структур технічних систем:

- розробка моделей баркероподібних послідовностей;

Методів кодування сьогодні розроблено достатньо багато, причому для передачі даних в умовах перешкод з успіхом використовується технологія розширення спектра. Основна ідея полягає в тому, щоб від вузькосмугового спектра сигналу, що виникає при звичайному потенційному кодуванні, перейти до широкосмугового спектра. Саме це дозволяє значно підвищити перешкодостійкість переданих даних.

Для того, щоб підвищити перешкодостійкість переданої кодової послідовності, тобто, щоб збільшити вірогідність безпомилкового розпізнавання кодової послідовності на приймальній стороні в умовах завад, можна скористатися методом переходу до надлишкової кодової послідовності, додаючи надмірність у початкову кодову послідовність. Для цього в кожен переданий інформаційний біт вбудовують певний код, що складається з послідовності так званих чіпів.

Однією з найвідоміших таких послідовностей є послідовність Баркера, яка має якнайкращими серед відомих послідовностей властивостями шумоподібності, що і зумовило їх широке застосування. Баркероподібна кодова послідовність – це спеціальний набір нулів і одиниць, за допомогою якого кодують один біт (чіп) для подальшої передачі. Основними властивостями кодових послідовностей є дуже хороші автокореляційні і взаємкореляційні властивості. При однакових довжинах властивості послідовностей можуть кардинально відрізнятися.

Методика побудови на основі числових в'язанок за критерієм мінімального значення функції автокореляції баркероподібних послідовностей полягає в побудові L_N -позиційного коду (L_N – сума

елементів числової в'язанки) m_i , $i=1,2,\dots,L_N$ з на базі вибраного варіанта числової в'язанки $(k_1, k_2, \dots, k_l, \dots, k_N)$ з параметрами $N/R \approx 2$, де на N позиціях коду з порядковими номерами x_l , $l=1,2,\dots,N$, які визначаються з формули $x_l \equiv 1 + \sum_{i=1}^l k_i \pmod{L_N}$, розміщуються "1", а на решті $L_N - N$ позиціях – "-1".

Одержана кодова послідовність є послідовністю з властивістю "не більше R -збігів" та мінімальним значенням автокореляційної функції. Вибираючи інший варіант числової в'язанки з такими параметрами, можемо одержати інші баркероподібні кодові послідовності з властивістю "не більше R -збігів" та мінімальним значенням автокореляційної функції [2].

- розробка моделей псевдовипадкових послідовностей;

За постулатами Голомба псевдовипадкові послідовності повинні мати такі властивості:

- у кожному періоді послідовності число 1 і 0 відрізняється не більше ніж на одиницю;
- серед груп з послідовних 1 і 0 в кожному періоді половина має тривалість в один символ, четверта частина має тривалість у два символи, восьма частина має тривалість у чотири символи тощо.
- кореляційна функція послідовності має єдиний значний пік амплітуди 1 і при всіх зрушеннях дорівнює $1/m$ (m – довжина послідовності).
- кореляція між векторами обчислюється за формулою $r(c, y) = \frac{A-B}{A+B}$, де A – число позицій, в яких символи послідовностей x і y збігаються; B – число позицій, в яких символи послідовностей x і y є різними.

Для побудови псевдовипадкових послідовностей за допомогою числових в'язанок порядку N кратності R виділимо рядок із L_N пронумерованих у зростаючому порядку клітинок одновимірного масиву і заповнимо інформаційними "одиницями" клітинки, номери яких збігаються з числами, визначеними з числової в'язанки [2]. У клітинки, що залишилися незаповненими, занесемо "нулі". Утворена послідовність одиниць і нулів є L_N -розрядною псевдовипадковою послідовністю, циклічним зсувом якої можна одержати й решту псевдовипадкових послідовностей за постулатами Голомба.

- розробка оптимальних схем відновлення для розподілених обчислень;

Знайти хороші схеми відновлення – фундаментальна задача в розподілених і кластерних системах. Такі схеми визначають як робоче навантаження треба перерозподілити, коли один або більше комп'ютерів виходять з ладу. Ми припускаємо, що робота, яка виконується кожним з n комп'ютерів, має бути переміщена, як одна атомна одиниця. Приклади – системи, де уся робота, яку виконує комп'ютер, робиться від однієї зовнішньої системи або коли усю роботу виконує один процес чи системи, де зовнішньою комунікацією управляє IP протокол. Мета – розподілити навантаження порівну, навіть за найнесприятливіших комбінацій комп'ютерів, що вийшли з ладу, тобто оптимізувати поведінку за найгіршого випадку. Розглянуто n ідентичних комп'ютерів, які за нормальних умов виконують один процес. Усі процеси виконують ту ж кількість роботи. Схемою відновлення, що гарантує оптимальне поширення навантаження в найгіршому випадку, коли x комп'ютерів вийшли з ладу, є так звана оптимальна схема відновлення для значення n і x на основі числових в'язанок [3].

Схема відновлень отримується додаванням значень елементів числової в'язанки – це послідовність з часткових сум. Перша частина списків відновлень складається з числової в'язанки, тобто перші x входи для найбільшого x такі, що сума оптимальної послідовності довжини x менша, ніж сума числової в'язанки L_N . Частина списку відновлень, що залишилася, наповнена залишком номерів (комп'ютерів) аж до $L_N - 1$.

- розробка ефективних фільтрів зображень;

У галузях техніки, науки, медицини основна інформація про об'єкти досліджень надходить у вигляді зображень – двовимірних проєкцій просторових сцен. Для обробки і аналізу такої

інформації необхідно забезпечити високу візуальну якість зображення, яка втрачається через незадовільні умови її отримання, недосконалості систем відтворення чи систем передавання відеоінформації, появу різного роду завад і спотворень. Тому для успішної обробки растрових зображень необхідно мати методи фільтрації, на основі яких можливо розробити якісно швидкі фільтри початкової обробки зображень. Недолік наявних прямокутних фільтрів полягає у можливій появі помилкового зображення, коли в зображенні є високі просторові частоти. Також недоліком лінійної фільтрації зображень є те, що поряд зі зменшенням шумів одночасно відбувається розмивання контурів зображення. Це зумовлено тим, що всі елементи вихідного зображення обробляються з однаковим коефіцієнтом, тобто лінійні фільтри незалежні від структури елементів і тому не можна визначити межу між шумовими і контурними елементами [1].

У результаті проведених досліджень, використовуючи числові в'язанки на стандартному алгоритмі просторової фільтрації, отримано нові різновиди матриць перетворення. А саме матриці фільтрів розмиття, підвищення різкості, виділення контурів зображення та тиснення. Коефіцієнтами матриць перетворення були взяті елементи числової в'язанки $(k_1, k_2, \dots, k_l, \dots, k_N)$. Після порівняння зображень, опрацьованих фільтром розмиття, використовуючи кожен з цих матриць, можна зазначити, що зі збільшенням порядку матриці зростає ефект фільтра. Використовуючи числові в'язанки, як матриці перетворення для фільтрації зображень, можемо отримати фільтри розмиття, різкості, тиснення та виділення контурів, причому з покращеними характеристиками порівняно до стандартних фільтрів. Досліджено використання методів просторової обробки зображень та розроблений метод фільтрації зображення з використанням числових в'язанок, що дозволяє використовувати його як один з складових загальної оцінки візуальної якості зображення, що не залежить від суб'єктивного сприйняття.

- розробка моделей розташування датчиків сенсорної мережі.

Для моделей розташування датчиків сенсорної мережі з рівномірним (еквідистантним) розподілом елементів (всі відстані/вектори між сусідніми елементами однакові) характерна наявність повторюваних ознак – рівновіддалених пар елементів, що обмежує інформаційні можливості такої моделі з точки зору зняття досліджуваних параметрів середовища. Суть побудови моделей з нерівномірним (нееквідистантним) розподілом елементів полягає в мінімізації повторюваних ознак. Отже, ці моделі доцільно представляти у вигляді з'єднаних між собою елементів, у яких значення векторів відстаней між ними є різними. Якщо така модель має мінімальну апертуру, то вона описується відповідно дво- чи тривимірною оптимальною числовою в'язанкою. Подібні моделі застосовують й в галузі радіоелектроніки.

Сенсорну мережу можна охарактеризувати векторами розташування датчиків та їх кількістю, на підставі чого може бути сформований масив досліджуваних значень. Просторову функцію кореляції можна оцінити на основі цього масиву. Якщо масив багатовимірний, то він має бути представлений набором векторів. Якщо масив лінійний, то він представлений набором чисел. Якщо масив має більше ніж одну пару датчиків, відокремлених тією ж відстанню/вектором, ці пари роблять надмірні оцінки кореляції. У такому випадку цей масив має надмірності. Якщо немає ніякої пари датчиків, відокремлених деякою відстанню/вектором, яка менша, ніж апертура масиву, цей масив має діру в цьому розташуванні. Оптимальний масив надмірності або масив мінімальної надмірності – єдиний, який не має ніяких більше надмірностей, ніж будь-який інший масив надмірності з тою самою кількістю датчиків [4]. За однакових відстаней між парами елементів датчиків сенсорної мережі інформація є надлишковою, що означає її втрату. Найвища точність обчислень досягається, коли кожна пара датчиків має різну відстань. Ця умова задовольняється саме за розташування датчиків за багатовимірними числовими в'язанками.

Багатовимірні комбінаторні конфігурації типу числових в'язанок як моделей розташування датчиків сенсорної мережі доцільно представляти у вигляді координат точок t -вимірного простору ($t=2$ для двовимірного розташування, $t=3$ для тривимірного розташування), де необхідно знайти координати заданого числа N точок на апертурі деякого t -вимірного паралелепіпеда. Різниці всіх пар координат повинні бути різними і зустрічатися не більше одного разу ($R=1$) на множині всіх

нає пар координат за умови мінімальних розмірів t -вимірного паралелепіпеда. Досліджено методи синтезу багатовимірних числових в'язанок на основі одновимірних числових в'язанок з параметрами (L_N^R, N, R) , значення векторів відстаней яких не можуть повторюватись більше ніж R разів, що, своєю чергою, дає змогу реалізувати числову послідовність у вигляді t -вимірної числової в'язанки з параметрами $l_1 \times l_2 \times \dots \times l_t \leq L_N^R$. У загальному випадку різним варіантам багатовимірних ЧЛВ можуть відповідати будь-які одновимірні ЧЛВ. Однак найбільший інтерес для розташування датчиків у сенсорних мережах становлять дослідження компактних багатовимірних числових в'язанок, тобто, таких числових в'язанок, які мають максимальну кількість елементів у просторі $l_1 \times l_2 \times \dots \times l_t \leq L_N^R$.

Висновки

Результати моделювання та дослідження комбінаторних конфігурацій за допомогою числових в'язанок дають підстави стверджувати про широкі можливості використання числових в'язанок у новітніх інформаційних технологіях для ефективного проектування та оптимізації технічних пристроїв і систем з нееквідистантною структурою у різних галузях науки і техніки.

1. Гонсалес Р. *Цифровая обработка изображений* / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с. 2. Пізник В.В. *Синтез оптимальних комбінаторних систем*. – Львів: Вища шк., 1989. – 168 с. 3. Pfister G. F., *In Search of Clusters*, Prentice–Hall, 1998. 4. D. Pearson, S. Pillai, and Y. Lee, “An algorithm for near-optimal replacement of sensor elements,” *IEEE Trans. Inform. Theory*, vol. 36, pp. 1280–1284, Nov. 1990.

УДК 519.216:612.3

М. Тимчак, М. Хвостівський, Л. Дедів

Тернопільський національний технічний університет імені Івана Пулюя

МАТЕМАТИЧНА МОДЕЛЬ ЕЛЕКТРОГАСТРОЕНТЕРОСИГНАЛУ ДЛЯ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ЕЛЕКТРОГАСТРОЕНТЕРОСИСТЕМ

© Тимчак М., Хвостівський М., Дедів Л., 2012

Проаналізовано останні дослідження шлунково-кишкового тракту людини за електрогастроентеросигналом та обґрунтовано адекватну його модель.

Ключові слова: електрогастроентеросигнал, шлунково-кишковий тракт, періодично-корельований випадковий процес

Recent research analysis of the gastrointestinal tract of human have been made by electrogastroenterosignal and its adequate model is graunded.

Key words: electrogastroenterosignal, gastrointestinal tract, periodically correlated random process

Вступ

Сьогодні у медичній практиці після оперативних втручань в органи шлунково-кишкового тракту (ШКТ) швидкість відновлення моторики контролюється за допомогою методів, які можна розділити на дві групи [1–4]:

- методи, що дають змогу безпосередньо реєструвати скоротливу активність ШКТ;

- методи оцінки моторної функції органів на основі даних, що характеризують їх електричну активність.

До першої групи належать методи, які базуються на безпосередньому вимірюванні внутрішньо-просвітнього тиску ШКТ за допомогою балонів, мікросенсорів, радіокапсул, катетерів. Їх особливістю є введення чужорідного тіла безпосередньо в просвіт органа, що призводить до роздратування механорецепторів слизової оболонки і змінює її моторну активність. Саме тому сьогодні особливої актуальності набувають неінвазивні і нетравмуючі методи діагностики ШКТ.

До другої групи належать електрофізіологічні методи, які базуються на вивченні електричної активності ШКТ, і дають змогу дослідити тісні взаємозв'язки між електричною і скоротливою діяльністю ШКТ [5–10]. Електрофізіологічні методи об'єднують як безпосередню реєстрацію біопотенціалів гладком'язових стінок органів з фіксованих на них електродів – пряма електрогастроентерографія, так і їх реєстрацію з на шкірних електродів – периферична електрогастроентерографія [11–14]. Необхідність імплантувати електроди в стінку органу обмежує використання прямої електрогастроентерографії в клінічній практиці. До того ж, у деяких роботах [11–14] виявлений достовірний зв'язок між результатами прямої і периферичної електрогастроентерографії.

Периферична електрогастроентерографія як неінвазивний метод дослідження ШКТ дає змогу оцінити біоелектричну активність шлунку, дванадцятипалої кишки і інших відділів ШКТ і базується на реєстрації змін електричного потенціалу від органів ШКТ (шлунок, дванадцятипала кишка, кишечник, пряма та товста кишки). Застосування периферичної електрогастроентерографії в процесі діагностики стану ШКТ уможливорює своєчасне (раннє) виявлення змін у функціонуванні органів травлення, що забезпечує подальший контроль за лікуванням та визначення доцільності оперативних втручань.

Ефективність діагностики ШКТ людини методом електрогастроентерографії залежить від виду математичної моделі електрогастроентеросигналу, яка і визначає методи його опрацювання, які дають змогу визначати інформативні параметри сигналу.

Сьогодні усі відомі методи опрацювання електрогастроентеросигналу у комп'ютерних електрогастроентерографах, такі як морфологічний, Вейвлет- та Фур'є-аналізи, базуються на детермінованій та стаціонарній моделі, що є неадекватним для цього типу сигналу. Оскільки електрогастроентеросигнал за природою походження є випадковим та періодичним, тому адекватна математична модель повинна враховувати у своїй структурі поєднання властивостей періодичності із випадковістю, що є важливим під час дослідження фазово-часових змін у сигналі із метою своєчасного (раннього) виявлення змін у функціонуванні ШКТ.

Аналіз характеристик електрогастроентеросигналу

Проаналізовано характеристики ЕГЕС (рис. 1) з позицій детермінованого та стаціонарного підходів.

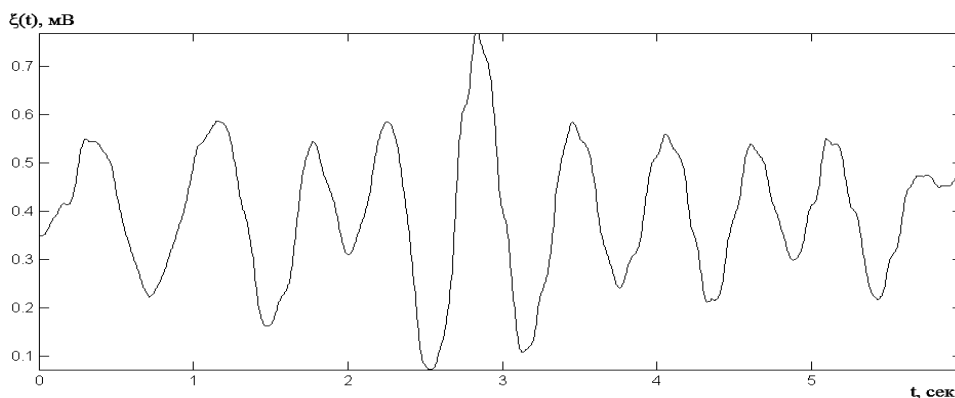


Рис. 1. ЕГЕС з бази даних системи "Гастроскан-ГЭМ" (норма)

Результати аналізу ЕГЕС методами гармонічного аналізу у межах детермінованого підходу підтверджують, що отримані амплітудні спектри відгуків ЕГЕС (рис. 2) є мінливими, що свідчить про наявність в сигналі стохастичної складової.

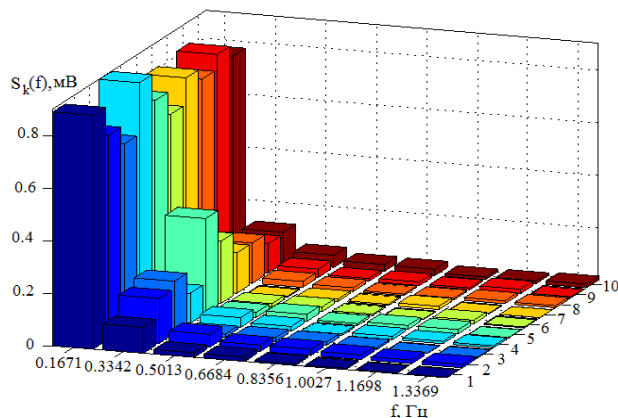


Рис. 2. Реалізації амплітудних спектрів ЕГЕС на k -х періодах

Розглядаючи сигнал у межах стаціонарної моделі, помічено, що кореляційна функція від ЕГЕС є періодичною в часі t та циклічно-загасаючою по зсуву u (рис. 3).

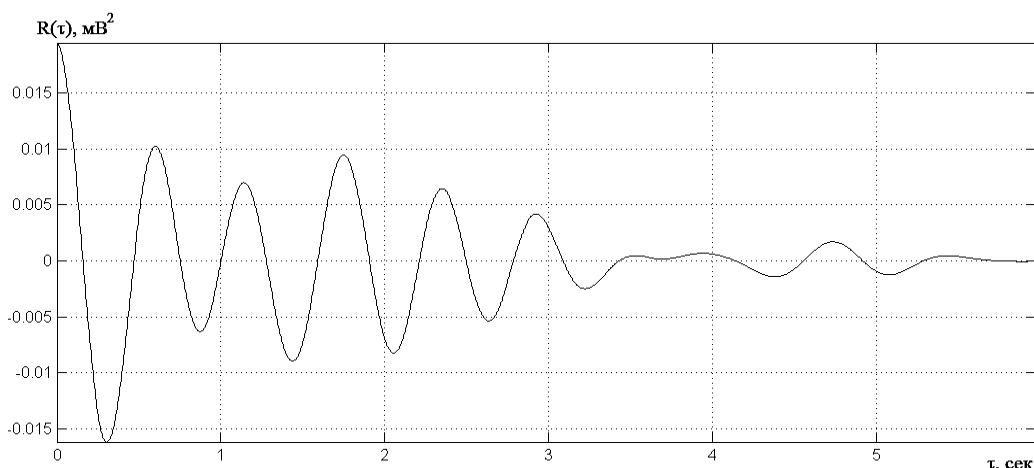


Рис. 3. Реалізація автокореляції ЕГЕС

Адекватна модель ЕГЕС повинна мати властивість стохастичності, гармонізованості і періодичності її статистичних характеристик. Випадкова стаціонарна модель відображає складність ЕГЕС у спектральному розподілі потужності, проте не відображає його фазово-часової структури, яка є важливим показником у разі виявлення фазово-часових змін у сигналі. У термінах енергетичної теорії ці вимоги задовольняє модель у вигляді періодично корельованого випадкового процесу, яка має засоби врахування як пов'язаності гармонічних складових, так і зміни ймовірнісних характеристик.

Математична модель електрогастроентеросигналу у вигляді періодично-корельованого випадкового процесу

Оскільки потужність ЕГЕС є скінченною в межах періоду, тоді можемо зарахувати сигнал до класу p^T [15].

Враховуючи наведене, можемо припустити, що моделлю в цьому разі є принаймні нестационарний випадковий процес скінченної середньої потужності (класу p^T). Тоді енергетична теорія стохастичних сигналів (ЕТСС) обґрунтовує алгоритм обчислення оцінок характеристик стаціонарного наближення для випадкових процесів класу p , які дорівнюватимуть оцінкам такого стаціонарного

процесу, що складається з таких самих як і процес класу \mathbf{p} , гармонік і з такими самими потужностями, але вже некорельованих (бо стаціонарність і корельованість гармонік для процесів класу \mathbf{p} рівносильні) [15].

Енергетична теорія стохастичних сигналів обґрунтовує зображення такого типу сигналів, із законом збереження середньої потужності при цьому та вказанням типу їхньої корельованості у часовій області або у часі повторюваністю імовірнісних характеристик, зображення через стаціонарні компоненти і стаціонарні послідовності відліків [15].

Відповідно до ЕТСС з цього випливає, що адекватною моделлю ЕГЕС буде математична модель у вигляді загального періодично корельованого випадкового процесу класу \mathbf{p}^T , яка найзагальніше поєднує випадковість значень з повторністю, трактуючи її як періодичність імовірнісних характеристик. Тоді задача зведеться до того, щоб на підставі апріорного теоретичного аналізу структури цієї моделі виявити можливі інваріанти, що їх може надати опрацювання емпіричних даних, та обґрунтувати алгоритм цього опрацювання [15].

Періодично-корельовано-випадкові процеси (ПКВП) – це моделі стохастичних коливань з періодичною зміною імовірнісних характеристик. Вони творять підклас \mathbf{p}^T [15].

Періодично корельовано – це такі випадкові процеси, математичне сподівання і коваріація якого задовольняють умови [15]: існує таке певне $T > 0$, яке називається періодом корельованості, що

$$m_x(t+T) = m_x(t), r_x(t+T, s+T) = r_x(t, s), \forall t, s \in R, \quad (1)$$

де $m_x(t)$ – математичне сподівання, а $r_x(t, s)$ – кореляційна функція процесу, або, якщо використати параметричну коваріацію (функцію кореляції $b_x(t, u) = r_x(t+u, t)$), то з цих формул виходить, що математичне сподівання і параметрична коваріація є періодична з періодом T :

$$m_x(t+T) = m_x(t), b_x(t+T, s+T) = b_x(t, s), \forall t, s \in R. \quad (2)$$

Означення класу \mathbf{p} мало однією із причин вивчення ПКВП у спектральній області – їхньої гармонізованості, вигляду спектра (типу корельованості гармонічних складових), закону збереження (потужності в узагальненій теоремі Вінера-Хінчина) та зображення таких процесів через спектральні компоненти. При цьому очевидно, що умова (1) під час обчислення середніх характеристик призводить до того, що усереднення по всій осі переходить в усереднення по відрізьку довжини T , що (внаслідок інваріантності усереднення зсувів) можна вважати як усереднення на відрізьку $[0, T)$, тобто прийняти, що середні величини характеристики процесу даються виразами:

$$m_x = M_t \{m(t)\} = \frac{1}{T} \int_0^T m(t) dt, \quad (3)$$

$$B(u) = M_t \{r(t+u, t)\} = \frac{1}{T} \int_0^T r(t+u, t) dt = \frac{1}{T} \int_0^T b(t, u) dt, \quad (4)$$

де M_t – символ усереднення по всій осі; T – період корельованості ПКВП.

Ці характеристики тому мають розклади у ряди Фур'є:

$$m(t) = \sum_{k \in Z} m_k e^{ik\Lambda t}, \quad (5)$$

$$b(t, u) = \sum_{\tau \in Z} B_k(u) e^{ik\Lambda t}, \quad (6)$$

як слід розуміти у сенсі теорії узагальнених функцій Шварца [15], коли розглядати ПКВП скінченної середньої потужності. Тут $\Lambda = \frac{2\pi}{T}$. Справді, оскільки у випадку періодичної функції:

$$M_t \{f(t)\} = \lim_{\Theta \rightarrow \infty} \frac{1}{2\Theta} \int_{-\infty}^{\infty} f(t) dt = \lim_{\Theta \rightarrow \infty} \frac{1}{2NT + 2\Delta} \left[N \int_0^T + N \int_{-T}^{-\Theta} + N \int_{NT}^{\Theta} \right] f(t) dt = \frac{1}{T} \int_0^T f(t) dt, \quad (7)$$

де $N = E\left(\frac{\Theta}{T}\right)$, $E(\bullet)$ – ціла частина числа, $\Delta = \Theta - NT$, то звідси середня потужність ПКВП визначається усередненням на періоді корельованості, тобто на відрізку $[0, T)$, тоді:

$$P_x^T = \frac{1}{T} \int_0^T E|X(t)|^2 dt = \frac{1}{T} \int_0^T r(t, t) dt, \quad (8)$$

а умова належності до класу \mathbf{p} набуде вигляду

$$P_x^T = \frac{1}{T} \int_0^T r(t, t) dt < \infty, \quad (9)$$

Тому клас ПКВП, для якого виконується ця умова, коли його трактувати як підклас у класі \mathbf{p} , був названий класом \mathbf{p}^T . На цей клас поширюється з відповідними видозмінами теорія класу \mathbf{p} [15]. Зокрема замість простору \mathbf{h}^p вводиться простір $\mathbf{h}^T = L^2\left([0, T); \frac{1}{T}, K\right)$, тобто простір інтегровних на $[0, T)$ з квадратом за мірою $\frac{dt}{T}$ функцій над колгоморівським гільбертовим простором K випадкових величин скінченної дисперсії, і норма у цьому просторі $\|x\|_{\mathbf{h}^T} = \sqrt{P_x^T}$.

Кореляційні компоненти:

$$B_k(u) = \frac{1}{T} \int_0^T r(t+u, t) e^{-ik\Delta t} dt, \quad (10)$$

в силу їхньої обмеженості $B_k(u) \leq B_0(u) \leq B_0 = P_x^T$ належать до класу B^2 , тому мають зображення у вигляді Фур'є за мірами (загалом комплексно значними):

$$B_k(u) = \int_R e^{iul} F_k(dl), \quad (11)$$

де R – множина дійсних чисел, $F_k(dl)$ – спектральна міра.

Коваріація ПКВП має зображення у вигляді

$$r(t, s) = \iint_{R^2} e^{i(dl, dm)} F(dl, dm), \quad (12)$$

де $F(dl, dm)$ – спектральна біміра.

Встановлені властивості коваріації ПКВП дають підставу вивести вираз самого процесу через його стаціонарні складові, тобто встановити структуру цього класу процесів [15].

ПКВП належить до класу \mathbf{p}^T тоді і тільки тоді, коли він має зображення:

$$x(t) = \sum_{k \in Z} x_k(t) e^{ik\Delta t}, \quad (13)$$

де, $x_k(t)$ – стаціонарні компоненти ПКВП, Z – множина всіх цілих і відповідно підмножина додатних чисел, Λ – циклічна частота $\Lambda = 2p/T$.

На основі обгрунтованої математичної моделі можна реалізувати прості алгоритми опрацювання ЕГЕС засобами ЕТСС (синфазний і компонентний) для отримання статистичних оцінок його імовірнісних характеристик, які є показниками стану шлунково-кишкового тракту.

Висновки

З аналізу електрогастроентеросигналу та описаних властивостей періодично-корельованих випадкових процесів випливає, що математична модель процесу такого класу дає змогу адекватно описати сигнал, а саме врахувати поєднання випадковості та періодичності сигналу, а тому і розробити методи визначення інваріантних інформаційних ознак електрогастроентеросигналу

враховуючи статистику таких сигналів для задач ранньої (своєчасної) діагностики стану шлунково-кишкового тракту.

1. Зальцман И.Н. Комплексная методика электрогастрографического и рентгенологического исследования двигательной функции желудка: автореф. дис. ...канд.мед.наук / И.Н. Зальцман. – М., 1965. – 23 с. 2. Ногаллер А.М. Новое в электрогастрографии / А.М. Ногаллер. – М.: Тер. архив, 1971. – № 5. – С. 113–114. 3. Саблин О. А. Функциональная диагностика в гастроэнтерологии / О.А. Саблин, В.Б. Гриневич, Ю.П. Успенский, В.А. Ратников: Учеб.-метод. пособие. – М., СПб., 2002. — 88 с. 4. Ступин В.А., Богданов А.Е., Закиров Д.Б., Силуянов С.В., Смирнова Г.О. Новый метод диагностики моторно-эвакуаторной функции желудочно-кишечного тракта. 5. Климов П.К. Биоэлектрическая активность гладких мышц пищеварительного тракта и ее связь с сократительной деятельностью / П.К. Климов, В.Н. Устинов // Успехи физиологических наук. – М., 1973. – Т. 4. – № 4. – С. 3–33. 6. Ребров В.Г. Особенности регистрации электрической активности желудка и кишечника с поверхности тела пациента / В.Г. Ребров, Б.А. Станковский, Г.И. Куланина // Рос. журн. гастроэнтерологии, гепатологии, колопроктологии. – М., 1996. – № 2. – С. 48–52. 7. Ребров В.Г. Особенности регистрации электрической активности желудка и кишечника с поверхности тела пациента / В.Г. Ребров, Б.А. Станковский, Г.И. Куланина // Рос. журн. Гастроэнтерол. гепатол. колопроктол. – М., 1996. – Т. 6, № 2. – С. 48–52. 8. Смирнов А.А. Диагностическая оценка изменений электрической активности тонкой кишки у больных с распространенным перитонитом / А.А. Смирнов: Автореферат дис. ...канд. мед. наук, 14.00.27. – хирургия. ГИУВ. – Иркутск, 2006. 9. Устинов В.Н. Конфигурация биопотенциалов гладких мышц желудка и двенадцатиперстной кишки / В.Н. Устинов // Физиологический журнал СССР им. И.М. Сеченова.. – М., 1974. – Т. 60. – № 6. – С. 961–970. 10. Шуба М.Ф. Электрофизиологические особенности гладких мышц желудочно-кишечного тракта / М.Ф. Шуба: В кн. «Моторная функция желудочно-кишечного тракта». – К., 1965. – С. 155–166. 11. Пономарева А.П. Периферическая электрогастроэнтеромиография в детской гастроэнтерологии / А.П. Пономарева, Н.С. Рачкова, С.В. Бельмер, А.И. Хавкин. – М.: Методические аспекты, 2007. – 48 с. 12. Смирнова Г.О. Периферическая электрогастроэнтерография в клинической практике: Пособие для врачей / Г.О. Смирнова, С.В. Силуянов. – М., 2009 – 20 с. 13. Ступин В.А. Периферическая электрогастроэнтерография в диагностике нарушений моторно-эвакуаторной функции желудочно-кишечного тракта / В.А. Ступин, Г.О. Смирнова, М.В. Баглаенко, С.В. Силуянов, Д.Б. Закиров. – М.: Лечащий врач, 2005. – № 2. – С. 60–62. 14. Cole L.G. Roentgenocinematography of stomach and cap / L.G. Cole / Am. J. Roentgenol. – 1914. – Vol. 107. – P. 212–228. 15. Драган Я.П. Енергетична теорія лінійних моделей стохастичних сигналів / Я.П. Драган. – Львів: Центр стратегічних досліджень еко-біо-технічних систем, 1997. – XVI+333с.

Вікторія Гамар, Віталія Гамар, Богдан Демида
Національний університет “Львівська політехніка”,
кафедра інформаційних управляючих систем та мереж

ДОСЛІДЖЕННЯ МЕТОДИКИ ПОШУКУ ОПТИМАЛЬНОГО МАРШРУТУ ПАСАЖИРІВ У ТРАНСПОРТНІЙ МЕРЕЖІ МІСТА

© Гамар Вікторія, Гамар Віталія, Демида Богдан, 2012

Запропоновано алгоритм пошуку оптимального маршруту пасажирів у транспортній мережі міста методом відгалужень і меж.

Ключові слова: оптимальний маршрут, метод відгалужень і меж, транспортна мережа.

The shortest path search algorithm in the public transport network is worked out taking into account time of transfers by a branch-and-bound method.

Key words: optimal route, branch-and-bound method, transport network.

Вступ

В умовах сьогодення проблема розв'язання задачі пошуку оптимальних шляхів є надзвичайно актуальною, особливо враховуючи постійну зайнятість населення та бажання ефективно використовувати свій час. Але це завдання є значно складнішим, ніж задача пошуку найкоротшого шляху на транспортній мережі, оскільки в деяких випадках виникає необхідність зміни маршруту та (або) виду транспорту (тобто, здійснення пересадки). Слід також врахувати, що не завжди безпересадочний шлях є найкоротшим за критерієм тривалості чи відстані, оскільки тривалість пересадки найчастіше складається з тривалості пішого переходу пасажирів між зупинками (залежить від просторового розміщення зупинок) та тривалості очікування транспорту, що залежить від інтервалу руху транспорту.

Тільки шляхом повного перебору всіх можливих варіантів можна розв'язати задачу пошуку оптимального маршруту у міських транспортних мережах [1]. Як показано у [2], кількість можливих варіантів маршрутних схем дорівнює $2n*(n-1)-1$, де n – кількість зупинок громадського транспорту. Зі збільшенням n це значення швидко зростає, і вже при $n=10$ становить приблизно $1,24 \times 10^{27}$ варіантів. Зрозуміло, що повний перебір такої кількості варіантів потребує багато часу і вимагає дуже потужної обчислювальної техніки.

Аналіз публікацій [2–4] показав, що автори не приділили достатньої уваги цій задачі. Так, у [2] наведений приклад розробки раціональної маршрутної схеми на доволі простому прикладі та зазначено, що пошук найкоротших шляхів з врахуванням тривалості пересадок розраховується методом потенціалів за аналогією з пошуком найкоротших шляхів без врахування тривалості пересадок. Однак, у такому випадку цей метод зводиться до перебирання всіх можливих варіантів прямування пасажирів, здійснити який можна лише для транспортних мереж з малою кількістю транспортних районів. Аналогічний підхід пропонується і в роботі [3], без конкретної реалізації використовуюваного алгоритму. У роботі [4] для побудови маршрутної схеми відшуковуються шляхи з однією, двома та трьома пересадками, метод пошуку таких шляхів не наведено.

Опис алгоритму

Метою статті є розробка ефективного алгоритму пошуку найкоротших шляхів на маршрутній мережі з врахуванням тривалості пересадок. Для цього ми використали метод відгалужень і меж.

Для детальнішого пояснення алгоритму наведемо приклад.

На рис. 1 наведена схема маршрутної мережі. У ній є $n=5$ вершин, якими позначені центри транспортних районів, та $R=5$ маршрутів.

Кожен маршрут – це список вершин, через які він проходить. Отже,

- маршрут 1 – (1, 2, 3);
- маршрут 2 – (2, 3, 4);
- маршрут 3 – (1, 4);
- маршрут 4 – (1, 5);
- маршрут 5 – (4, 5).

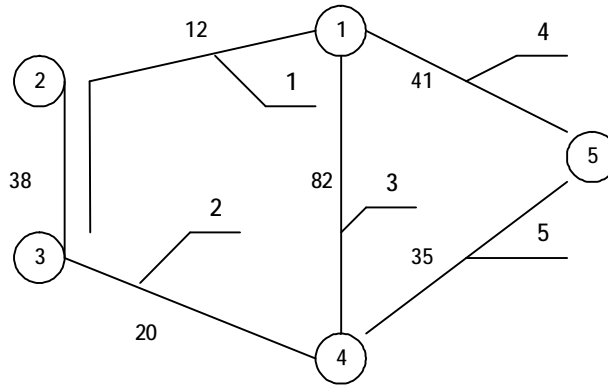


Рис. 1. Схема маршрутної мережі

Вважатимемо, що тривалість руху між вершинами є постійною, а тривалість пересадки у деякій i -й вершині залежить тільки від її номера, таблиця.

Тривалості пересадок у кожній з вершин маршрутної мережі (у хвиликах)

Номер вершини, k	1	2	3	4	5	6	7	8
Тривалість пересадки, хв	5	8	10	2	1	6	2	3

Шукатимемо найшвидший шлях між вершинами $p = 1$ та $q = 4$.

Оскільки між потрібними вершинами може бути декілька шляхів різної довжини та тривалості, то задачу можна розв'язати простим перебором всіх можливих варіантів. Але в реальних умовах цей вихід може бути неефективним і дуже затратним, зважаючи на величезну кількість можливих шляхів. У такому разі доцільно скористатися методом відгалужень і меж, який дасть нам змогу значно скоротити кількість варіантів перебору.

Ідея методу відгалужень і меж під час розв'язання задач комбінаторної оптимізації полягає у послідовному розбитті множини всіх можливих розв'язків задачі на підмножини, які не перетинаються, з подальшим аналізом отриманих елементів розбиття та відкиданням тих з них, які заздалегідь не містять оптимального розв'язку.

Наведемо алгоритм розв'язку поставленої задачі методом відгалужень і меж.

1. Розрахунок найкоротших відстаней між всіма вершинами без врахування необхідності пересадок. Для цього доцільно скористатися алгоритмом Флойда-Воршала.

2. Знаходження довжини найкоротшого шляху між початковою та кінцевою вершинами без врахування пересадок $h(T)$. У нашому прикладі найкоротшим між вершинами 1 та 4 буде шлях $L=(1-2-3-4)$ довжиною = 70 хв.

Отже, $h(T)=70$ хв.

3. Пошук верхньої оцінки довжини найкоротшого шляху $H(T)$.

Щоб встановити верхню оцінку довжини найкоротшого шляху, рухатимемось від вершини 1 до вершини 4 так, щоб за можливості здійснювати на шляху прямування якнайменшу кількість пересадок.

Обираємо найдовший маршрут на шляху L – це маршрут 1. Ним з вершини 1 можна без пересадок дістатися вершини 3, у якій ми пересідаємо на маршрут 2. Він нас доводить до кінцевої

вершині 4. Отже, на шляху з вершини 1 до вершини 4 ми здійснили лише одну пересадку у вершині 3. Отже, верхня оцінка довжини найкоротшого шляху дорівнюватиме $H(T)=h(t)+\tau z=80$ хв.

Кроки 2 та 3 нам показали, що шуканий найкоротший шлях буде не коротшим, ніж $h(t)=70$ хв. і не довшим, ніж $H(T)=80$ хв. Якщо $h(t)=H(T)$ шлях L є оптимальним, а отже, задача розв'язана.

Дуже часто процес вирішення проблеми зручно представити у вигляді дерева, на гілках якого розмітити допустимі розв'язки, а поруч з вершинами дерева – вказати нижню оцінку відповідної підмножини.

4. Розгалуження дерева пошуку розв'язків.

Щоб розгалузити корінь дерева пошуку розв'язків, знайдемо всі маршрути, які проходять через початкову вершину 1. У цьому випадку це маршрути 1, 3 та 4. Кожному з цих маршрутів поставимо у відповідність окрему гілку дерева пошуку розв'язків. Нижню оцінку кожної з цих вершин приймають такою, що дорівнює нижній оцінці довжини найкоротшого шляху $h(t)$, тобто $b(1,1)=b(1,3)=b(1,4)=h(T)=70$ хв. (рис. 2). Вершина дерева позначається двома цифрами: перша є номером вузла на маршрутній мережі, а друга – номером маршруту.

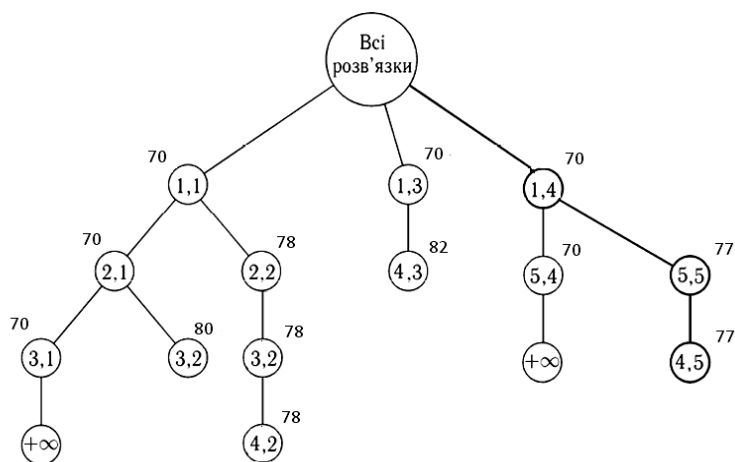


Рис. 2. Дерево пошуку розв'язків задачі

5. Наступним кроком є вибір вершини-кандидата для розгалуження. Необхідно вибрати нерозгалужену вершину (k^*, r^*) з мінімальним значенням оцінки, не враховуючи ті вершини, оцінка яких вже перевищує верхню оцінку довжини найкоротшого шляху $H(T)$. Отже, ми зменшуємо кількість кроків, необхідних для розв'язку поставленої задачі. У нашому випадку три вершини мають однакову мінімальну оцінку:

$$b^*=b(1,1)=b(1,3)=b(1,4)=70.$$

Обираємо будь-яку з них. Нехай це буде вершина (1,4).

6. Перевірка вершини, її оцінки.

Якщо $k^*=q$, то оптимальний розв'язок знайдено і довжина найкоротшого шляху дорівнює b^* . Інакше слід перейти до виконання кроку 7. Наразі, у нашому прикладі $(k^*=1) \neq (q=5)$, тому рухаємося далі.

7. Розгалуження вершини-кандидата.

Для розгалуження обраної вершини (k^*, r^*) необхідно розглянути наступну вершину мережі, яка повинна бути сусідньою вершиною до обраної і знаходитись на тому ж маршруті. Нагадаємо, що k^* – номер вершини, r^* – номер маршруту в маршрутній мережі. Вже відвідані вершини не враховуються, тобто можливість повернення на крок назад виключається.

Для обраної на кроці 5 вершини (1,4) це буде вершина $k^*=5$.

Розглянемо маршрути, до яких належить вершина 5: це маршрути $r_5^{(5)}$ та $r_5^{(4)}$. Отже, з вершини-кандидата (1,4) маємо два відгалуження – у вершини (5,4) та (5,5).

Якщо на обраній вершині k^* завершується маршрут r^* , то наступним відгалуженням з цієї вершини з оцінкою $+\infty$ позначимо неможливість подальшого розгалуження.

8. Розрахунок оцінок відгалужених вершин.

Оцінка кожної з вершин, відгалужених на кроці 7, визначається за формулою

$$b^{\wedge} = b^* + t_{k^*k^{\wedge}} + d_{k^{\wedge}q} - d_{k^*q} + f(k^{\wedge}, r^*, r^{\wedge}), \quad (1)$$

де b^* – оцінка попередньої вершини дерева пошуку розв'язків, хв; $t_{k^*k^{\wedge}}$ – відстань між вершинами k^* та k^{\wedge} за маршрутом r^* , хв; $d_{k^{\wedge}q}$ – найкоротша відстань між відгалуженою вершиною k^{\wedge} та кінцевою вершиною q на маршрутній мережі без врахування пересадок (див. крок 1), хв; d_{k^*q} – найкоротша відстань між попередньою вершиною дерева пошуку розв'язків та кінцевою вершиною q на маршрутній мережі без врахування пересадок, хв; $f(k^{\wedge}, r^*, r^{\wedge})$ – функція, що враховує тривалість пересадки:

$$f(k^{\wedge}, r^*, r^{\wedge}) = \begin{cases} 0, & \text{якщо } r^* = r^{\wedge}; \\ t_{k^{\wedge}}, & \text{якщо } r^* \neq r^{\wedge}. \end{cases} \quad (2)$$

Покажемо, наприклад, обчислення оцінки для відгалуженої вершини (5,5). Оскільки попередньою вершиною дерева розв'язків є вершина (1,4), це відгалуження відповідає пересуванню з вершини $k^*=1$ до вершини $k^{\wedge}=5$ з використанням маршруту $r^*=4$ з пересадкою у вершині 5 на маршрут $r^{\wedge}=5$.

Оцінка вершини (5,5) відповідно до формули (1):

$$b(5,5) = b(1,4) + t_{15} + d_{54} - d_{14} + f(5,4,5) = 70 + 41 + 35 - 70 + 1 = 77 \text{ хв.}$$

Тут, $f(5, 4, 5) = t_{54} = 1 \text{ хв.}$

Визначивши оцінки всіх відгалужених на кроці 7 вершин, повертаємося до кроку 5.

Результатом виконання алгоритму для розглянутого прикладу є отримання оптимального розв'язку у вершині (4,5) з оцінкою 77 хв. Відповідна гілка дерева позначена на рис. 2 потовщеною лінією. Рухаючись від вершини (4,5) до кореня дерева, легко знайти оптимальний шлях. Він відповідає послідовності вершин (1,4) – (5,5) – (4,5). Отже, найкоротший шлях між вершинами 1 та 4 заданої у прикладі маршрутної мережі проходить через вершини 1 – 5 – 4 та дорівнює 77 хв. При цьому необхідно виконати пересадку у вершині 5 (з маршруту 4 на маршрут 5).

Структура бази даних

Для успішного розв'язання задачі знаходження оптимального маршруту в міській мережі варто користуватися такою структурою бази даних, яка наведена на рис. 3 (тут зображено лише основні таблиці).

The figure shows three screenshots of a database management tool (likely MySQL Workbench) displaying the structure of three tables: 'routes', 'stops', and 'transport'.

Table 'routes' structure:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	RouteID	int(11)			No	None	AUTO_INCREMENT	Change
2	TransportNum	int(11)			No	None		Change
3	Interval	int(11)			No	None		Change
4	TransportType	varchar(50)	cp1251_general_ci		No	None		Change

Table 'stops' structure:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	StopID	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	XCoord	int(11)			No	None		Change Drop More
3	YCoord	int(11)			No	None		Change Drop More
4	Section	int(11)			No	None		Change Drop More

Table 'transport' structure:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	RouteID	int(11)			No	None		Change Drop More
2	StopA	int(11)			No	None		Change Drop More
3	StopB	int(11)			No	None		Change Drop More
4	Distance	varchar(50)	cp1251_general_ci		No	None		Change Drop More

Рис. 3. Структура бази даних

Всю карту ми розбили на секції для зручнішого знаходження зупинок.

Таблиця *Routes* містить інформацію про інтервали руху кожного виду транспорту, таблиця *Stops* вказує, до якої секції входить зупинка, з уточненням її координат, а *Transport* – відстань між зупинками, хв.

На рис. 4 показана екранна форма адміністративної частини наповнення бази даних параметрами транспортної мережі.

Висновки

Алгоритм призначений для пошуку найкоротшого шляху між заданою парою вершин, але фактично він дає змогу отримати набагато більше корисної інформації. Якщо повернутися до розглянутого прикладу, то ми знайшли найкоротші шляхи не лише між парою вершин 1 та 4, а й найкоротші шляхи від вершини 1 до всіх інших вершин маршрутної мережі. Це значно скорочує час та обсяг розрахунків у разі пошуку найкоротших шляхів між всіма парами вершин маршрутної мережі.

На кроці 53'являється необхідність зберігати і сортувати оцінки всіх нерозгалужених вершин у порядку зростання. Найкращим рішенням під час реалізації на ЕОМ буде зберігання нерозгалужених вершин та їх оцінок у бінарній купі, яка виконує операції додавання та витягнення вершини за логарифмічний час $O(\log_n)$. При цьому кожна розгалужена вершина до купи не повертається. Такий підхід дозволяє значно скоротити час, затрачений на виконання алгоритму, оскільки якщо попередньо обчислити оцінку кожної з вершин b^i і у випадку $b^i > H(T)$, відгалуження у цю вершину не виконувати.

1. Magnanti T.L. *Network design and transportation planning: models and algorithms* / T.L. Magnanti, R. T. Wong // *Transportation Science*. – 1984. – № 18(1). – P. 1–55. 2. Геронимус Б.Л. *Экономико-математические методы в планировании на автомобильном транспорте* / Б.Л. Геронимус, Л.В. Царфин. – М.: Транспорт, 1988. – 192 с. 3. Хрущев М.В. *Исследование методов маршрутизации автобусного транспорта в городах: дис. ... д-ра экон. наук* / М.В. Хрущев. – М., 2000. – 206 с. 4. Ігнатенко О.С. *Організація автобусних перевезень у містах* / О.С. Ігнатенко, В.С. Маруніч. – К.: УТУ, 1998. – 196 с. 5. Кузькін О.Ф. *Пошук найкоротших шляхів у міських маршрутних мережах* / О.Ф. Кузькін // *Східноєвропейський журнал передових технологій*. – 2011. – № 6/4(54). – С. 8–12.

Transport No:

Interval:

Transport type:

RouteID	Transport No	Interval	Transport Type	
<input type="radio"/> 1	2	8		<input type="checkbox"/>
<input type="radio"/> 2	2	8		<input type="checkbox"/>
<input type="radio"/> 4	7	13		<input type="checkbox"/>
<input type="radio"/> 5	2	17		<input type="checkbox"/>
<input type="radio"/> 6	8	8		<input type="checkbox"/>
<input type="radio"/> 7	9	15		<input type="checkbox"/>
<input type="radio"/> 8	1	8		<input type="checkbox"/>
<input type="radio"/> 9	1	5		<input type="checkbox"/>
<input type="radio"/> 10	60	8		<input type="checkbox"/>
<input type="radio"/> 11	12	8		<input type="checkbox"/>
<input type="radio"/> 12	24	6		<input type="checkbox"/>
<input type="radio"/> 13	11	8		<input type="checkbox"/>

Рис. 4. Екранна форма адміністративної частини транспортної мережі

МОДЕЛЮВАННЯ МЕТОДИК РЕАЛІЗАЦІЇ ТЕМПЕРАТУРНОЇ ШКАЛИ БАГАТОТОЧКОВИМ МЕТОДОМ ДЛЯ ТЕРМОМЕТРІВ ВИПРОМІНЕННЯ

© Гоц Н., 2012

Розглянутий багатоточковий метод реалізації температурної шкали. На основі комп'ютерного моделювання проведений порівняльний аналіз різних методик її побудови. Обґрунтовано переваги застосування комбінованого методу для реалізації температурної шкали.

Ключові слова: температурна шкала, похибка, багатоточковий метод побудови шкали.

In the article the method of multipoint temperature scale. Based on computer modeling of the comparative analysis of different methods conducted. The advantages of the use of the combined method for the realization of the temperature scale.

Key words: temperature scale, error, multipoint method for realisation the temperature scale.

Актуальність

У світі розвиваються два основних напрямки побудови температурної шкали, закладені ще на початку XVIII сторіччя в роботах Амонтоні і Фаренгейта. Перший з них базується на єдиній фіксованій точці і не залежить від властивостей термометричної речовини, що використовується та веде до створення термодинамічної температурної шкали. Другий шлях, заснований на використанні певних фіксованих точок і інтерполяційних термометрів, веде до створення практичних температурних шкал, еволюція яких наведена в табл. 1.

Таблиця 1

Еволюція розвитку міжнародних практичних температурних шкал

№	Назва шкали	Температурні діапазон	Прилад для реалізації	Чинність
1	Normal Hydrogen scale NHS	-25 °С.....100 °С	Газовий термометр	Не чинна
2	Міжнародна температурна шкала МТШ-27	-198 °С.....660 °С	Платиновий термометр опору	Не чинна
		660 °С.....1063 °С	Платино-ренієва термопара	
		1063 °С.....	Термометр випромінення	
3	Міжнародна практична температурна шкала МПТШ-48	-183.....660 °С	Платиновий термометр опору	Не чинна
		660 °С.....1063 °С	Платино-ренієва термопара	
		1063 °С.....	Термометр випромінення	
4	Міжнародна практична температурна шкала МПТШ-68	13,81 К.....660,74 °С	Платиновий термометр опору	Не чинна
		660,74 °С...1064,43°С	Платино-ренієва термопара	
		1064,43 °С.....	Термометр випромінення	
5	Тимчасова температурна шкала ТТШ-76	13,81 К.....660,74 °С	Платиновий термометр опору	Не чинна
		660,74 °С...1064,43 °С	Платино-ренієва термопара	
		1064,43 °С.....	Термометр випромінення	
6	Міжнародна температурна шкала МТШ-90	0,65 К...5 К	Вимірювання тиску газів	Чинна
		3К.....24,5561 К	Газовий термометр	
		13,803 К...961,78 °С	Платиновий термометр опору	
		961,78 °С.....	Термометр випромінення	
7	Тимчасова міжнародна низько-температурна шкала ТМНШ-2000	0,902 мК К до 1 К	Вимірювач тиску газу	Чинна

Міжнародним бюро мір та ваг визначені такі напрями розвитку температурної шкали [1]:

- збереження і модернізація Міжнародної температурної шкали – 1990 (МТШ-90);
- відмова від практичних шкал і відтворення термодинамічної температурної шкали за допомогою первинних термометрів;

Міжнародний консультативний комітет з термометрії рекомендує такі основні напрямки досліджень з вдосконалення МТШ-90 [1]:

1. Удосконалити інтерполяційні прилади для діапазону температур від точки тверднення алюмінію до точки тверднення срібла, можливо, на основі золото-платинової термопари, оскільки надійність звичайних високотемпературних термометрів опору в цих умовах низька.

2. Досліджувати можливості використання нових математичних методів для створення точніших інтерполяційних функцій.

3. Завершити дослідження наявних реперних точок в МТШ-90 і провести дослідження з метою створення нових реперних точок, зважаючи на обмеження у використанні з цієї метою ртуті.

Отже, можна вважати, що сучасним напрямком реалізації температурної передачі її контактним термометрам та термометрам випромінення є розвиток методів первинної термометрії, що базуються на використанні природних закономірностей, зокрема на законі випромінення Планка.

Мета дослідження

Тому метою статті є дослідження можливостей реалізації температурної шкали (ТШ) на засадах первинної термометрії, а саме термометрії випромінення. Це дасть можливість реалізувати температурну шкалу незалежно від властивостей засобів вимірювання на основі природних законів, зокрема закону Планка. Цей метод міг би існувати паралельно з методом, визначеним МТШ-90, та слугувати для передачі температурної шкали термометрам випромінення.

Аналіз публікацій.

У температурному діапазоні, в якому функціонують термометри випромінення, згідно з МТШ-90 температура визначається в двох основних температурних інтервалах: від 0°C до 961,78°C та від 961,78°C і вище. Розглянемо докладніше методи реалізації температурної шкали.

МТШ-90 від 0 °C до 961,78 °C визначає температуру між потрійною точкою води та точкою тверднення срібла (961,78 °C) за допомогою еталонного платинового термометра опору, який калібрується в реперних точках (потрійній точці води, точці плавлення галію, точках тверднення індію, олова, цинку, алюмінію, срібла) на основі інтерполяційної залежності (1).

$$T - 273,15 = D_0 + \sum_{i=1}^9 D_i \left[\frac{W_r(T_{90}) - 2,64}{1,64} \right], \quad (1)$$

де T – температура; D_i – сталі; $W_r(T_{90})$ – відносний опір платинового термометра.

Аналіз показує, що такий підхід не може забезпечити підвищені вимоги до точності реалізації шкали, оскільки вона значно залежить від варіації властивостей платинових термометрів опору. Причинами непевності при реалізації температурної шкали є: варіації властивостей різних термометрів опору, ефект окислення платини, зміна концентрації вакансій кристалічної ґратки, напруження та деформації чутливого елемента, довготривалий дрейф опору тощо [2].

Вище від точки тверднення срібла 961,78°C побудова температурної шкали відбувається відповідно до МТШ-90 [3] оптичними методами на основі рівняння Планка та ґрунтується на випромінювачі типу абсолютно чорне тіло та основних реперних точках тверднення срібла $T_{90}(\text{Ag})$, золота $T_{90}(\text{Au})$ або міді $T_{90}(\text{Cu})$. Згідно з МТШ-90 температура $T_{90}(x)$ визначається за таким виразом:

$$\frac{b(T_{90})}{b(T_{90}(x))} = \frac{\exp\left(\frac{C_2}{IT_{90}(x)} - 1\right)^{-1}}{\exp\left(\frac{C_2}{IT_{90}} - 1\right)^{-1}}, \quad (2)$$

де $T_{90}(x)$ – одна із альтернативних реперних точок ; $C_2 = 0,014388 \text{ м}\cdot\text{К}$, $b(T_{90})$ та $b(T_{90}(x))$ – потік випромінення відповідно при температурах реперної точки та шуканої.

Цей метод побудови температурної шкали має деякі особливості, а саме для побудови температурної шкали використовуються високоточні термометри випромінення та абсолютно чорні тіла, які можна зарахувати до засобів первинної термометрії. Але він не враховує немонохроматичності спектральних каналів та спектральної чутливості термометра випромінення (ТВ). Основною перевагою цього методу є те, що основою для побудови шкали потрібна лише одна реперна точка.

Постановка задачі

На базі проведеного аналізу можна стверджувати, що актуальним є дослідження таких питань:

- розвиток багатоточкового методу побудови температурної шкали на базі еталонних абсолютно чорних тіл (АЧТ) з реперними точками, як методу первинної термометрії;
- розвиток методик інтерполяції температурної шкали на основі закону Планка з врахуванням немонохроматичності спектральних каналів та спектральної чутливості (ТВ);
- моделювання та порівняльний аналіз методик реалізації багатоточкового методу реалізації ТШ.

Виклад основного матеріалу

Щоб уникнути вищезгаданих проблем для реалізації ТШ, запропоновано багатоточковий метод передачі температурної шкали. Він базується на п'яти реперних точках тверднення металів Zn, Sb, Al, Ag, Cu та використанні інтерполяційних рівнянь [4], які дозволяють відтворити всю множину температур температурної шкали в діапазоні від 419,527 °С до 1084,62 °С. Згідно з цим методом, термометром випромінення візують еталонні АЧТ при декількох (від трьох до п'яти) реперних точках тверднення металів. Отримані вихідні сигнали опрацьовують за відповідним інтерполяційним рівнянням.

З метою аналізу методик реалізації багатоточкового методу реалізації температурної шкали проведено математичне моделювання з використанням програми MathCad різних методик реалізації багатоточкового методу. Дослідження проведені в діапазонах, наведених в табл. 2.

Таблиця 2

Границі відповідних спектральних та температурних діапазонів моделювання методів реалізації температурної шкали

№	Температурний діапазон, К	Спектральний діапазон, мкм
1	273,16 ÷ 429,748	10 ÷ 14
2	429,748 ÷ 692,677	3 ÷ 3,5
3	692,677 ÷ 1234,93	1 ÷ 2

Для реалізації температурної шкали багатоточковим методом можна використовувати інтерполяційне рівняння Сакума-Хатторі, яке має вигляд (3), згідно з рівняннями Віна та Планка [4]:

$$S(T) = \frac{C}{\exp\left(\frac{C_2}{AT+B}\right)} \quad \text{та} \quad S(T) = \frac{C}{\exp\left(\frac{C_2}{AT+B}\right) - 1}, \quad (3)$$

де А, В і С – параметри, які підбираються методом найменших квадратів по результатах градування в реперних точках; $S(T)$ – вихідний сигнал термометра випромінення [5].

У результаті проведеного моделювання з використанням програми MathCad було встановлено, що абсолютна похибка побудови температурної у високотемпературній області на основі рівняння Сакума-Хатторі та закону Віна не перевищує 1,5 °С. У низькотемпературній області похибка реалізації температурної шкали на основі рівняння Сакума-Хатторі та закону Планка зростає до декількох десятків градусів. Це пояснюється залежністю результатів вимірювання температури від ширини робочого спектрального каналу при побудові шкали у дальній

інфрачервоної області спектра. Визначено, що рівняння Сакума-Хатторі на основі закону Планка не у всій області значень має розв'язки.

Для врахування ширини спектрального каналу запропоновано використання температурної залежності середніх довжин хвиль для конкретного температурного та спектрального інтервалу. За законом Планка для спектрального інтервалу $\lambda_1 \div \lambda_2$ та певної температури T_i існує така довжина хвилі λ_{ci} , для якої виконується рівність, потік випромінювання $b(I_{ci}, T_i)$ визначається згідно з (4):

$$b(I_{ci}, T_i) = \frac{1}{I_2 - I_1} \int_{I_1}^{I_2} b(I, T_i) dI \quad . \quad (4)$$

Графік температурної залежності значень середніх довжин хвиль для спектрального інтервалу $\lambda_1 \div \lambda_2$ та температурного діапазону $T_1 \div T_i$ наведений на рис.1.

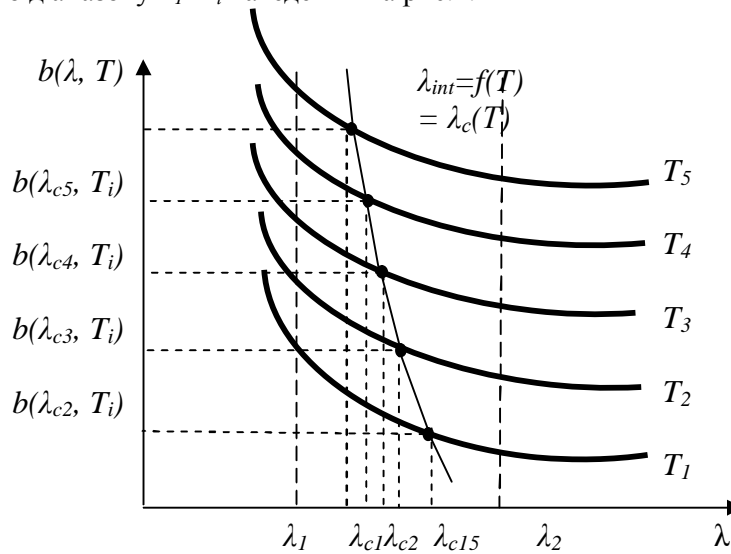


Рис. 1. Схематичне представлення температурної залежності значень середніх довжин хвиль для спектрального інтервалу $\lambda_1 - \lambda_2$ та температурного діапазону $T_1 - T_i$

Під час побудови та передачі температурної шкали для певного температурного діапазону середні значення довжини хвилі змінюватимуться зі значенням температури. І ця зміна тим значніша, чим більша ширина спектрального та температурного інтервалу. Однак, визначивши середні значення довжини хвилі в декількох точках цього діапазону, можемо побудувати функціональну залежність середніх довжин хвиль від температури. На базі даних про значення усереднених довжин хвиль для реперних температур ми можемо побудувати апроксимаційну залежність $\lambda_{int} = f(T)$ методом найменших квадратів.

Умовою побудови цієї залежності є послідовне зростання або спадання значень середніх довжин хвиль λ_{cn} , що визначає ширину температурного та спектрального діапазону під час реалізації ТШ.

$$\lambda_{c1} < \lambda_{c2} < \lambda_{cn} \quad \text{або} \quad \lambda_{c1} > \lambda_{c2} > \lambda_{cn} \quad . \quad (5)$$

Цю залежність можна визначити для двох та більше реперних температурних точок. Температурна залежність довжини хвилі може мати лінійний, гіперболічний, степеневий або експоненційний характер (залежно від значення похибки апроксимації) та бути використана в інтерполяційному рівнянні шкали на основі закону Планка залежності вихідного сигналу термометра випромінювання $S_{int}(T)$ від температури:

$$S_{int}(T) = \frac{C_1 I_c(T)^{-5}}{\frac{C_2}{e^{I_c(T) T}} - 1} \quad . \quad (6)$$

Результати моделювання різних видів температурних залежностей середньої довжини хвилі наведені в табл. 3.

Результати моделювання різних видів температурних залежностей середньої довжини хвилі

№	Температурний діапазон, К	Види визначених температурних залежностей температурної залежності середньої довжини хвилі
1	273,16 ÷ 429,748	$\lambda = -2,724 \times 10^{-10} \times T + 1,814 \times 10^{-6}$
		$\lambda = 2,308 \times 10^{-4} \cdot T + 1,298 \times 10^{-6}$
		$\lambda = -1,981 \times 10^{-12} \times T^2 + 3,618 \times 10^{-9} \times T$
2	429,748 ÷ 692,677	$\lambda = -7,78 \times 10^{-11} \times T + 3,291 \times 10^{-6}$
		$\lambda = 2,354 \times 10^{-5} \cdot T + 3,204 \times 10^{-6}$
		$\lambda = -1,049 \times 10^{-11} \times T^2 + 1,192 \times 10^{-8} \times T$
3	692,677 ÷ 1234,93	$\lambda = -2,369 \times 10^{-9} \times T + 1,291 \times 10^{-5}$
		$\lambda = 2,928 \times 10^{-4} \cdot T + 1,121 \times 10^{-5}$
		$\lambda = -1,055 \times 10^{-10} \times T^2 + 7,292 \times 10^{-8} \times T$

Запропоновано критерій вибору виду інтерполяційного рівняння на основі температурної залежності середньої довжини хвилі: найкраще інтерполяційне рівняння буде в тому разі, коли для n точок всієї множини досліджуваних температур сума квадратів відхилень інтерполяційної функції від розрахунку за законом Планка буде мінімальною:

$$\left[\sum_{i=1}^n (S_{int}(I, T_i) - S_{Pl}(I, T_i))^2 \right] \min, \quad (7)$$

де $S_{int}(I, T_i)$ – інтерполяційне рівняння температури; $S_{Pl}(I, T_i)$ – рівняння закону Планка для цього температурного та спектрального інтервалу; за умови $n \rightarrow n_{\max}$.

Проведене моделювання методики побудови температурної шкали згідно з рівнянням (6) та критерієм (7) для різних видів температурної залежності середньої довжини хвилі. Визначено доцільність використання лінійної та гіперболічної залежностей для різних спектральних діапазонів.

Під час моделювання розглядали можливості застосування різних типів математичних співвідношень інтерполяційних рівнянь для реалізації температурної шкали. Визначено доцільність реалізації температурної шкали за відношенням вихідних сигналів у двох точках з врахування лінійної температурної залежності середніх довжин хвиль у цьому температурному та спектральному діапазоні згідно з (8).

Ця методика дає можливість побудувати температурну з врахуванням ширини спектрального інтервалу термометра випромінювання, він не потребує інформації про спектральну чутливість термометра випромінювання і враховує немонахроматичність робочого спектрального інтервалу. За відношення вихідних сигналів приймача випромінювання в реперній точці та точці, в якій визначається значення температури та відношення виразів інтерполяційних рівнянь, можна визначити значення температури:

$$m = \frac{S_{\lambda}(T_P)}{S_{\lambda}(T_X)} = \frac{\int_{I_1}^{I_2} C_1 I_{int}^{-5} \left(\exp\left(\frac{C_2}{\lambda_{int} T_P}\right) - 1 \right)^{-1} dI}{\int_{I_1}^{I_2} C_1 I_{int}^{-5} \left(\exp\left(\frac{C_2}{\lambda_{int} T_X}\right) - 1 \right)^{-1} dI}. \quad (8)$$

Згідно з проведеним моделюванням похибка реалізації температурної шкали комбінованим методом для різних температурних та спектральних діапазонів наведена на рис. 2. Для розглянутих температурного та спектрального діапазонів її значення не перевищує 1 °С.

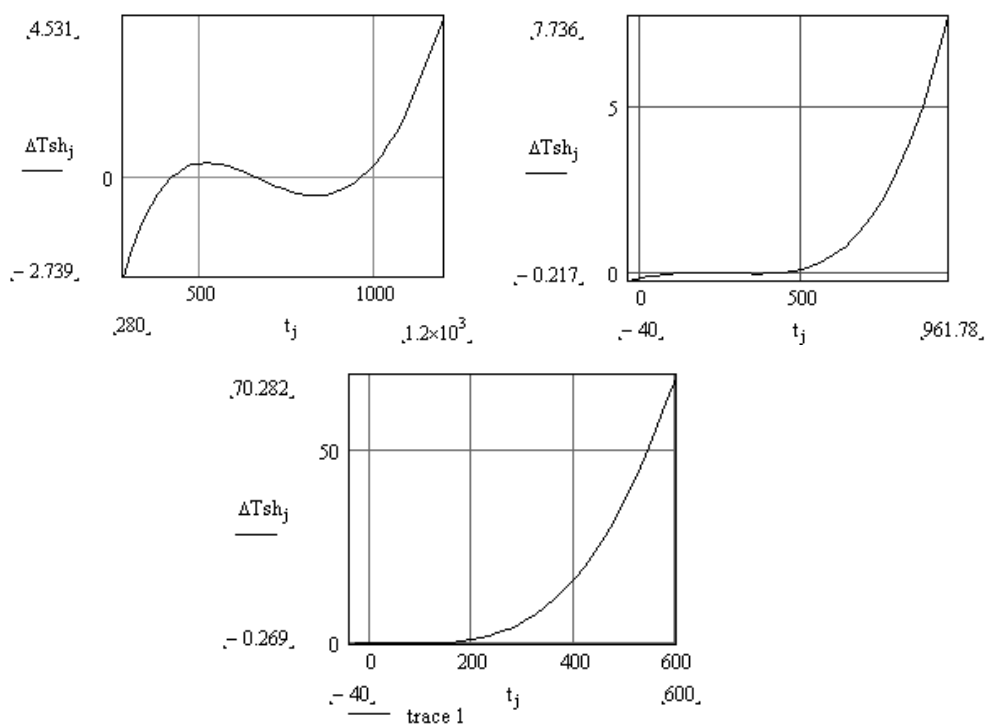


Рис. 2. Температурні залежності абсолютної похибки побудови температурної шкали комбінованим методом для різних температурних та спектральних діапазонів: а – для діапазону 1 – 2 мкм; б – для діапазону 3 – 3,5 мкм; в – для діапазону 10 – 14 мкм

Висновки

Використання інтерполяційного рівняння на основі закону Планка дає можливість реалізувати температурну шкалу на засадах первинної термометрії, що дозволяє позбутися впливу таких характеристик, як спектральна чутливість та ширина робочого спектрального інтервалу засобів вимірювання на точність побудови шкали. Проведене моделювання показало, що тільки дві методики реалізації температурної шкали дають можливість однозначної її побудови у широкому температурному та спектральному інтервалі – це метод на основі рівняння Сакума-Хатторі та комбінованого методу відношення інтерполяційних рівнянь. Точність побудови температурної шкали в широкому температурному діапазоні від 0 до 1500 °С не перевищує 1 °С. На точність визначення температури значний вплив має математичний метод обчислення.

Запропоновано визначення температурної залежності різних видів довжини хвилі з використанням багатоточкового методу побудови температурної шкали. Це дозволяє в інтерполяційному рівнянні користуватися реальною залежністю довжини хвилі від температури для різних спектральних і температурних діапазонів вимірювання. Використання поняття інтерполяційної довжини хвилі дає змогу реалізувати температурну шкалу для різних спектральних та температурних діапазонів та калібрувати термометри випромінювання, які мають декілька робочих спектральних каналів. Запропонований комбінований метод побудови температурної шкали дозволяє будувати температурну шкалу для визначених температурних та спектральних інтервалів з використанням лише двох реперних точок, не потребує інформації про спектральну чутливість термометра випромінювання, оскільки використовує інтерполяційні рівняння на базі інтерполяційної довжини хвилі, що не вимагає обов'язкової лінійності функції перетворення приймача термометра випромінювання.

1. www.bipm.org. 2. www.temperaturas.ru. 3. Preston-Thomas H. *The International Temperature Scale of 1990 (ITS-90)*, *Metrologia*, 1990, 27(1), p. 3–10. 4. Sakuma F., Hattori S., *In Temperature : Its Measurement and Control in Science and Industry, Vol. 5 (Edited by J. F. Schooley)*, New York, American Institute of Physics, Monograph, 1982, p. 421–427. 5. *Interpolation errors for radiation thermometry* / P. Saunders and R White. *Metrologia*, 2004. – № 41. – P. 41–46.

О. Кузьмін¹, А. Ковальчук², Р. Ступень³

Національний університет “Львівська політехніка”,

¹кафедра менеджменту і міжнародного підприємництва,²кафедра інформаційних технологій видавничої справи,³кафедра менеджменту персоналу та адміністрування

РОЗДІЛЕННЯ ТЕРИТОРІЇ НА ЕКОНОМІЧНІ СКЛАДОВІ В УМОВАХ НЕВИЗНАЧЕНОСТІ

© Кузьмін О., Ковальчук А., Ступень Р., 2012

Запропоновано модель розподілу території на економічні зони з використанням методів теорії нечітких множин.

Ключові слова: економічна зона, територія, нечітка множина, модель.

Model of separation of territory on economic zones with by use of methods of the theory of fuzzy sets is offered.

Key words: economic zone, area, fuzzy sets, model.

Вступ

Мета роботи полягає в тому, щоб описати метод розділення на торговельні зони в нечітких умовах, коли інформація за своєю суттю неповна і рішення споживача про вибір не точний. Припущення про однорідність ринку послаблені, наскільки це можливо, з тим, щоб кількість аксіом під час побудови моделей була мінімальною. Основу методу дослідження становить теорія нечітких множин, яка оперує нечітким представленням нечітких понять. Хоча цю теорію вже застосовували в загальному аналізі поведінки людини та, зокрема, в аналізі просторової економічної активності, фундаментальне поняття розділення торговельних зон між фірмами детально не розглядали. Першою спробою такого роду було, очевидно, застосування нечітких кластерів для групування споживачів по фірмах*.

Моделі розділення на торговельні зони в нечітких умовах

Традиційні дослідження торговельної зони часто ґрунтуються на поняттях інформації та деяких припущеннях про однорідність ринку. Поведінкові постулати часто виявляються або занадто спрощеними або неузгодженими.

Такі припущення, як у разі постійності транспортних витрат і однакового рівня фірм, замінюються нечітким сприйняттям відстані і привабливості фірм відносно різних характерних властивостей. Перевага, яку віддають користувачі тій чи іншій фірмі, виявляється у вигляді випуклої нечіткої підмножини для дослідження перекриття торговельних зон. Встановлюється поріг розподілу. У цьому підході замість чітко окресленого опису торговельних зон у традиційному аналізі використовується ступінь розподілу. Побудовані дві моделі, підтверджені прикладами. У першій моделі передбачається, що різниця у відстані – це єдиний вирішальний фактор просторової переваги користувача, у той час як друга модель формулюється з використанням соціально-психологічних і економічних факторів, які впливають на прийняття користувачем рішення про поїздку.

Модель 1 — розділення при лінійній моделі ринку. У моделі прийняті такі припущення:

- 1) існування ринку, що відповідає лінійній моделі;
- 2) довільний характер розподілу населення;
- 3) дві конкуруючі фірми F_1 та F_2 розміщуються в заданих точках відповідно a_1 та a_2 ;

* *Нечеткие множества и теория возможностей. Последние достижения / Пер. с англ.; Под ред. Р. Р. Ягера. – М.: Радио и связь, 1986.*

- 4) виробляється однорідний товар одного типу;
- 5) ціни однакові;
- 6) існує функція просторової переваги споживача, яка змінюється зворотно пропорційно до складності долання шляху до фірми.

З а у в а ж е н н я. Обговорюючи зроблені припущення, зазначимо дві особливості. Перша полягає в тому, що тут не приймається припущення про регулярний розподіл популяції. Розподіл на торговельні зони в цьому формулюванні залежить від виявленого споживачем ступеня переваги для фірм F_1 та F_2 .

Побудувавши нечітке поняття переваги, можемо визначити спосіб, за допомогою якого ринок можна розділити між фірмами. Оскільки нечіткі підмножини A_1 та A_2 обмежені максимальними ступенями $\sup_x m_{A_1}(x)$ та $\sup_x m_{A_2}(x)$ в точках a_1 та a_2 відповідно, то їх перетин $A_1 \cap A_2$ також обмежений випуклою нечіткою підмножиною та визначається функцією належності

$$m_{A_1 \cap A_2}(x) = \begin{cases} e^{-k(x-a_2)^2} & \text{при } k \geq 1, x \leq g, \\ [e^{-k(x-a_1)^2}]^2 & \text{при } k \geq 1, x \geq g, \end{cases} \quad (1)$$

яка приймає максимальне значення $\sup_x m_{A_1 \cap A_2}(x)$ в g . Використовуючи теорему про віддільність, отримуємо, що найвищий ступінь розділення на торговельні зони, який дорівнює $1 - \sup_x m_{A_1 \cap A_2}(x)$,

досягається в точці g , через яку проходить гіперплощина. У класичній моделі фірма монополізує ринок. Однак модель збуту з домінуванням більш адекватна для опису торговельних зон реального світу. Із цієї ідеї випливає, що перекриття торговельної зони скоріше загальний феномен, ніж виняток. Для перекриття торговельних зон фірм F_1 та F_2 можна використовувати поняття порогу роздільності. Поріг розділення l визначається умовою

$$l < \max_x \min[m_{A_1}(x), m_{A_2}(x)] = \sup_x m_{A_1 \cap A_2}(x). \quad (2)$$

Отже, для обраного порогу l торговельна зона M_i фірми F_i , $i=1, 2$ визначається нечіткою підмножиною рівня l . Вибираючи різні значення для l , можна отримати різні торговельні зони. Загальне правило полягає в тому, щоб вибрати найбільш можливе значення l , яке менше $\max_x \min[m_{A_1}(x), m_{A_2}(x)]$:

$$M_i = \left\{ x \mid m_{A_i}(x) \geq \max_x \min[m_{A_1}(x), m_{A_i}(x)] \right\} \text{ для всіх } x \in M_i. \quad (3)$$

Описану тут модель можна узагальнити на ринок з m конкуруючими фірмами F_1, F_2, \dots, F_m , які розміщені в ділянках з густотою населення відповідно a_1, a_2, \dots, a_m . Нехай A_1, A_2, \dots, A_m – обмежені випуклі нечіткі підмножини, які описують перевагу F_1 , перевагу F_2 , ..., перевагу F_m та визначаються функціями належності $m_{A_1}, m_{A_2}, \dots, m_{A_m}$. У зв'язку з випуклістю та обмеженості нечітких множин A_1, A_2, \dots, A_m нечіткі підмножини $A_1 \zeta A_2, A_1 \zeta A_3, \dots, A_1 \zeta A_m, A_2 \zeta A_3, \dots, A_{m-1} \zeta A_m$ будуть також випуклими та обмеженими. Використовуючи вже згадану теорію про віддільність, за допомогою порогу розділення

$$l < \min_{ij} \max_x \min[m_{A_i}(x), m_{A_j}(x)] \quad (4)$$

можемо визначити торговельні зони відповідно M_1, M_2, \dots, M_m . Торговельна зона M_i , $i=1, 2, \dots, m$ буде знову нечіткою рівневою підмножиною, яка визначається відношенням

$$M_i = \left\{ x \mid m_{A_i}(x) \geq \min_{ij} \max_x \min[m_{A_i}(x), m_{A_j}(x)] \right\} \text{ для всіх } x \in M_i. \quad (5)$$

Модель 2 – загальна модель розділення торговельної зони. У моделі прийняті такі припущення:

- 1) існування ринку;
- 2) довільна схема розселення населення;

- 3) розміщення m конкуруючих фірм F_1, F_2, \dots, F_m в певних пунктах;
- 4) продукція одної якості;
- 5) фірми характеризуються p ознаками;
- 6) ступені важливості ознак при прийнятті рішення про поїздку змінюються між індивідуумами;
- 7) одна фірма переважає над іншою, якщо її ознаки за своїм ступенем важливості ближчі до оцінки споживача.

Нехай $X = \{x_1, x_2, \dots, x_n\}$ – множина покупців, а $Y = \{y_1, y_2, \dots, y_p\}$ – множина ознак фірми і $Z = \{z_1, z_2, \dots, z_m\}$ – множина фірм.

Нехай $\Phi_R: X \times Y \rightarrow [0,1]$ є функцією належності нечіткого бінарного відношення R . Для всіх $x \in X$ та всіх $y \in Y$ функція $\Phi_R(x,y)$ – ступінь важливості ознаки y за оцінкою індивідууму x при визначенні ним переваги фірми. Відношення R можна подати в матричній формі:

$$R = \begin{matrix} & y_1 & y_2 & \dots & y_p \\ \begin{matrix} x_1 \\ x_2 \\ \dots \\ x_n \end{matrix} & \left[\begin{array}{cccc} \Phi_R(x_1, y_1) & \Phi_R(x_1, y_2) & \dots & \Phi_R(x_1, y_p) \\ \Phi_R(x_2, y_1) & \Phi_R(x_2, y_2) & \dots & \Phi_R(x_2, y_p) \\ \dots & \dots & \dots & \dots \\ \Phi_R(x_n, y_1) & \Phi_R(x_n, y_2) & \dots & \Phi_R(x_n, y_p) \end{array} \right] \end{matrix}$$

Нехай $\pi: Y \times Z \rightarrow [0,1]$ є функцією належності нечіткого бінарного відношення S . Для всіх $y \in Y$ та всіх $z \in Z$ $\pi_S(y,z)$ – ступеню належності або сумісності фірми z з ознакою y . У матричній формі відношення має вигляд:

$$S = \begin{matrix} & z_1 & z_2 & \dots & z_m \\ \begin{matrix} y_1 \\ y_2 \\ \dots \\ y_p \end{matrix} & \left[\begin{array}{cccc} \pi_S(y_1, z_1) & \pi_S(y_1, z_2) & \dots & \pi_S(y_1, z_m) \\ \pi_S(y_2, z_1) & \pi_S(y_2, z_2) & \dots & \pi_S(y_2, z_m) \\ \dots & \dots & \dots & \dots \\ \pi_S(y_p, z_1) & \pi_S(y_p, z_2) & \dots & \pi_S(y_p, z_m) \end{array} \right] \end{matrix}$$

Тепер можна отримати матрицю T :

$$T = \begin{matrix} & z_1 & z_2 & \dots & z_m \\ \begin{matrix} x_1 \\ x_2 \\ \dots \\ x_n \end{matrix} & \left[\begin{array}{cccc} m_{A_1}(x_1, z_1) & m_{A_2}(x_1, z_2) & \dots & m_{A_m}(x_1, z_m) \\ m_{A_1}(x_2, z_1) & m_{A_2}(x_2, z_2) & \dots & m_{A_m}(x_2, z_m) \\ \dots & \dots & \dots & \dots \\ m_{A_1}(x_n, z_1) & m_{A_2}(x_n, z_2) & \dots & m_{A_m}(x_n, z_m) \end{array} \right] \end{matrix},$$

елементи якої визначаються функцією належності

$$m_{A_i}(x, z_i) = \frac{\sum_y \Phi_R(x, y) \cdot \pi_S(y, z_i)}{\sum_y \Phi_R(x, y)} \quad \text{для всіх } x \in X, y \in Y \text{ та } z \in Z \quad (6)$$

Сума $\sum_y \Phi_R(x, y)$ дорівнює ступеню нечіткої підмножини, що вказує на кількість найважливіших ознак y , які споживач x використовує для оцінки фірми, а $m_{A_i}(x, z_i)$ інтерпретувати як виважений ступінь переваги фірми z_i індивідумом x . Функція переваги, яка описана рівнянням (6), задовольняє визначення випуклої нечіткої підмножини

$$m_{A_i} [I(x_1, z_i) + (1 - I)(x_2, z_i)] \geq \min[m_{A_i}(x_1, z_i), m_{A_i}(x_2, z_i)] \forall x_1, \forall x_2, \forall z_i \in Z, \forall I \in [0, 1]. \quad (7)$$

Оскільки всі $m_{A_i}(x, z_i)$ випуклі, їх перетини також випуклі функції. Отже, можна побудувати матрицю W :

$$W = \begin{bmatrix} m_{A_1}(x_1, z_1) \dot{\cup} m_{A_2}(x_1, z_2) & \dots & m_{A_{m-1}}(x_1, z_{m-1}) \dot{\cup} m_{A_m}(x_1, z_m) \\ m_{A_1}(x_2, z_1) \dot{\cup} m_{A_2}(x_2, z_2) & \dots & m_{A_{m-1}}(x_2, z_{m-1}) \dot{\cup} m_{A_m}(x_2, z_m) \\ m_{A_1}(x_n, z_1) \dot{\cup} m_{A_2}(x_n, z_2) & \dots & m_{A_{m-1}}(x_n, z_{m-1}) \dot{\cup} m_{A_m}(x_n, z_m) \end{bmatrix}$$

З тих самих міркувань, які були наведені в лінійній моделі розділення ринку, у цій моделі поріг розділення торговельної зони може бути обмежений умовою

$$l < \min_{ij} \max_x \min[m_{A_i}(x, z_j)]. \quad (8)$$

Якщо поріг l вибраний, то торговельна зона M_i , $i=1, 2, \dots, m$ описується рівневою множиною

$$M_i = \{x | m_{A_i}(x) \geq \min_{ij} \max_x \min[m_{A_i}(x, z_j), m_{A_j}(x, z_j)]\} \text{ для всіх } x \in M_i. \quad (9)$$

Для ілюстрації практичного використання описаних теоретичних результатів про задачу розділення торговельної зони розглянемо приклад.

Нехай $X = \{x_1, x_2, \dots, x_{12}\}$ – множина покупців, а $Y = \{y_1, y_2, y_3, y_4\}$ – множина ознак фірми і $Z = \{z_1, z_2, z_3, z_4\}$ – множина фірм. При значеннях елементів матриць R і S , наведених нижче, територія розподіляється на зони M_1, M_2, M_3, M_4 , що графічно показано на рис. 1.

$$R = \begin{bmatrix} 1 ; 0 ; 0 ; 0 \\ 0 ; 1 ; 0 ; 0 \\ 0 ; 0 ; 1 ; 0 \\ 0 ; 0 ; 0 ; 1 \\ 1 ; 1 ; 1 ; 1 \\ 0.8 ; 0.4 ; 0.5 ; 0.9 \\ 0.7 ; 0.3 ; 0.4 ; 0.8 \\ 0.5 ; 0.8 ; 0.8 ; 0.2 \\ 0.5 ; 0.5 ; 0.5 ; 0.5 \\ 0.6 ; 0.7 ; 0.8 ; 0.5 \\ 0.1 ; 0.1 ; 0.1 ; 0.1 \\ 0 ; 0 ; 1 ; 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.9; 0.1; 0.5; 0.7 \\ 0.5; 0.9; 0.6; 0.6 \\ 0.4; 0.9; 0.5; 0.4 \\ 0.8; 0.1; 0.5; 0.6 \end{bmatrix}$$

$$M_1 = \{x_1, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\},$$

$$M_2 = \{x_2, x_3, x_8, x_{10}\},$$

$$M_3 = \{x_2, x_8, x_{10}\},$$

$$M_4 = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}.$$

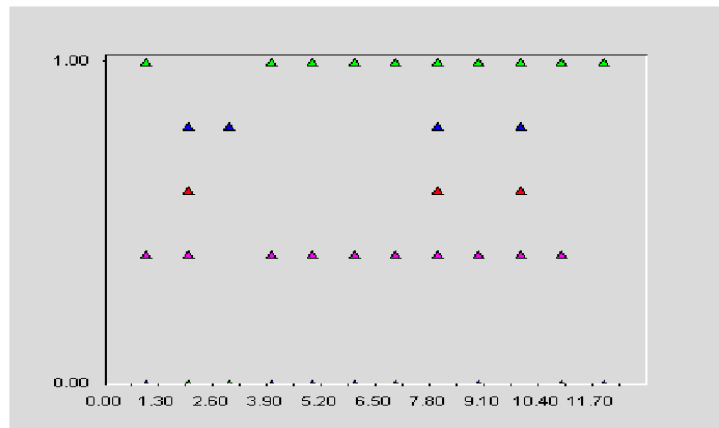


Рис. 1. Графічне зображення зон, на які розподіляють територію

Якщо матриці R і S мають вигляд

$$R = \begin{bmatrix} 1; 0; 0; 0 \\ 0; 1; 0; 0 \\ 0; 0; 1; 0 \\ 0; 0; 0; 1 \\ 1; 1; 1; 1 \\ 0.8; 0.4; 0.5; 0.9 \\ 0.7; 0.3; 0.4; 0.8 \\ 0.5; 0.8; 0.8; 0.2 \\ 0.5; 0.5; 0.5; 0.5 \\ 0.6; 0.3; 0.3; 0.5 \\ 0.1; 0.1; 0.1; 0.1 \\ 0; 0; 1; 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.9; 0.1; 0.5; 0.7 \\ 0.5; 0.9; 0.6; 0.6 \\ 0.4; 0.1; 0.5; 0.4 \\ 0.8; 0.1; 0.5; 0.6 \end{bmatrix}$$

територія розподіляється на зони M_1, M_2, M_3, M_4

$$M_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\},$$

$$M_2 = \{x_2\},$$

$$M_3 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\},$$

$$M_4 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\},$$

що графічно показано на рис.2.

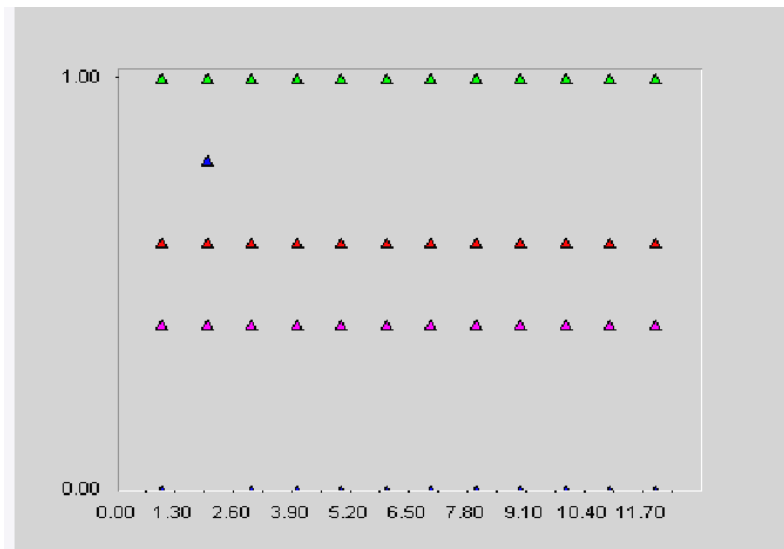


Рис. 2. Графічне зображення зон

Висновки

У розглянутих прикладах показано можливість застосування описаної моделі в розподілі території на економічні зони. Запропонована модель може бути застосовна під час планування економічних експериментів в конкурентній боротьбі між фірмами.

МЕТОДИ І АЛГОРИТМИ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 629.735.015.7.2:656.7.08

Д. Шевчук
Национальный авиационный университет

КОНЦЕПЦИЯ РЕКОНФИГУРАЦИИ УПРАВЛЕНИЯ ДЛЯ ВОССТАНОВЛЕНИЯ УПРАВЛЯЕМОСТИ И УСТОЙЧИВОСТИ САМОЛЕТА В ОТКАЗНЫХ СИТУАЦИЯХ

© Шевчук Д., 2012

Обоснована возможность применения системных методов реконфигурации управляющих поверхностей для возобновления управляемости и устойчивости самолета в условиях внезапного возникновения особой ситуации в полете. Приведена структурная и функциональная схема предложенной реконфигурируемой системы управления.

Ключевые слова: система управления, отказная ситуация, управляемость и устойчивость самолета, реконфигурация управления, рулевые органы.

Possibility of application of system methods of reconfiguration controlling surfaces is in-process grounded for proceeding in controllability and stability of airplane in the conditions of unnormal situations. The flow and functional diagram of offered reconfiguration control system is in-process resulted.

Key words: control system, unnormal situation, stability and controllability of the airplane, reconfiguration, steering organ.

Постановка проблемы

Эффективность применения авиационной техники неразрывно связана с проблемой безопасности полетов, успешное решение которой в значительной мере определяет перспективы развития как гражданской, так и военной авиации. Сложность решения проблемы обеспечения безопасности полетов непрерывно возрастает в связи с повышением интенсивности использования авиационной техники и расширением круга выполняемых ею функциональных задач. Текущая полетная ситуация в первую очередь характеризуется: условиями и режимами полета, функциональным состоянием экипажа, работоспособностью авиационных систем, а также аэродинамическим состоянием несущих поверхностей самолета. В результате действия неблагоприятных внешних факторов и деградирующих внутренних процессов может возникнуть особая полетная ситуация, которая подразделяется на четыре вида: усложнение условий полета, сложная ситуация, аварийная ситуация, катастрофическая ситуация [1]. Чрезвычайно высокая быстротечность развития особой ситуации (ОС) требует мгновенного вмешательства в ситуацию для принятия решений направленных на восстановление управляемости и устойчивости самолета и тем самым предотвращение перехода текущей полетной ситуации в катастрофическую.

В данной работе класс возможных отказов ограничим отказами (повреждениями) рулевых органов и приводов. Этот класс отказов достаточно распространен: так, по данным работы [2], 20 %

случаев потери самолетов связаны с отказами повреждениями системы автоматического регулирования (САУ), причем главным образом с отказами и повреждениями приводов рулевых органов. В [3] отмечено, что главной причиной ненадежности САУ современных самолетов являются отказы рулевых органов. Отказы приводов и рулевых органов следует рассматривать вместе, так как они по результатам проявления часто неотличимы друг от друга: отказ привода обычно приводит к фиксации соответствующей управляющей поверхности в некотором положении, отказ управляющей поверхности (заклинивание) приводит к тому же результату; снижение вследствие отказа добротности привода, так же, как и уменьшение площади управляющей поверхности вследствие попадания механического или биологического объекта, приводит к снижению эффективности соответствующего органа управления.

Традиционным подходом является введение аппаратной избыточности (резервирование). Однако вследствие этого возрастают масса и габариты самолета, что ограничивает универсальность этого подхода. Поэтому возникает вопрос о поиске более эффективных методов предотвращения развития ОС в полете. Одним из таких направлений может стать развитие системных методов реконфигурации управляющих поверхностей для возобновления управляемости и устойчивости самолета в условиях внезапного возникновения ОС в полете.

Основная часть

Под реконфигурацией будем понимать перераспределение управления на исправные органы с целью создания необходимых управляющих сил и моментов для восстановления управляемости и устойчивости самолета в условиях возникновения ОС в полете. Осуществление реконфигурации возможно для самолетов, обладающих избыточными органами управления. Так, в [4, 5] отмечено, что для высокоманевренных самолетов с пониженной статической устойчивостью, выходящих на критические значения угла атаки, для успешной реконфигурации при отказах необходимо хотя бы восемь независимых органов управления. Современные и перспективные высокоманевренные самолеты обладают большим количеством органов управления: элевоны (секционные, с независимым управлением каждой секцией), переднее горизонтальное оперение, интерцепторы, рули направления, поворотные сопла и др.

В самолетах, разработанных ранее, отсутствуют средства автоматического осуществления реконфигурации, эти функции полностью возлагаются на летчика. Поэтому результат реконфигурации полностью зависит от опыта и умения летчика, хотя принципиально реконфигурация позволила бы предотвратить 70 % случаев авиационных происшествий из-за отказов приводов и рулевых органов (это заключение сделано в [2] на основании анализа летных происшествий, имевших место в США). В [6, 7] приводятся два примера удачного и неудачного осуществления летчиком реконфигурации. В первом случае левая секция руля высоты самолета Delta L1011 оказалась зажата в положении 19° при взлете. Летчику удалось предотвратить аварию путем реконфигурации. Второй случай связан с аварией самолета DC-10 25 мая 1979 г. в Чикаго, вызванной потерей секции предкрылка самолета. Последующее моделирование ситуации показало, что летчик мог бы избежать аварии с помощью реконфигурации [8].

В связи с этим целесообразно на этапе разработки САУ самолетов заложить в систему средства автоматического осуществления реконфигурации. В этом направлении в 1994 г. лабораторией динамики полета ВВС США была начата разработка программы SRFCS (Self-Repairing Flight Control System) [2], целью которой является осуществление при разработке САУ современных и перспективных самолетов мероприятий, обеспечивающих реконфигурацию и диагностирование системы и направленных в конечном счете на повышение надежности, отказоустойчивости, живучести. Программа SRFCS предусматривает два основных подхода [2]. Первый связан с автоматической реконфигурацией, второй – с построением экспертных диагностических систем, дающих рекомендации летчику в зависимости от сложившейся полетной ситуации.

Разработки методов и моделей реконфигурации управляющих воздействий на самолет в условиях возникновения особых ситуаций в полете посвящена работа [4]. Для реконфигурации управляющих воздействий при отказах приводов и органов управления используется два подхода

[4]: параметрическая и структурная. Параметрическая – изменение коэффициентов обратной связи исполнительных механизмов с учетом технического состояния самолета, для улучшения эффективности их функционирования. Структурная – перераспределение управления между исправными органами управления для восстановления приемлемых характеристик управляемости и устойчивости в условиях ОС в полете.

На рис. 1 представлена упрощенная структурная схема реконфигурируемой системы управления.

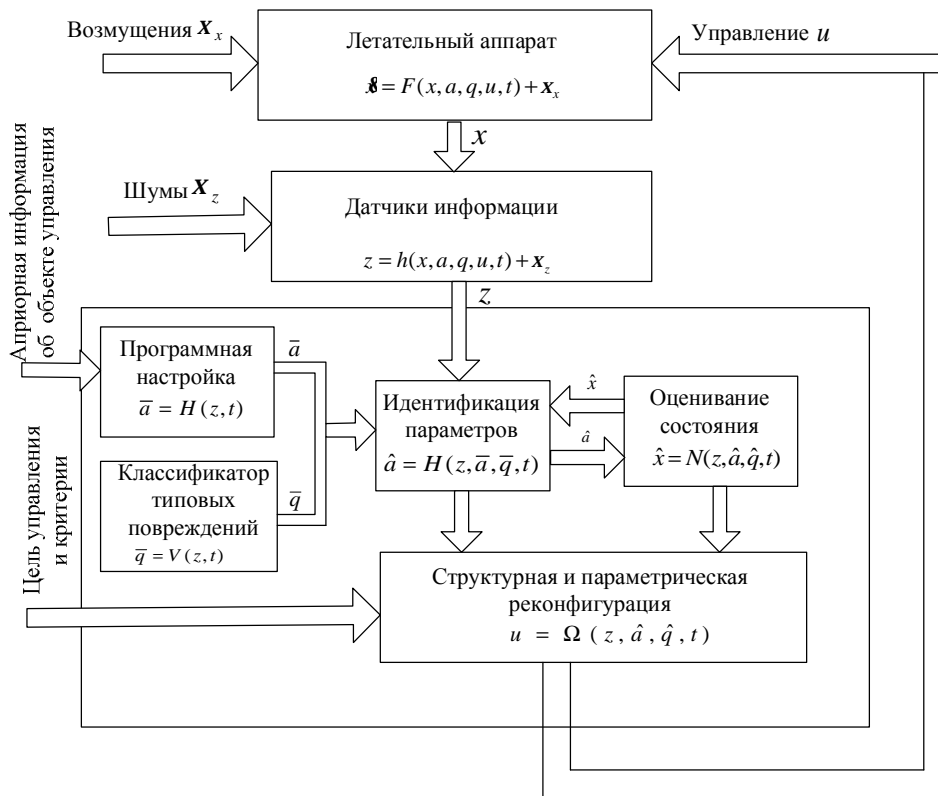


Рис. 1. Упрощенная структурная схема реконфигурируемой системы управления

Реконфигурируемая система управления рассматриваемого типа предполагает выполнение трех взаимосвязанных процедур:

- определение динамических характеристик самолета в процессе выполнения полета;
- оценивание технического состояния основного самолетного оборудования, а также целостности внешнего обвода самолета;
- формирование управляющих сигналов на исполнительные органы самолета с использованием информации, получаемой с помощью первых двух процедур.

Будем полагать, что движение самолета описывается дифференциальным уравнением:

$$\dot{x} = F(x, a, q, u, t) + x_x \quad (1)$$

где x – n -мерный вектор состояния объекта, определенный в пространстве X , a – r -мерный вектор параметров, принимающий значения из множества A и определяемый свойствами среды; q – вектор целостности внешнего обвода самолета в полете, учитывающий влияние типовых повреждений на аэродинамические свойства самолета, u – m -мерный вектор управляющих воздействий, формируемый реконфигурируемой системой управления и принадлежащий множеству U ; t – текущее время, принадлежащее отрезку $[t_0, t_k]$ на котором определена ОС в полете; x_x – n -мерный вектор неконтролируемых возмущений (шумы, помехи измерения и т. д.);

F – n -мерная векторная функция указанных аргументов, известная, по предположению, на основе теоретических и экспериментальных исследований.

Наблюдение за движением самолета осуществляется с помощью комплекса датчиков, измеряющих компоненты состояния объекта и управления, а также целостность его внешнего обвода в полете:

$$z = h(x, a, q, u, t) + x_z \quad (2)$$

где z – l -мерный вектор наблюдений в пространстве Z , x_z – l -мерный вектор аддитивных шумов, искажающих показания датчиков; h – l -мерная векторная функция указанных аргументов, известная на основе теоретических и экспериментальных исследований датчиков информации. Результаты измерений поступают в реконфигурируемую систему управления, где используются для определения динамических характеристик самолета и оптимального (субоптимального) оценивания его состояния.

Следующий этап функционирования предложенной реконфигурируемой системы управления является процесс параметрической идентификации динамических характеристик самолета в условиях возникновения ОС в полете, который в общем виде описывается оператором:

$$\hat{a} = H(z, \bar{a}, \bar{q}, t) \quad (3)$$

Таким образом, в рассматриваемой структуре полагается, что идентификация осуществляется в некоторой окрестности программного значения вектора параметров. В процессе идентификации реконфигурируемая система управления учитывает факторы, влияющие на динамические свойства самолета (неблагоприятные внешние факторы и деградирующие внутренние процессы).

На основе сигналов датчиков и оценок параметров объекта осуществляется оптимальное (или субоптимальное) оценивание состояния объекта, позволяющее в значительной степени повысить точность информации о векторе x :

$$\hat{x} = N(z, \hat{a}, \hat{q}, t) \quad (4)$$

где \hat{x} – n -мерный вектор оценки параметров вектора x .

Итоговой процедурой реконфигурируемой системы управления является оптимизация управляющих воздействий на исполнительные органы управления самолета на основе, задаваемых цели управления и критериев оптимизации для предотвращения развития ОС в полете. Оператор, описывающий формирование вектора оптимальных управлений, имеет вид:

$$u = \Omega(z, \hat{a}, \hat{q}, t) \quad (5)$$

Критерии оптимальности, сформированные заблаговременно, определяют меру, опираясь на которую алгоритм управления выбирает наиболее благоприятный путь достижения объектом заданного состояния $x_{зад}$. Структура оператора Ω зависит от способа задания цели управления, минимизируемых критериев и выбора метода оптимизации, располагаемым временем развития ОС в полете, а также аэродинамическим состоянием внешнего обвода самолета.

Выводы

Сформулируем основные выводы по данной статье.

1. Отказы приводов и органов управления приводят обычно к следующим последствиям: пропорциональное снижение эффективности рулевого органа; фиксация управляющего органа в некотором положении (максимальном, минимальном, нулевом), снижение эффективности соответствующего органа управления вследствие уменьшения площади управляющей поверхности в результате попадания механического или биологического объекта.

2. В условиях избыточности органов управления задача реконфигурации принципиально может быть решена.

3. Алгоритм создания предложенной реконфигурируемой системы управления содержит следующие этапы:

- формулирование критериев оптимальности;

- разработка математической модели самолета с учетом поправок, которые вносит в процесс управления ОС;
- синтез законов и моделей реконфигурации управления;
- разработка алгоритмов адаптации (настройки) законов управления по режимам функционирования пилотажно-навигационного оборудования самолета;
- реализация полученных законов и моделей реконфигурации с помощью управляющих БЦВМ.

1. *Нормы лётной годности гражданских самолётов СССР. (НЛГС – 3). Межведомственная комиссия по нормам лётной годности гражданских самолётов и вертолётот СССР. – 3-е изд. – 1984. – 464 с.* 2. *Eslinger R.A., Chandler P.R. Self-repairing flight control system program overview // Proc. IEEE National Aerospace and Electronics Conf. – 1988. –Vol. 2. – P. 504–511.* 3. *Ostroff A.J., Hueschen R.M. Investigation of control law reconfigurations to accommodate a control element failure on a commercial airplane // Proc. American Control Conf. – 1984. – Vol. 3. – P. 1746–1754.* 4. *Системні методи відновлення живучості літальних апаратів в особливих ситуаціях у польоті: монографія / В. М. Казак. – К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. – 284 с.* 5. *Глумов В.М., Земляков С.Д., Рутковский В.Ю., Силаев А.В. Алгоритмическое обеспечение отказоустойчивости систем автоматического управления // АИТ. 1988. – № 9. С. 3-33.* 6. *Napolitano M.R., Swaim R.L. A new technique for aircraft flight control reconfiguration // Proc. AIAA Guidance, Navigation and Control Conf. – 1989. – Pt. 1. – P. 1–9.* 7. *Bonnice W.F., Wagner E., Motyke P., Hall S. The application of the detection filter to aircraft control surface and actuator failure detection and isolation // Proc. AIAA Guidance, Navigation and Control Conf., 1985. – P. 732–740.* 8. *Земляков С.Д., Рутковский В.Ю., Силаев А.В. Реконфигурация систем управления летательными аппаратами при отказах // АИТ. – 1996. – N 10. – С. 3–20.*

PARALLEL METAHEURISTICS IN GRAPH COLORING

© Kokosiński Z., 2012

In this survey paper applications of parallel metaheuristics to solving graph coloring problems are described. The Graph Coloring Problem (GCP), Graph Coloring Sum Problem (GCSP) and Robust Graph Coloring Problem (RGCP) are known to be NP-complete. They do not have any polynomial algorithms. Therefore, a number of approximation, iterative and hybrid algorithms was developed for their solving. Recently a number of parallel algorithms was proposed for GCP and related coloring problems, including parallel metaheuristics like Parallel Genetic Algorithm (PGA), Parallel Tabu Search (PTS), Parallel Simulated Annealing (PSA) etc. DIMACS benchmarks as well as random graphs were used for their experimental verification. The results obtained for GCSP contributed to finding better lower and upper bounds on chromatic sum and chromatic sum number for many DIMACS graph instances, outperforming results known from the literature. The reported data support a conclusion, that parallel metaheuristics can be used efficiently for approximate solving of many graph coloring problems and for finding better upper bounds of many hard-to-compute graph parameters.

Key words: graph coloring, graph coloring sum, robust graph coloring, parallel metaheuristics, parallel iterative algorithms, chromatic sum, chromatic sum number.

Наведено огляд застосувань паралельних метаевристик для вирішення проблем колоризації графів. Проблеми колоризації графів (GCP), сумарної колоризації графів (GCSP) та робастної колоризації графів (RGCP) є NP-повними і не мають поліноміальних алгоритмів. З цієї причини для різних варіантів основної проблеми колоризації графів розроблено багато наближених алгоритмів, ітераційних і гібридних. Останнім часом для задачі колоризації графів і подібних їй проблем були розроблені паралельні алгоритми, зокрема паралельні метаевристики, зокрема паралельний алгоритм табу пошуку (PTS), паралельний генетичний алгоритм (PGA) і паралельний алгоритм імітації відпалу (PSA). В експериментальній перевірці алгоритмів використано графи зі сховищем DIMACS, а також випадкові графи. Дослідження застосування PGA для задач сумарної колоризації спричинило визначення нових верхніх і нижніх оцінок хроматичної суми і числа хроматичної суми для класу тестів з бази DIMACS, які є точнішими від відомих теоретичних оцінок. Отримані результати підтверджують думку, що паралельні метаевристики можуть стати потужним інструментом для наближеного розв'язування задач колоризації графів у практичних застосуваннях, а також для експериментального визначення верхньої оцінки обраних параметрів важко обчислювальних графів.

Ключові слова: колоризація графів, сумарна колоризація, робастна колоризація, паралельна метаевристика, паралельний ітераційний алгоритм, хроматична сума, число хроматичної суми.

Introduction

Graph coloring is a popular mathematical model for solving combinatorial optimization problems. In this survey paper applications of parallel iterative algorithms to graph coloring problems are described. Graph k -colorability problem (GCP) belongs to the class of NP-hard combinatorial problems [17]. The

graph coloring was the subjects of the Second DIMACS Implementation Challenge in 1993 [23] and Computational Symposium on Graph Coloring and Generalizations in 2002. Numerous classes of graph colorings were defined and characterized in [22, 28], among them classical vertex and edge colorings, distance coloring, on-line coloring, equitable coloring, sum coloring, T-coloring, harmonious coloring, circular coloring, list coloring, total coloring etc. Most graph coloring problems are NP-hard. Graph coloring is strongly related to scheduling problems [18, 36].

The sequential approximation algorithms for GCP are given in [28]. Parallel approximation algorithms for GCP adapted efficiently for modern multi-core multithreaded architectures are shown in [4].

Metaheuristics are versatile iterative algorithms capable of solving a wide class of optimization problems. Several known iterative algorithms were also applied to graph coloring problem [2, 5, 16, 20]. Hybrid algorithms combining known metaheuristics were also developed [37, 38, 40].

Graph benchmarks for GCP include:

- DIMACS benchmark suite – graphs recommended for DIMACS computing challenges; at present almost all graphs have known chromatic numbers $\chi(G)$
- random graphs (various generation schemes) – usually with unknown value of $\chi(G)$;
- R-MAT graphs – a new class of scalable parametrized random graph instances, defined recursively (virtually any possible sizes are available) – no $\chi(G)$ is known.

Parallel metaheuristics are complex methods combining at least two metaheuristics of the same type that search for a solution either independently or in cooperation (with information exchange). The first book devoted solely to parallel metaheuristics was published in 2005 by Alba [1]. At that time the only research paper on application of a parallel metaheuristic to graph coloring was [25]. Since then, many new results in that area were reported in the literature. Parallel metaheuristics used for several graph coloring problems are listed and characterized in the following sections. The new original results include time-efficient optimal or suboptimal coloring algorithms for the benchmark graphs and also new bounds found experimentally for such hard-to-compute graph parameters like chromatic number, chromatic sum and chromatic sum number. The present survey reports the latest results from this fast growing research area.

1. Vertex coloring

GCP is defined for an undirected graph $G(V,E)$ as an assignment of available colors $\{1, \dots, k\}$ to graph vertices providing that adjacent vertices receive different colors and the number of colors k is minimal. The resulting coloring is called conflict-free and k is called the graph chromatic number $\chi(G)$.

Models and properties of parallel genetic algorithms were described in [3]. The first population-based metaheuristic, i.e. PGA for solving the classical Graph Coloring Problem (GCP) in migration models was published by Kokosiński at all. [25]. Migration models of PGA consist of a finite set of disjoint populations that co-evolve and occasionally exchange genetic information under control of a migration operator. Populations are built of individuals of the same type and are ruled by the one adaptation function. In the paper new recombination operators: Sum-Product Partition Crossover (SPPX) and Conflict Elimination Crossover (CEX) were introduced and compared with UISX and GPX operators. In computer experiments DIMACS benchmark graphs were used [41–43]. An extended version of the paper was published in [26]. Another version of PGA for GCP was developed by Domagała [10]. In diffusion model of PGA the overlapping subpopulations are placed in the nodes of a mesh. The population size, the mesh size and architecture and other factors have to be considered in that model. The optimal tuning of the algorithm for the given problems can be achieved. An MPI implementation of PGA for GCP was described in [20].

In the following years many other parallel metaheuristics for GCP and related coloring problems were developed.

Three Parallel Tabu Search (PTS) algorithms (master-slave search, independent search, cooperative search) on the basis of TABUCOL for GCP (Herz, de Werra) was proposed by Dąbrowski [8]. They were implemented on two different high cluster architectures. DIMACS graphs and random graphs with various node densities are used for experiments. PTS for GCP outperforms popular greedy DSATUR heuristic. The obtained experimental results showed the limited applicability of PTS for GCP due to problem complexity. Another PTS algorithm for GCP was developed by Kirsz [24].

The first Parallel Simulated Annealing Algorithm (PSA) for GCP was proposed by Łukasik et al. [32]. In synchronous master-slave model with periodic solution update two basic techniques were used:

1. multiple threads for computing independent chains of solutions and exchanging the obtained results on a regular basis;
2. parallel moves, where single Markov chain is being evaluated by multiple processing units calculating possible moves from one state to another.

The paper contains recommendations for optimal parameters settings. A comparison of PSA to PGA metaheuristic was provided which is not favorable for any of those two methods.

A Parallel Immune System (PIS) for graph coloring was proposed by Dąbrowski [9]. The algorithm is based on the mechanism of a clonal selection. Every processor operates on its own pool of antibodies and a migration operator is used to allow processors to exchange information.

2. Minimum sum coloring

GCS is defined for an undirected graph $G(V,E)$ as an assignment of available colors $\{1, \dots, h\}$ to graph vertices providing that sum of all color numbers in a conflict-free coloring must be minimal. The minimum number of colors h in a minimum-sum coloring is called chromatic sum number $s(G)$, $s(G) \geq \chi(G)$.

The problem was introduced by Kubicka and Schwenk [29]. Theoretical properties and bounds can be found in [32]. New theoretical lower bounds were given in [27].

The best lower bounds on $\Sigma(G)$ are as follows:

$$\lceil \sqrt{8m} \rceil \leq \Sigma(G) \quad (1)$$

$$n + \chi(G)(\chi(G) - 1) / 2 \leq \Sigma(G) \quad (2)$$

The best upper bounds on $\Sigma(G)$ are the following:

$$\Sigma(G) \leq n + m \quad (3)$$

$$\Sigma(G) \leq n(\chi(G) + 1) / 2 \quad (4)$$

Table 1

MSCP – research results (part I)

$G(V,E)$	n	m	$\chi(G)$	$s(G)$	$\Sigma(G)$								
					theoretical bounds				experimental bounds				
					L.B.	rule	U.B.	rule	L.B.	source	U.B.	source	gap [%]
anna	138	493	11	11	193	2	631	3	272	[35]	277	[35]	1,81
david	87	406	11	11	142	2	494	3	234	[35]	237	[40]	1,27
huck	74	301	11	11	129	2	375	3	243	[35]	243	[27]	0
jean	80	254	10	10	125	2	334	3	216	[35]	217	[30]	0,46
queen5.5	25	160	5	5	36	1	75	4	75	[35]	75	[27]	0
queen6.6	36	290	7	7-8	57	2	144	4	126	[35]	138	[27]	8,70
queen7.7	49	476	7	7	70	2	196	4	196	[35]	196	[27]	0
queen8.8	64	728	9	9	100	2	320	4	288	[35]	291	[40]	1,03
games120	120	638	9	9	156	2	600	4	442	[35]	443	[40]	0,23
miles250	128	387	8	8	156	2	515	3	316	[35]	328	[40]	3,66
miles500	128	1170	20	20	318	2	1298	3	677	[35]	709	[40]	4,51
myciel3	11	20	4	4	17	2	27	4	16	[35]	21	[27]	19,05
myciel4	23	71	5	5	33	2	69	4	34	[35]	45	[27]	24,44
myciel5	47	236	6	6	62	2	164	4	70	[35]	93	[27]	24,73
myciel6	95	755	7	7	116	2	380	4	142	[35]	189	[27]	24,86
myciel7	191	2360	8	8	219	2	859	4	286	[35]	381	[30]	24,93

$$gap = [(\min(UB_{th}, UB_{exp}) - \max(LB_{th}, LB_{exp})) * 100\% / (\min(UB_{th}, UB_{exp}))]$$

An application of PGA to Graph Coloring Sum Problem (GCSP) was shown in Kokosiński and Kwarcianny [27]. That paper initialized the intensive international research on experimental finding of better upper bounds on chromatic sum and chromatic sum number for a class of DIMACS graph instances.

The recent methods and results are given in [11–15, 30, 34, 35]. Local search heuristic for GCSP was described in [21]. In [40] a new hybrid algorithm EXSCOL is presented which incorporates a TS metaheuristic for finding a vertex partition into large independent sets. EXSCOL was tested on a set of 52 DIMACS graphs. For 17 out of 28 previously examined problem instances better colorings and bounds were found (cf. Tables 1 and 2).

Table 2

MSCP – research results (part II)

$G(V,E)$	n	m	$\chi(G)$	$s(G)$	$\Sigma(G)$								
					theoretical bounds				experimental bounds				
					L.B.	rule	U.B.	rule	L.B.	source	U.B.	source	gap[%]
fpsol2.i.1	496	11654	65	65	2576	2	12150	4	2590	[14]	3405	[10]	23,9
inithx.i.1	864	18707	54	54	2295	2	19571	4	2801	[14]	3679	[10]	23,8
mug88-1	88	146	4	4	94	2	220	4	163	[14]	190	[10]	14,2
mug88-25	88	146	4	4	94	2	220	4	162	[14]	187	[10]	13,3
mug100-1	100	166	4	4	106	2	250	4	187	[14]	211	[10]	11,3
mug100-25	100	166	4	4	106	2	250	4	185	[14]	214	[10]	13,5
2-Inser 3	37	72	4	4	43	2	92	4	55	[14]	63	[10]	12,7
3-Inser 3	56	110	4	4	62	2	140	4	84	[14]	92	[10]	8,70
zeroin.i.2	211	3541	30	30	646	2	3270	4	1003	[14]	1013	[10]	0,99
zeroin.i.3	206	3540	30	30	641	2	3193	4	997	[14]	1007	[10]	0,99

$$gap = [(\min(UB_{th}, UB_{exp}) - \max(LB_{th}, LB_{exp})) * 100\% / (\min(UB_{th}, UB_{exp}))]$$

The research on MSCP brought the following results :

- The theoretical gap between L.B. and U.B. for MSCP was reduced;
- DIMACS benchmarks for GCP has been accepted for examination of MSCP;
- Parallel metaheuristic can be used for finding unknown hard-to-compute parameters of benchmark graphs like $\chi(G)$ and $s(G)$;
- The gap between L.B. and U.B. for most examined graph problems was significantly reduced.

3. Robust graph coloring

RGCP is defined for undirected graph $G(V,E)$ as an assignment of available colors $\{1, \dots, k\}$ to graph vertices, providing that

$$\forall (u, v) \in E : (p_{uv} = 1) \Rightarrow c(u) \neq c(v), \quad (5)$$

and

$$\sum_{((u, v) \in E) \wedge (c(u) \neq c(v)) \wedge (p_{uv} < 1) \wedge (u < v)} p_{uv} \geq T, \quad (6)$$

where p_{uv} – is an edge $e(u,v)$ weight in $[0, 1]$; p_{uv} may be considered as a probability of an edge existence, in the classical vertex coloring $p_{uv} \in \{0,1\}$; T – is an assumed threshold of robustness.

The robust graph coloring problem (RGCP) was introduced in [39]. Some applications of basic metaheuristics for RGCP was reported in [6,31]. They include uncertainty management, crew assignment problem and robust energy supply. Metaheuristics for RGCP [31] were tested for simple random graphs.

The first parallel metaheuristic – PGA for RGCP – was developed by Chrzyszcz [6]. In the algorithm the efficient Best Crossover (BCX) recombination operator [36] was used in addition to those tested in [25, 26]. For computer experiments parametrized problem instances were generated by a random modification of

selected DIMACS graphs. A given number of $p_{uv}=1$ in adjacency matrix representation was replaced with p_{uv} in $(0,1)$. No other parallel metaheuristic for RGCP is available for comparison.

Conclusions

PGA, PSA, PTS and PIS are well known parallel iterative algorithms that are often applied for solving approximate solution of NPO problems. Efficient use of parallel metaheuristics requires careful design of details, proper tuning of parallel scheme, hybridization etc. Parallel metaheuristics are well suited for solving graph coloring problems and new algorithms can still be designed for graph colorings, f.i. PACO. Parallel metaheuristic can also be used for computing unknown hard-to-compute parameters of benchmark graphs.

Few classes of graph colorings has been carefully tested with computers but research in this area is prospective. It seems that for many coloring problems DIMACS graphs will be gradually replaced by random problem instances, in particular R-MAT graphs [4]. On the other hand DIMACS benchmarks shall be of the first choice when we look for the test graphs with known parameters. They can be modified with a little effort according to the particular needs.

There are many open questions concerning parallelization of computations using new techniques and modern multi-core computers. In particular, parallel metaheuristics may be more difficult to implement on massively parallel architectures than dedicated approximation algorithms [4].

Acknowledgements

The paper is an extended version of the conference paper presented at the The First International Conference on Automatic Control and Information Technology *ICACIT'2011*, held in Lviv (Ukraine) on December 15-17, 2011. The work was supported in part by Cracow University of Technology, Kraków, Poland.

1. Alba E. (Ed.) *Parallel metaheuristics: a new class of algorithms*, John Wiley & Sons. – 2005.
2. Bouziri H., Jouini M. *A tabu search approach for the sum coloring problem // Electronic Notes in Discrete Mathematics*. – Vol. 36. – 2010. – P. 915–922.
3. Cantú-Paz E. *Efficient and accurate parallel genetic algorithms*, Kluwer. – 2000.
4. Catalyurek U., Feo J., Gebremedhin A., Halappanavar M., Pothen A. *Graph coloring algorithms for multi-core and massively multithreaded architectures // Parallel Computing*. – 2012. (accepted)
5. Costa D., Hertz A. *Ants can color graphs // J. of the Operational Research Society*. – Vol. 48. – 1997. – P. 295–305.
6. Chrzyszcz G. *Parallel evolutionary algorithm for the design of robust scheduling in power systems*, M.Sc. thesis, Politechnika Krakowska. – 2009. (in Polish).
7. Crainic T.G., Nourredine, H. *Parallel meta-heuristics applications*, [in:] Alba E. (Ed.): *Parallel metaheuristics: a new class of algorithms*, John Wiley & Sons. – 2005.
8. Dąbrowski J. *Parallelization techniques for tabu search*, *Proc. 8th Int. Conference on Applied parallel computing: state of the art in scientific computing*. – 2007.
9. Dąbrowski J. *Parallel immune system for graph coloring // Lecture Notes in Computer Science*. – Vol. 4910. – 2008. – P. 497–505.
10. Domagała P. : *A paralel genetic algorithm for graph coloring in a diffusion model*, M.Sc. thesis, Politechnika Krakowska. – 2006. (in Polish).
11. Douiri M., El Bernoussi S. *New algorithm for the sum coloring problem // Int. J. Contemp. Math. Sciences*. – Vol. 6. – 2011. – No.10. – P.453 – 463.
12. Douiri M., El Bernoussi S. *A new heuristic for the sum coloring problem // Applied Math. Sciences*. – Vol. 5. – 2011. – No.63 – P. 3121–3129.
13. Douiri M., El Bernoussi S. *A hybrid algorithm for the sum coloring problem*, *Proc. 2011 International Conference on Multimedia Computing and Systems (ICMCS)*, 7–9 April 2011, Ouarzazate, Morocco, P. 1–4.
14. Douiri S.M., El Bernoussi S. *An ant colony optimization for the sum coloring problem // International Journal of Applied Mathematics & Statistics*. – Vol. 27. – 2011. – No.3. – P. 102–110.
15. Douiri S.M., El Bernoussi S. *An new ant colony optimization algorithm for the lower bound of sum coloring problem // Int. J. Math. Modeling and Algorithms*, Springer. (DOI: 10.1007/s10852-012-9172-x).
16. Erfani M., Ghanizadeh A., Abarghouei A.A., Sinaie S., Shamsuddin S.M. *A Modified PSO method enhanced with fuzzy inference system for solving the planar graph coloring problem*, *WORLDCOMP'10, Proceedings of the 2010 International Conference on Artificial Intelligence, IC-AI, July 12-15, Las Vegas, Nevada, CSREA Press*. –

2010. – P.160 – 165. 17. Garey R., Johnson D.S. *Computers and intractability. A guide to the theory of NP-completeness*, Freeman, San Francisco. – 1979. 18. Giaro K. *Task scheduling by graph coloring // Gdańsk University of Technology, Monographs. – Vol. 37. – 2003. (in Polish)*. 19. Hadjikyriacou E., Samaras N., Margaritis K. *An experimental evaluation of a parallel genetic algorithm using MPI, Proc. 13th Panhellenic Conference on Informatics, Corfu, Greece, Computer Society Press. – 2009. – P. 75–79*. 20. Han L., Han Z. *A novel bi-objective genetic algorithm for the graph coloring problem, Int. Conf. on Computer Modeling and Simulation ICCMS'2010, January 22–24, Sanya, China, 2010, P. 3–6*. 21. Helmar A., Chiarandini M. *Local search heuristic for chromatic sum, Proc. of The Metaheuristics Int Conference, MIC 2011, July 25–28, Udine, Italy*. 22. Jensen T.R., Toft B. *Graph coloring problems*, Wiley-Interscience. – 1995. 23. Johnson D.S., Trick M.A. *Cliques, coloring and satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discr. Math. and Theor. Comp. Sc. – Vol. 26. – 1996*. 24. Kirsz K. *Parallel tabu search algorithm in application to graph coloring problem, M.Sc. thesis, Politechnika Krakowska. – 2011. (in Polish)*. 25. Kokosiński Z., Kołodziej M., Kwarciany K. *Parallel genetic algorithm for graph coloring problem // Lecture Notes in Computer Science. – Vol. 3036. – 2004. – P. 217–224*. 26. Kokosiński Z., Kołodziej M., Kwarciany K. *Efficient graph coloring with parallel genetic algorithms // Computing and Informatics. – Vol. 24. – 2005. – P. 123–147*. 27. Kokosiński Z., Kwarciany K. *On sum coloring of graphs with parallel genetic algorithms // Lecture Notes in Computer Science. – Vol. 4431. – 2007. – P. 211–219*. 28. Kubale M. (Ed.) *Discrete optimization. Models and methods for graph coloring*, WNT, Warszawa. – 2002. (in Polish). 29. Kubicka E., Schwenk A.J. *An introduction to chromatic sums, Proc. 17th Annual ACM Computer Science Conf. – 1989. – P. 39–45*. 30. Li Y., Lucet C., Moukrim A., Sghiouer K. *Greedy algorithms for the minimum sum coloring problem, Proc. Int. Workshop “Logistique et transports”, March 22-24, Sousse, Tunisia, – 2009*. 31. Lim A., Wang F. *Meta-heuristic for robust graph coloring problem, IEEE Int. Conference on Tools with Artificial Intelligence, ICTAI. – 2004*. 32. Łukasik S., Kokosiński Z., Świętoń G. *Parallel simulated annealing algorithm for graph coloring problem, Lecture Notes in Computer Science. – Vol. 4967. – 2008. – P. 229–238*. 33. Malafiejski M. *Sum coloring of graphs, [in:] Kubale M. (Ed.): Graph colorings. American Math. Society. Contemporary Mathematics. – Vol. 352. – 2004. – P.55–65*. 34. Mohamed D.S., El Bernoussi S. *Max-Min ant system for the sum coloring problem, Proc. International Conference on Communications, Computing and Control Applications, CCCA 2011, March 3–5, Hammamet, Tunisia. – P. 1–4*. 35. Moukrim A., Sghiouer K., Lucet C., Li Y. *Lower bounds for the minimal sum coloring problem // Electronic Notes in Discrete Mathematics. – Vol. 36. – 2010. – P. 663–670*. 36. Myszkowski P.B.: *Solving scheduling problems by evolutionary algorithms for graph coloring problem, [in:] Xhafa F., Abraham A. (Eds.) Metaheuristics for Scheduling in Industrial and Manufacturing Applications. Studies in Computational Intelligence. – Vol.128. – 2008. – P. 145– 167*. 37. Pahlavani A., Eshghi K. *A hybrid algorithm of simulated annealing and tabu search for graph colouring problem // International Journal of Operational Research. – Vol. 11. – 2011. – No.2. – P. 136–159*. 38. Salari E., Ehgshi K. *An ACO algorithm for the graph coloring problem // Int. J. Contemp. Math. Sciences. – Vol. 3. – 2008. – No.6. – P. 293–304*. 39. Yáñez J., Ramírez J. *The robust coloring problem // European Journal of Operational Research. – Vol. 148. – 2003. – No. 3. – P. 546–558*. 40. Wu Q., Hao J.-K. *An effective heuristic algorithm for sum coloring of graphs // Computers and Operations Research. – Vol. 39. – 2012. – No.7*. 41. COLOR web site, <http://mat.gsia.cmu.edu/COLOR/instances.html>. 42. DIMACS ftp site, <ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/>. 43. COLORING 3 web site, <http://mat.gsia.cmu.edu/COLORING03/>.

Я. Драган¹, В. Овсяк², О. Овсяк³¹Національний університет "Львівська політехніка",
кафедра програмного забезпечення²Українська академія друкарства,³Київський національний університет культури і мистецтв, Львівська філія

МЕТОДОЛОГІЯ СИНТЕЗУ МОДЕЛЕЙ АЛГОРИТМІЧНОЇ СКЛАДОВОЇ АВТОМАТІВ

© Драган Я., Овсяк В., Овсяк О., 2012

Описано етапи синтезу, оптимізації і дослідження математичних моделей алгоритмічної складової автоматів методом алгебри алгоритмів, декомпозиції і математичної індукції.

Ключові слова: метод, методологія, модель, інформаційна технологія, декомпозиція, секвентуння, елімінування, паралелення, функційний унітерм, унітерм.

There is describe by the stages of synthesis, optimization and study mathematical models of machine algorithmic component by the methods of algorithm algebra, decomposition and mathematical induction..

Key words: method, methodology, model, information technology, decomposition, sequentization, elimination, parallelization, function uniterm, uniterm.

Вступ і формулювання задачі

Розвиток технологій і практична необхідність виконання великих розрахункових робіт під час конструювання атомної бомби, приховування і перехоплення відомостей (криптографія), автоматизації керування вогнем протиавіаційного захисту сприяли започаткуванню кібернетично-інформаційної доби [1]. В основі її є концепція автомата, сформульована у працях Мак-Калок – Пітса [2, 3]. Формально автомат (A) є п'ятіркою величин (V, S, W, φ, ψ), де V – дії (впливи), S – стани, W – реакції автомата, φ – функція переходів, яка є відображенням (→) декартового добутку (V×S), що записується як $V \times S \rightarrow S$, ψ – функція виходів як відображення $S \times V \rightarrow W$. Алгебра автомата є триосновною; основи її утворюють множини V, S і W.

В.Глушков створив абстрактну модель електронної обчислювальної машини у вигляді двох автоматів, а власне керуючого і операційного з двосторонніми зв'язками між ними [4]. Для побудови математичної моделі автомата В.Глушков розробив систему алгоритмічних алгебр [4]. Утворена вона двома алгебрами, а саме алгеброю операторів та алгеброю умов. Алгебра операторів складається з операторів композиції (множення (*)) – послідовного застосування операторів), α-диз'юнкції (альтернативи), диз'юнкції, фільтра і циклу. Алгебра умов містить узагальнені на тризначний алфавіт операції булевої алгебри і запроваджену нову операцію прогнозування (лівого множення умови на оператор). Г.Цейтлін модифікував систему алгоритмічних алгебр В.Глушкова, зокрема введені операції контрольної точки і синхронізування [5].

Використання двох алгебр (операторів і умов) ускладнює синтез автоматів. Крім того, операція композиції є асоціативною (див. с. 126 [4]). Однак в автоматах послідовне застосування операторів (властивість асоціативності) може мати місце тільки у часткових випадках. Для того, щоб переконатися у цьому, розглянемо тривіальний приклад. Припустимо, що маємо три змінні: $x=0$, $y=0$ і $z=0$, над якими виконуються такі дії: $x=5+5$, $y=x:2$ та $z=y-1$. Якщо спочатку виконати ці дії у такій послідовності: $((x=5+5)*(y=x:2))*(z=y-1)$, а потім – у такій: $(x=5+5)*((y=x:2))*(z=y-1)$, то отримаємо значення, які відрізняються.

Відома одноосновна алгебра алгоритмів [6] та її модифікація [7] з неасоціативною операцією секвентування, засобами яких виконується опис автоматів.

Автомати можуть бути реалізовані апаратно або апаратно-програмно. Найчастіше типовою реалізацією є якраз апаратно-програмна. До того ж досягнуті успіхи інтегральних напівпровідникових технологій забезпечують належний рівень функціонування апаратних засобів автоматів. Проблема-тичнішим є створення алгоритмічного і програмного забезпечення автоматів. Тому нижче розглянуто методологію застосування алгебри алгоритмів для побудови математичних моделей програмних складових автоматів, а власне інструментальних засобів комп'ютерних систем. Ця методологія таким самим способом може бути також використана і для синтезу апаратних складових автоматів.

Моделювання інформаційних технологій і систем

Основою інформаційних технологій є теорія автоматів, а з часів В.Глушкова утвердилась концепція, що мовою її є відповідна версія сучасної алгебри. Для неї ж характерне те, що в центрі уваги стають властивості операцій, а не об'єкти, над якими їх виконують, що своєю чергою призвело до поняття алгебричної структури та алгебричної системи як множини з означеною на ній сукупністю операцій і стосунків, які сформувались у середині ХХ ст. Одним з основних напрямків цієї теорії є вивчення стосунків між загальними властивостями класів алгебричних систем та синтаксичними властивостями мови, якою означають ці структури [3].

Оскільки апаратом теорії алгоритмів є різного роду варіанти алгебр, то природно скористатись для розроблення відповідних варіантів теорії так званим аксіоматичним методом. Хоч при запровадженні його в основи математики надії Д.Гільберта, що він може стати підставою всієї математики – усіх її розділів, не оправдались, він може бути корисним (з належною модифікацією) сьогодні. Для того пригадаймо, що аксіоматичний метод – це спосіб побудови наукової теорії (головно в математиці), коли в основу побудови покладено певні твердження, звані аксіомами, а всі решти факти отримують як логічні висліди з аксіом (дедукція). І тоді постають не тільки проблеми – звідки взяти аксіоми, але й проблеми несуперечності, повноти і незалежності цих аксіом. А як відомо з концепцій К.Геделя неповноти аксіоматичних теорій аксіоматичний метод не може бути взятий за остаточну основу обґрунтування теорії, проте він може стати засобом обґрунтування від супротивного несуперечності й повноти системи аксіом, коли доводять не саму теорему, а рівносильну їй протилежну до зворотної та коли вислід змінюють на супротивний і показують суперечливість засновкові (*reduction ad absurdum* – зведення до абсурду), або ж використовують поняття моделі, чи інакше, інтерпретації, коли кожному початковому поняттю й стосункові цієї тестованої теорії ставиться у відповідність певний поля інтерпретації, яка теж має свою теорію, що може бути аксіоматичною й дає підстави установаження відносних несуперечності, повноти і незалежності [3, 8].

Якщо можна побудувати модель, як її трактують у математичній логіці, тобто об'єкт, на якому реалізуються аксіоми системи, то цю систему *post factum* трактують, що вона має названі властивості. Історично подібну функцію виконала в обґрунтуванні геометрії Лобачевського-Бойяї інтерпретація А.Пуанкаре, реалізуючи аксіоми геометрії площини Лобачевського (гіперболічної геометрії) на одиничному крузі $E=\{z:|z|\leq 1\}$ площини комплексної змінної. Ф. Кляйн та А.Пуанкаре установили несуперечність неевклідової геометрії щодо евклідової, а несуперечність її Д. Гільберт звів до несуперечності арифметики [3, 8].

Наведені факти підтверджують тезу, що є два принципово різні тлумачення терміну математична модель: 1) це об'єкт відповідного розділу математики, у структурі якого в стислій конструктивній формі втілені істотні для розв'язування цього класу задач властивості досліджуваних об'єктів [9] – так прийнято у математиці, а відтак, у всіх технічних науках і 2) це вже описана інтерпретація реалізацій аксіом, як прийнято в логіці та основах математики. Наведене підтверджує, що перше тлумачення стосується якісних властивостей досліджуваних об'єктів, а друге – формальних їхніх властивостей.

Інформаційні технології – це процеси відбору, накопичення, передавання та опрацювання інформації для створення автоматизованих систем різноманітного призначення. Сучасні автоматизовані системи, зазвичай включають інструментальні засоби і є складними системами. Для

проектування інформаційних систем предметних галузей широко використовуються методи аналітичного, імітаційного, інфологічного, об'єктно-орієнтованого моделювання. Метод математичного моделювання, порівняно з іншими методами, має такі переваги, як забезпечення можливості точного опису, проведення за вибраними критеріями оптимізації і дослідження моделей. Математичне моделювання інформаційних технологій і систем полягає у побудові і дослідженні їхніх математичних моделей [10]. Синтез і оптимізація математичних моделей інформаційних технологій і систем може бути виконаним на підставі використання модифікованої алгебри алгоритмів [7].

Етапи синтезу моделей

Проектування інструментальних засобів інформаційних технологій виконується на трьох етапах з використанням методів декомпозиції та алгебри алгоритмів. З метою зменшення складності проектування і забезпечення можливості розпаралелення синтезу складних систем на першому етапі виконується структурна декомпозиція системи інструментальних засобів на підсистеми. Структурною декомпозицією визначаються складові і підпорядкованість складових моделі інструментальних засобів інформаційних технологій. Засобами алгебри алгоритмів створюється модель декомпозиції, яка на підставі властивостей операцій алгебри алгоритмів оптимізується за кількістю підсистем кожного із етапів декомпозиції. На другому етапі засобами алгебри алгоритмів синтезується і оптимізується модель функціонування інструментальних засобів. Третій етап полягає у дослідженні, реалізації та апробації математичної моделі інструментальних засобів. У разі виявлення помилок у модель та її реалізацію вносяться необхідні правки.

Декомпозиція структури автоматів. Зазвичай програмно реалізовані автомати утворені двома складовими. Передусім це інтерфейс користувача, яким здійснюється зв'язок користувача з комп'ютерною системою і використовуються її можливості. Другою складовою є функціональна частина, яка, на відміну від першої складової, є невидимою, але саме вона описує розв'язання тих задач, які візуалізовані в інтерфейсі користувача інструментальних засобів. Уже на самому початку синтезу інструментальних засобів, за функціональним призначенням, систему (S) декомпоновано на графічну (G) і функціональну (F) підсистеми. Ця декомпозиція засобами алгебри алгоритмів описується двома виразами. Один із них використовує операцію секвентування ($\overline{}$ – знак операції секвентування)

$$S = \overline{G, F},$$

яка виражає необхідну наявність обох підсистем і використовує знак коми для розділення, що свідчить про відсутність їхнього впорядкування. За потреби відображення послідовності підсистем використовується як розділювач “крапка з комою”. У загальному випадку можна було б задати розділювач “дві крапки”, який є ідентифікатором коми і крапки з комою. Друга модель декомпозиції найвищого рівня записується з використанням операції паралелення (\sqcap – знак операції паралелення)

$$S = G \sqcap F,$$

яка виражає незалежну наявність обох підсистем.

Кожна із підсистем, за вибраними критеріями (наприклад, функціональним призначенням), своєю чергою, може теж бути декомпонована на субсистеми. Враховуючи задачі, потреби і можливості, кількість рівнів декомпозиції визначає розробник інструментальних засобів. Формула (1) ілюструє можливість алгебри алгоритмів описати багаторівневу декомпозицію систем. У цій формулі підсистему-унітерм F декомпоновано на складові (підунітерми), які позначимо як P₀, P₁, ..., P_{p-1}. Очевидно, що в F входять усі p унітермів від P₀ до P_{p-1}. Між F і унітермами, на які він декомпонований та описаний засобами алгебри алгоритмів, ставивиться знак рівності.

По декомпозиції поточного рівня виконуємо декомпозицію чергового рівня, яка полягає у розбитті унітермів P₀, P₁, ..., P_{p-1} на підунітерми. Наприклад, унітерм P₀ декомпоновано на унітерми, які позначимо як Q₀, Q₁, ..., Q_{q-1}, а унітерм P_{p-1} – на унітерми R₀, R₁, ..., R_{r-1}. Аналогічно розбиваємо всі решта унітермів. Кожному із унітермів P₀, P₁, ..., P_{p-1} приписуємо

паралелення його "підунітермів" з відповідними розділювачами. Серед унітермів P_0, P_1, \dots, P_{p-1} можливі такі унітерми, декомпозицію яких недоцільно виконувати. Q_0, Q_1, \dots, Q_{q-1} – унітерми першого рівня декомпозиції унітерму P_0 , а q – кількість унітермів; S_0, S_1, \dots, S_{s-1} – унітерми передостаннього рівня декомпозиції унітерму Q_0 , а s – кількість унітермів; U_0, U_1, \dots, U_{u-1} – унітерми останнього рівня декомпозиції (функціональні унітерми) унітерму S_0 , а u – кількість унітермів. Останнім рівнем декомпозиції є декомпонування функціональних унітермів на предметні змінні та оператори-унітерми. Аналогічно описується декомпозиція усіх решта унітермів. Декомпозицією системи інструментальних засобів на підсистеми досягається зменшення складності створення підсистем.

$$F = \begin{bmatrix} P_0 = \begin{bmatrix} Q_0 = \dots \begin{bmatrix} S_0 = \overline{U_0: U_1: \dots U_{u-1}} \\ \vdots \\ S_1 = \overline{L_0: L_1: \dots L_{l-1}} \\ \dots \\ S_{s-1} = \overline{K_0: K_1: \dots K_{k-1}} \end{bmatrix} \\ \vdots \\ Q_1 = \dots \begin{bmatrix} E_0 = \overline{A_0: A_1: \dots A_{a-1}} \\ \vdots \\ E_1 = \overline{H_0: H_1: \dots H_{h-1}} \\ \dots \\ E_{s-1} = \overline{B_0: B_1: \dots B_{b-1}} \end{bmatrix} \\ \vdots \\ Q_{q-1} = \dots \begin{bmatrix} D_0 = \overline{C_0: C_1: \dots C_{c-1}} \\ \vdots \\ D_1 = \overline{G_0: G_1: \dots G_{g-1}} \\ \dots \\ D_{s-1} = \overline{V_0: V_1: \dots V_{v-1}} \end{bmatrix} \\ \dots \\ \vdots \\ P_{p-1} = \begin{bmatrix} R_0 = \dots \begin{bmatrix} O_0 = \overline{\Delta_0: \Delta_1: \dots \Delta_{s-1}} \\ \vdots \\ O_1 = \overline{T_0: T_1: \dots T_{t-1}} \\ \dots \\ O_{s-1} = \overline{F_0: F_1: \dots F_{f-1}} \end{bmatrix} \\ \vdots \\ R_1 = \dots \begin{bmatrix} M_0 = \overline{I_0: I_1: \dots I_{i-1}} \\ \vdots \\ M_1 = \overline{J_0: J_1: \dots J_{j-1}} \\ \dots \\ M_{m-1} = \overline{W_0: W_1: \dots W_{w-1}} \end{bmatrix} \\ \vdots \\ R_{p-1} = \dots \begin{bmatrix} Z_0 = \overline{N_0: N_1: \dots N_{n-1}} \\ \vdots \\ Z_1 = \overline{Y_0: Y_1: \dots Y_{y-1}} \\ \dots \\ Z_{z-1} = \overline{X_0: X_1: \dots X_{x-1}} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad (1)$$

Декомпозиція функціонування автоматів. Функціональна декомпозиція утворена унітермами-змінними і функціональними унітермами. Інформаційно-технологічні процеси починаються з певної вихідної точки. Нею, наприклад, можуть бути ввімкнення системи чи запуск на виконання. У будь-якому разі має бути виконана певна умова, яку позначимо u , початку функціонування. Якщо умова не виконана, то відбувається очікування її виконання, що запишемо поверненням у цикл (cu) за умовою u . Коли ж умова виконана, то, наприклад, функціонування починається з унітерму Z . У вигляді формули процес очікування виконання умови опишемо формулою [11]

$$\frac{cu}{Z; c_w; u-?}$$

Зазвичай, Z задаються початкові значення.

Типові інструментальні засоби мають інтерфейс користувача. Складові інтерфейсу (%Ai, %Bj, %Ck, ...) вибираються (ai, bj, ck, ... – умови вибору відповідної складової інтерфейсу) користувачем системи інструментальних засобів, що опишемо формулою

$$Y_r = \overbrace{\overbrace{\overbrace{\%A_i; *; a_i-?} \cdot \%B_j; *; b_j-?} \cdot \%C_k; *; c_k-?} \cdot \dots ,$$

у якій знак % є ідентифікатором складової інтерфейсу, а знаком операції паралелення з розділювачом унітермів комою описано можливість вибору будь-якої графічної складової.

Кожна типова графічна складова %Ai, %Bj, %Ck, ... утворена своїми властивостями (Wi, Wi, Wk) і назвою функціонального унітерму, який запускається на виконання у разі вибору графічної складової. Наприклад,

$$\%A_i = \overbrace{W_i A_i}.$$

Враховуючи все вищеописане, загальна формула функціональної декомпозиції опишеться таким виразом:

$$\overbrace{\overbrace{\overbrace{\%A_i; *; a_i-?} \cdot \%B_j; *; b_j-?} \cdot \%C_k; *; c_k-?} \cdot \dots$$

Отож алгеброю алгоритмів описуються інтерфейси користувачів і функціонування систем інструментальних засобів інформаційних технологій.

Оптимізація математичних моделей автоматів. Можливості, які забезпечує алгебра алгоритмів для оптимізації моделей автоматів, проілюструємо таким твердженням:

Теорема. З формули алгоритму

$$\overbrace{\overbrace{\overbrace{\left(\begin{matrix} A \\ ; \\ X \\ ; \\ B \end{matrix} \right) ; \left(\begin{matrix} C \\ ; \\ X \\ ; \\ D \end{matrix} \right) ; u-?}} \cdot \dots} \quad (2)$$

введенням додаткової змінної p з приписуванням їй значень 0 і 1, від якої не залежать унітерми формули, виводиться вираз

$$\overbrace{\overbrace{\overbrace{\left(\begin{matrix} A \\ ; \\ p=0 \left(\begin{matrix} C \\ ; \\ p=1 \end{matrix} \right) \\ ; \\ X \\ ; \\ B; D; (p=0)-? \end{matrix} \right) ; \dots}} \cdot \dots} \quad (2a)$$

з меншою кількістю унітермів X.

Доведення. Якщо умова u виконується, то на основі означення операції елімінування [5, 6] виконуємо такі перетворення:

$$\overbrace{\overbrace{\overbrace{\left(\begin{matrix} A \\ ; \\ p=0 \left(\begin{matrix} C \\ ; \\ p=1 \end{matrix} \right) \\ ; \\ X \\ ; \\ B; D; (p=0)-? \end{matrix} \right) ; \dots}} \cdot \dots} = \overbrace{\overbrace{\overbrace{\left(\begin{matrix} A \\ ; \\ p=0 \\ ; \\ X \\ ; \\ B; D; (p=0)-? \end{matrix} \right) ; \dots}} \cdot \dots} = \overbrace{\overbrace{\overbrace{\left(\begin{matrix} A \\ ; \\ p=0 \\ ; \\ X \\ ; \\ B \end{matrix} \right) ; \dots}} \cdot \dots} \quad (3)$$

Друга рівність має унітерм В, який отриманий із елімінування попередньої рівності за умовою $(p=0)-?$, що виконується.

Якщо ж умова u не виконується, то отримуємо аналогічні рівності

$$\left(\begin{array}{c} \overline{A; C; u-?} \\ , \\ p=0 \quad p=1 \\ ; \\ X \\ ; \\ \overline{B; D; (p=0)-?} \end{array} \right) = \left(\begin{array}{c} C \\ , \\ p=1 \\ ; \\ X \\ ; \\ \overline{B; D; (p=0)-?} \end{array} \right) = \left(\begin{array}{c} C \\ , \\ p=1 \\ ; \\ X \\ ; \\ D \end{array} \right) \quad (4)$$

У такому разі умова $(p=0)-?$ елімінування передостанньої рівності не виконується, тому з цього елімінування отримується унітерм D.

З формули (2) у разі виконання умови u на підставі означення операції елімінування [4, 5] виводимо вираз

$$\left(\begin{array}{c} A \\ ; \\ X \\ ; \\ B \end{array} \right) \quad (5)$$

Коли ж умова u не виконується, то аналогічно отримуємо вираз

$$\left(\begin{array}{c} C \\ ; \\ X \\ ; \\ D \end{array} \right) \quad (6)$$

У виразі (3) значення змінної p не впливає на значення усіх решта її унітермів (A, X і B), тому з точністю до значень унітермів формули (3) і (5) є однакові. Аналогічно є однаковими формули (4) і (6). Отже, введення змінної p , яка не змінює значень унітермів формули (2), не змінює значення алгоритму. Теорему доведено.

Як це проілюстровано теоремою, введенням додаткової змінної p формула (2) оптимізується до формули (2а), у якій вже відсутнє дублювання змінної X. В алгоритмах інструментальних засобів та інформаційних технологій замість змінної X можуть бути десятки і більше функціональних унітермів. Застосування алгебри алгоритмів забезпечує виконання оптимізації алгоритмів без зміни їхньої функціональності.

Дослідження математичних моделей автоматів. Метод дослідження математичних моделей автоматів базується на застосуванні повної і трансфінітної математичної індукції. Для виконання дослідження необхідним є задання секвентних областей значень (впорядкованих множин значень) усіх унітермів-змінних і функціональних унітермів. Доконечним є врахування впорядкованості унітермів-змінних і функціональних унітермів. Дослідження виконується для початкових, кінцевих та одного із будь-яких проміжних значень усіх унітермів-змінних і функціональних унітермів. Виконаними дослідженнями алгоритмів установлюється забезпеченість ними очікуваних результатів для усіх заданих значень змінних і функціональних унітермів.

Висновки

1. Методологія синтезу алгоритмічного забезпечення автоматів передбачає використання методів декомпозиції, алгебри алгоритмів і повної та трансфінітної математичної індукції.
2. Алгебра алгоритмів використовується на етапах структурного і функціонального синтезу, оптимізації і дослідження математичних моделей автоматів.
3. Засобами одноосновної алгебри алгоритмів описуються математичні моделі автоматів.

1. Драган Я.П., Медиковський М.О., Овсяк В.К., Сікора Л.С., Яворський Б.І. Системний аналіз проблем управління та кібернетики як підстави інформаційних технологій // Праці конференції

Обчислювальні методи і системи перетворення інформації, 4–5 жовтня, 2012. 2. Автоматы. Сб. статей / Под ред. К.Э. Шенона, Дж. Маккарти, перев. с англ. под ред. А.А. Ляпунова. – М.: ИИЛ, 1956. – 604 с. 3. Математика. Большой энциклопедический словарь / Гл. ред. Ю.В. Прохоров, 3-е изд. – М.: Большая Рос. энциклопедия, 1998. – 848 с. 4. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – 2-е изд. перераб. – Киев: «Наукова думка», 1978. – 318 с. 5. Цейтлин Г.Е. Введение в алгоритмику. – К.: Сфера, 1998. – 310 с. 6. Овсяк В.К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем / Овсяк В.К. // Доповіді Національної академії наук України. – 1996. – № 9. – С. 83–89. 7. Owsiak W., Owsiak A. Rozszerzenie algebry algorytmów // Pomiar, automatyka, kontrola. – 2010. – № 2. – S. 184–188. 8. Математическая энциклопедия. – Т.4. – М.: Советская энциклопедия, 1982. – 1216 стб. 9. Драган Я., Овсяк В., Сікора Л. Самоорганізація – принцип еволюції складних систем // Дух, наука, думка, воля поступу української економіки: Зб. праць. / Упоряд. І.В. Барановський. – Львів: Укр. акад. друк. 2001. – 64 с. – С. 29–31. 10. Математическая энциклопедия. – Т. 3. – М.: Советская энциклопедия, 1982. – 1184 стб. 11. Овсяк О.В. Рекурентно-декомпозиційна методологія інформаційних технологій і систем // Поліграфія і видавнича справа. – 2011. – № 3 (55). – С. 74–84.

УДК 621.372:538.56

В. Передерій¹, С. Бабичев², В. Литвиненко^{1,2}

¹Миколаївський національний університет ім. О.М. Сухомлинського,
кафедра комп'ютерних систем та мереж

²Херсонський національний технічний університет,
кафедра інформаційних технологій та комп'ютерних систем

ЗАСТОСУВАННЯ МЕРЕЖІ БАЙЄСА ДЛЯ ОЦІНКИ СТУПЕНЯ ЗНАЧИМОСТІ ВПЛИВАЮЧИХ ФАКТОРІВ НА ЛПР В АВТОМАТИЗОВАНИХ СИСТЕМАХ ПРИ ПРИЙНЯТТІ РЕЛЕВАНТНИХ РІШЕНЬ

© Передерій, В., Бабичев С., Литвиненко В., 2012

Розроблено Байєсівську мережу для оцінки ступеня значимості особистих і зовнішніх факторів, що впливають на прийняття релевантних рішень ЛПР в автоматизованих системах управління.

Ключові слова: особа що приймає рішення, прийняття релевантний рішення, особистісні і зовнішні фактори, Байєсовська мережа.

Developed and investigated Bayesian network designed to assess the significance of personal and external factors affecting the performance of the automated control system

Key words: individual decision-maker, decision-making relevant, personal and external factors, Bayesian network, the coefficients for the change in the probability factor

Вступ

Сучасні автоматизовані системи прийняття рішень характеризуються наявністю складного об'єкта управління з розподіленими параметрами, неоднорідних і інтенсивних інформаційних потоків, що надходять до користувача в режимі реального часу. Робота користувача системи під час управління технологічними об'єктами є напруженою, а наслідки помилок призводять до значного матеріального збитку, людських жертв тощо [1]. Внаслідок цього високу актуальність набуває

завдання створення моделей і алгоритмів, що дають змогу підвищити як безпеку функціонування автоматизованих систем управління, так і прийняття релевантних рішень під час управління складними технологічними об'єктами.

Постановка проблеми

У роботах [1–3] автори проаналізували основні причини порушення працездатності автоматизованих систем управління, до того ж показали, що однією з головних причин є людський фактор. Під час функціонування системи релевантність прийнятого рішення визначається впливом на особу, яка приймає рішення, сукупності зовнішніх і внутрішніх факторів. У літературі [1–4] детально розглянуті питання визначення комфортного робочого середовища користувача системи. У [5, 6] описані розробки з створення математичних моделей і алгоритмів, що дозволяють оцінити релевантність прийнятих рішень з урахуванням впливу зовнішніх і особистих факторів на безпеку роботи автоматизованих систем. У [7] автори розробили алгоритми формалізації взаємозв'язку зовнішніх факторів і психофункціональних характеристик користувачів на основі теорії нечітких множин і алгоритму клонального відбору для оптимізації релевантності прийнятих рішень. Однак водночас не було оцінено ступінь впливу і, як наслідок, ступінь значимості того чи іншого фактора.

Мета роботи

Розробка системи оцінки ступеня значимості особистих і зовнішніх факторів, що впливають на роботу автоматизованої системи управління, на основі мережі Байеса.

Виклад основного матеріалу

Байєсівські мережі являють собою графічні моделі подій і процесів на основі об'єднання деяких висновків теорії ймовірностей і теорії графів [8]. Основу байєсівської мережі становить пара $\langle G, B \rangle$, де перша компонента G – це спрямований ациклічний граф, що відповідає випадковим змінним, а друга B – множина параметрів, що визначають систему.

Повна загальна ймовірність байєсівської мережі обчислюється за формулою

$$P_B(X^{(1)}, \dots, X^{(N)}) = \prod_{i=1}^N P_B(X^{(i)} | P_A(X^{(i)})),$$

де A – множина станів середовища; $N = (x_1, \dots, x_N)$ – вектор параметрів їх розподілів.

Умовна ймовірність події визначається співвідношенням:

$$p(E | H_k) = \frac{p(E \cap H_k)}{p(H_k)}, \quad (1)$$

де E і H_k – пов'язані між собою змінні. Ця залежність дає змогу визначити, якою буде ймовірність події E , якщо має місце конкретна подія H_k .

Взаємовиключні події формують вичерпне, якщо

$$\bigcup_{i=1}^n E_i = \Omega. \quad (2)$$

Дві змінні не перетинаються, якщо вони не мають однакових значень. Теорія побудови мереж Байеса ґрунтується на припущенні, що події є вичерпними і не перетинаються. У цьому випадку ймовірність події можна обчислити за допомогою умовних ймовірностей:

$$p(E) = \sum_{i=1}^n p(E \cap H_i) = \sum_{i=1}^n p(E | H_i) \cdot p(H_i). \quad (3)$$

Використовуючи формулу (1), ймовірність перетину подій E і H можна виразити так:

$$p(E \cap H_k) = p(E | H_k) \cdot p(H_k) = p(H_k | E) \cdot p(E), \quad (4)$$

Звідки одержуємо:

$$p(H_k | E) = \frac{p(E | H_k) \cdot p(H_k)}{p(E)}. \quad (5)$$

Враховуючи (3) формулу (5) можемо записати так:

$$p(H_k | E) = \frac{p(E | H_k) \cdot p(H_k)}{\sum_{i=1}^n p(E | H_i) \cdot p(H_i)}. \quad (6)$$

- Мережа Байєса, формована на підставі формули (6), являє собою спрямований ациклічний граф, де кожен вузол являє собою змінну, а кожна дуга – імовірнісну залежність, зумовлену кількісно використанням умовного розподілу ймовірностей для кожного вузла. До складу мережі Байєса входять такі компоненти:

§ множина вузлів, що визначають компоненти системи;

§ множина спрямованих зв'язків між компонентами системи.

Визначення коефіцієнтів значимості того чи іншого фактора здійснювалося в два етапи:

- розраховувалися ймовірності вхідних факторів при апріорі заданої ймовірності вихідної змінної, до того ж ймовірності значень вихідної змінної змінювалися в максимально можливому діапазоні;
- визначався коефіцієнт максимальної зміни ймовірностей i -го параметра:

$$k_i = \frac{P_{i\max} - P_{i\min}}{\sum P}, \quad (7)$$

де $\sum P$ – сума значень ймовірностей значень i -го фактора за різних апріорі заданих значень вихідної змінної. Очевидно, що чим вищий коефіцієнт максимальної зміни ймовірностей, тим чутливіша система до цього вхідного параметра, отже, значимість цього параметра буде вагомішою.

В [7] представлені дослідження з оцінки релевантності прийнятого рішення з урахуванням особистісних і зовнішніх факторів, що впливають на роботу автоматизованої системи управління. Сукупність факторів, що впливають, була розділена на дві групи:

- 1) фактори, пов'язані з впливом навколишнього середовища користувача;
- 2) фактори, пов'язані з поточним психологічним станом користувача.

До першої групи належать:

- а) інтенсивність шуму I_N ;
- б) інтенсивність вібрацій I_V ;
- в) освітленість робочого місця E ;
- г) температура T ;
- д) вологість f ;
- е) атмосферний тиск P .

До складу групи факторів, пов'язаних з поточним психологічним станом користувача, входять:

- а) рівень інформаційної пропускну здатності I_0 ;
- б) ступінь втоми користувача F ;
- г) обмеженість часу прийняття рішення T_P ;
- д) невідповідність ступеня напруженості TS ;
- е) концентрованість уваги A .

Перша група факторів визначає рівень стану середовища S_c , друга група факторів визначає рівень стану людини S_p . Розв'язання поставленої задачі зводиться до знаходження максимуму цільової функції:

$$R_p = F(S_c^+, S_p^+) \rightarrow \max \quad (8)$$

Рівень стану навколишнього середовища і поточного стану користувачів своєю чергою також виражаються через первинні фактори відповідно до рівнянь:

$$S_c = f_1 \left(I_N^-, I_V^-, E^+, T^{+-}, f^+, \left(\frac{\Delta P}{\Delta t} \right)^- \right) \quad (9)$$

$$S_p = f_2 \left(I_0^+, F^-, T_p^-, TS^-, A^+ \right) \quad (10)$$

Підставивши (9) і (10) в (8), отримуємо цільову функцію автоматизованої системи підтримки прийняття рішень:

$$R_p = F \left[f_1 \left(I_N^-, I_V^-, E^+, T^{+-}, f^+, \left(\frac{\Delta P}{\Delta t} \right)^- \right)^+, f_2 \left(I_0^+, F^-, T_p^-, TS^-, A^+ \right)^+ \right] \rightarrow \max \quad (11)$$

У табл. 1 і 2 наведена формалізація факторів поточного стану навколишнього середовища і стану користувача у вигляді лінгвістичних змінних і термів. У разі виникнення складності у виборі одиниці вимірювання відповідної величини використовувалася рангова шкала, кожен діапазон якої відповідає ступеню прояву того чи іншого фактора.

Таблиця 1

Формалізація факторів стану навколишнього середовища в лінгвістичні змінні

Первинний фактор	Границі зміни	Терми лінгвістичної оцінки
I_N (дБА)	0–75	0-20 – «Н»(низький), 21-45 – «С»(середній), 46-75 – «В»(високий)
I_V (мм/с)	0–15	0-4 – «Н»(низький), 5-10 – «С»(середній), 11- 15 «В»(високий)
E (Лк)	0–600	0–200 – «Н»(низький), 201-400 – «С»(середній), 401–600 – «В»(високий)
T (°C)	10–40	10–17 – «Н»(низька), 18–25 – «С»(середня), 26–40 – «В»(висока)
f (%)	0–90	0–30 – «Н»(низька), 31–60 – «С»(середня), 61-90 – «В»(висока)
$\frac{\Delta P}{\Delta t}$ (мм рт. ст/доб)	0–15	0-3 – «Н»(низька), 4-7 – «С»(середня), 8-15 – «В»(висока)

Таблиця 2

Формалізація факторів поточного стану користувача в лінгвістичні змінні

Первинний фактор	Границі зміни	Терми лінгвістичної оцінки
I_0 (ранг)	0–15	0-5 – «Н»(низька), 6-10 – «С»(середня), 11–15 – «В»(висока)
F (ранг)	0–15	0-5 – «Н»(низька), 6-10 – «С»(середня), 11–15 – «В»(висока)
T_p (хвил)	0–60	0-5 – «В»(висока), 5-30 – «С»(середня), 30–60 – «Н»(низька)
TS (ранг)	0–15	0-5 – «Н»(низька), 6-10 – «С»(середня), 11–15 – «В»(висока)
A (ранг)	0–15	0-5 – «Н»(низька), 6-10 – «С»(середня), 11–15 – «В»(висока)

Для оцінки стану користувача системи була введена рангова шкала: «ДН» (дуже низька, $0 < R_p \leq 3$), «Н» (низька, $3 < R_p \leq 6$), «С» (середня, $6 < R_p \leq 9$), «В» (висока, $9 < R_p \leq 12$), «ДВ» (дуже висока, $12 < R_p \leq 15$).

Розв'язували задачу побудови мережі Байєса з використанням програмного середовища GeNIe 2.0 [9]. Під час побудови мережі Байєса в якості вхідних змінних використовувалися характеристики стану зовнішнього середовища і користувача, а в якості вихідної змінної релевантність прийнятих рішень користувачем системи. Для побудови структури зв'язків мережі використовувалися знання експерта в певній предметній області. Спочатку був побудований граф взаємного впливу факторів стану навколишнього середовища, поточного стану користувача та релевантності прийняття ним рішень. Потім цей граф був розширений за допомогою введення експертом візуальних зв'язків між елементами системи. Отримана структура мережі Байєса наведена на рис. 1. На рис. 2 показано Байєсову модель співвідношення вузлів мережі при високій релевантності.

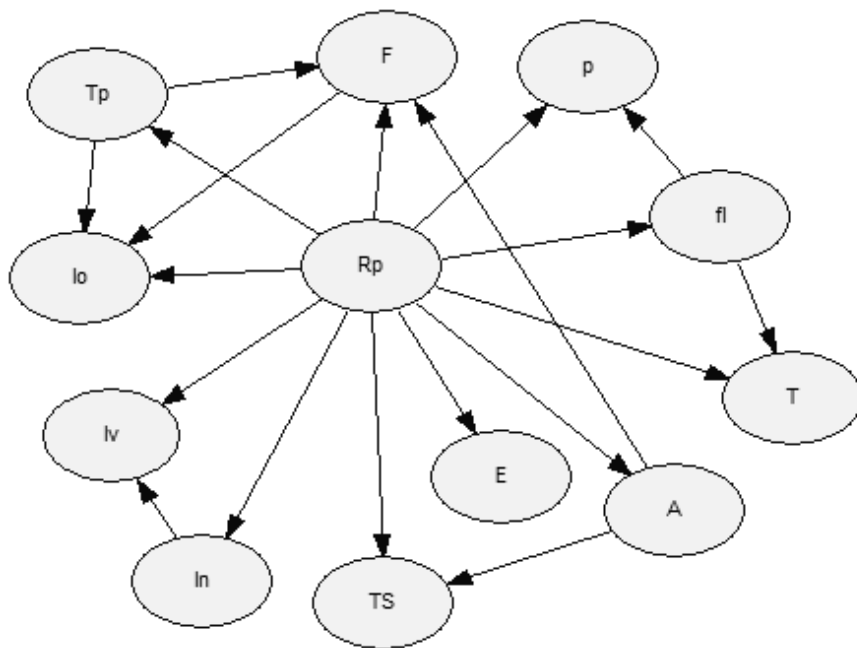


Рис. 1. Структура зв'язків мережі Байєса

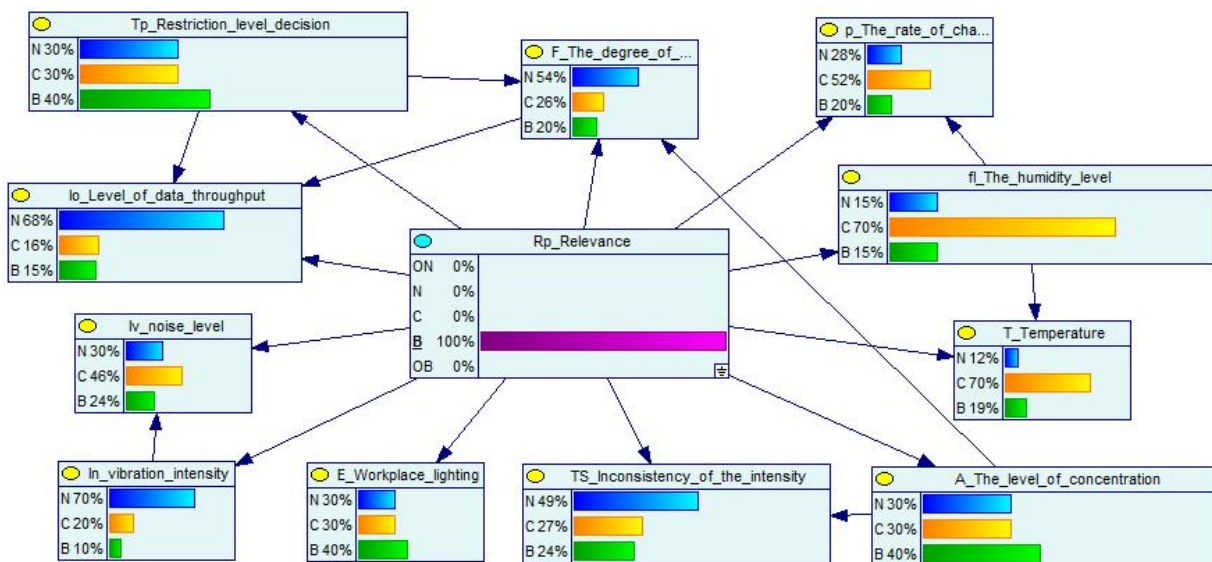


Рис. 2. Байєсова модель ймовірностей вузлів мережі при високій релевантності прийняття рішень ЛПП

Розподіл ймовірностей факторів стану середовища та стану користувача при крайніх діапазонах наведені в табл. 3. Значення коефіцієнтів максимальної зміни ймовірностей для відповідних факторів, розраховані за формулою (7), наведені в табл. 4. На рис. 3 показана гістограма розподілу коефіцієнтів.

Аналіз даних табл. 4 і гістограми дає змогу зробити висновок, що максимально значущі фактори, що впливають на релевантність прийняття рішення, є: інтенсивність шуму навколишнього середовища, інтенсивність вібрацій, рівень освітленості, ступінь втоми користувача, обмеженість часу прийняття рішень і рівень концентрації уваги користувача системи. Облік коефіцієнтів значущості факторів, що впливають, дозволяє підвищити точність оцінки релевантності прийнятих рішень користувачем у системі нечіткого логічного висновку шляхом урахування коефіцієнта значимості того чи іншого фактора. До того ж система буде чутливіша до більш значущих чинників і менш чутлива до факторів, коефіцієнт значимості яких мінімальний.

Таблиця 3

Розподіл ймовірностей факторів стану середовища та стану користувача при крайніх значеннях релевантності прийняття рішення користувачем системи

0 < R _p ≤ 3											
	I _N	I _V	E	T	f	$\Delta P / \Delta t$	I _o	F	T _p	TS	A
H	0	10	80	38	40	58	36	10	0	89	90
C	10	11	15	42	20	30	32	12	10	10	10
B	90	79	5	20	40	12	32	78	90	1	0
12 < R _p ≤ 15											
	I _N	I _V	E	T	f	$\Delta P / \Delta t$	I _o	F	T _p	TS	A
H	90	79	5	11	10	17	51	77	50	29	10
C	10	11	15	80	80	66	25	13	40	30	30
B	0	10	80	9	10	16	24	11	10	41	60

Таблиця 4

Значення коефіцієнтів максимального зміни ймовірностей для факторів, що впливають на релевантність прийняття рішень

Фактори стану середовища і стану користувача											
	I _N	I _V	E	T	f	$\Delta P / \Delta t$	I _o	F	T _p	TS	A
k	0,9	0,69	0,75	0,38	0,6	0,41	0,15	0,67	0,8	0,6	0,8

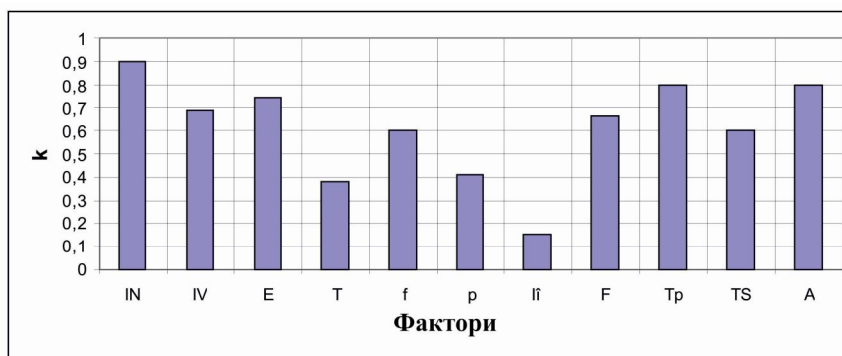


Рис. 3. Гістограма розподілу коефіцієнтів максимальної зміни ймовірностей впливаючих факторів

Висновки

У роботі розроблено модель урахування коефіцієнтів значущості факторів стану середовища та стану користувача при визначенні релевантності прийнятих рішень користувачем автоматизованої системи, заснована на мережі Байєса. Побудована Байєсова мережа, враховує характер взаємозв'язку між впливаючими факторами і релевантністю прийняття рішень користувачем. Дано рекомендації з обліку коефіцієнтів значущості факторів, що впливають на нечіткій системі визначення, що сприяє підвищенню об'єктивності при оцінці релевантності прийняття рішень користувачем системи. Надалі автори планують розробити комплексну систему визначення релевантності прийнятих рішень користувачем, засновану на мережі Байєса і системі нечіткого логічного висновку.

1. Человеческий фактор: В 6 т. Т. 2. Эргономические основы проектирования производственной среды: Пер. с англ. / Д. Джоунз, Д. Бродбент, Д.Е. Вассерман и др. – М.: Мир, 1991. – 500 с.

2. Справочник по инженерной психологии / Под ред. Б.Ф. Ломова. – М.: Машиностроение, 1982. – 368 с. 3. Шибанов Г.П. Количественная оценка деятельности человека в системах человек-техника. – М.: Машиностроение, 1983. – 263 с. 4. Шеридан Т.Б., Феррел У.Р. Системы человек-машина: Модели обработки информации, управления и принятия решений человеком-оператором: Пер. с англ. / Под ред. К.В. Фролова. – М.: Машиностроение, 1980. – 400 с. 5. Передерий В.И., Еременко А.П. Математические модели и алгоритмы определения релевантности принимаемых решений с учетом психофункциональных характеристик пользователей при управлении автоматизированными динамическими системами // Автоматика, автоматизация, электротехнические комплексы и системы. – 2007. – № 2. – С. 34–40. 6. Передерий В.И., Еременко А.П. Математические модели и алгоритмы принятия релевантных решений в автоматизированных системах с учетом личностных и внешних факторов на базе генетических алгоритмов // Автоматика, автоматизация, электротехнические комплексы и системы. – 2008. – № 2. – С. 28–37. 7. Передерий В.И., Бабичев С.А., Литвиненко В.И. Применение алгоритма клонального отбора для принятия релевантных решений пользователями с учетом личностных и внешних факторов // Системні технології. Регіональний міжвідомстний збірник наукових праць. – Вип. 6(77). Дніпропетровськ, 2011. – С. 117–131. 8. Murphy K. A brief introduction to graphical models and Bayesian networks / Technical report 2001-5-10, department of computer science, University of British Columbia, Canada, May 2001. – 19 p. 9. <http://genie.sis.pitt.edu/>

УДК 620.3, 621.3

А. Пукач, В. Теслюк, Р.-А. Іванців, М. Лобур
Національний університет “Львівська політехніка”,
кафедра систем автоматизованого проектування

РОЗРОБЛЕННЯ МЕТОДУ ВИЗНАЧЕННЯ ДІАПАЗОНУ ЗНАЧЕНЬ ВЕЛИЧИНИ ВИМІРЮВАНОВОГО ЕЛЕКТРИЧНОГО ОПОРУ

© Пукач А., Теслюк В., Іванців Р.-А., Лобур М., 2012

Розроблено метод визначення діапазону значень величини вимірюваного електричного опору, алгоритм і модель функціонування на основі кольорових мереж Петрі, та схемну модель підсистеми визначення діапазону значень величини вимірюваного електричного опору, які виступають ефективним інструментом під час розв’язання задачі вимірювання малих величин електричного опору в МЕМС.

Ключові слова: електричний опір, МЕМС, модель, кольорові мережі Петрі.

In this paper method for measured electric resistance value range determining, algorithm and functioning model based on colored Petri nets, and schematic model of measured electric resistance value range determining subsystem, that appears as effective tool for solving the problem of measuring small quantities of electric resistance in MEMS.

Keywords: electric resistance, MEMS, model, colored Petri nets.

Вступ

Широке впровадження у всі сфери людської діяльності технологій виготовлення мікроелектромеханічних систем (МЕМС) передбачає створення нових, або удосконалення вже існуючих [6] методів вимірювання малих величин електричного опору, зумовлених мікронними розмірами активних компонентів МЕМС [1–5]. Одним із способів покращання точності

вимірювання малих величин електричного опору є визначення діапазону значень величини вимірюваного електричного опору, що дає змогу здійснити попередню оцінку вимірюваної величини з метою подальшого встановлення її точного значення. Отже, визначення діапазону значень величини вимірюваного електричного опору є актуальною задачею, розв'язання якої дасть змогу покращити точність вимірювання малих величин електричного опору в МЕМС.

З метою підвищення точності вимірювання малих величин електричного опору в статті розроблений метод визначення діапазону значень величини вимірюваного електричного опору. Використання розробленого методу передбачає вимірювання величини невідомого електричного опору в два етапи: на першому етапі відбувається попередня оцінка вимірюваної величини за допомогою, власне, розробленого методу; на другому етапі відбувається безпосереднє вимірювання величини невідомого електричного опору з врахуванням результатів, отриманих на попередньому етапі.

Розроблення схемної моделі підсистеми визначення діапазону значень величини вимірюваного електричного опору

Нижче на рис. 1 зображена розроблена схемна модель підсистеми визначення діапазону значень величини вимірюваного електричного опору. Модель містить такі основні компоненти (рис. 1): блок зразкових резисторів R_1 – R_n , блок комутації на основі ключів k_1 – k_n , вимірюваний електричний опір R_x , два еталонні електричні опори $R_{ет1}$ та $R_{ет2}$ (причому $R_{ет1}=R_{ет2}$), блок виділення модуля напруги, блок розподілу залишкових напруг по каналах, схему визначення мінімальної напруги та схему управління.

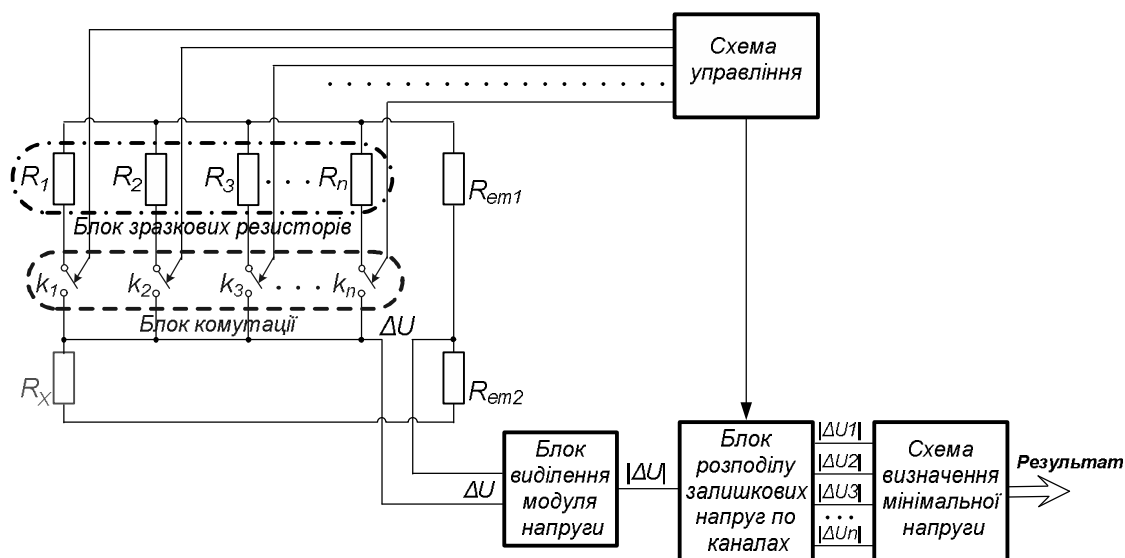


Рис. 1. Схемна модель підсистеми

Блок зразкових резисторів являє собою набір електричних опорів R_1 – R_n різних номіналів, які, власне, представляють можливі діапазони значень величини електричного опору. Блок комутації складається із ключів k_1 – k_n , котрі безпосередньо пов'язані із схемою управління і призначені для замикання потрібного зразкового резистора в електричне коло із вимірюваним опором R_x . Еталонні електричні опори $R_{ет1}$ та $R_{ет2}$ виступають сталою складовою подільника напруги. Блок виділення модуля напруги призначений для виділення абсолютної величини залишкової напруги ΔU . Блок розподілу залишкових напруг по каналах призначений для формування сигналів з амплітудами, що дорівнюють залишковим напругам ΔU_1 – ΔU_n для їх подальшого порівняння. Схема визначення мінімальної напруги призначена, відповідно, для визначення мінімальної із усіх вхідних залишкових напруг ΔU_1 – ΔU_n з метою визначення зразкового опору R_j , котрому відповідає виявлена мінімальна залишкова напруга U_j . Схема управління призначена для синхронізації роботи усієї підсистеми, з її допомогою відбувається включення усіх можливих діапазонів величини електричного опору шляхом почергового замикання усіх ключів блока комутації.

Нижче на рис. 2 зображені діапазони величини електричного опору розробленої підсистеми.

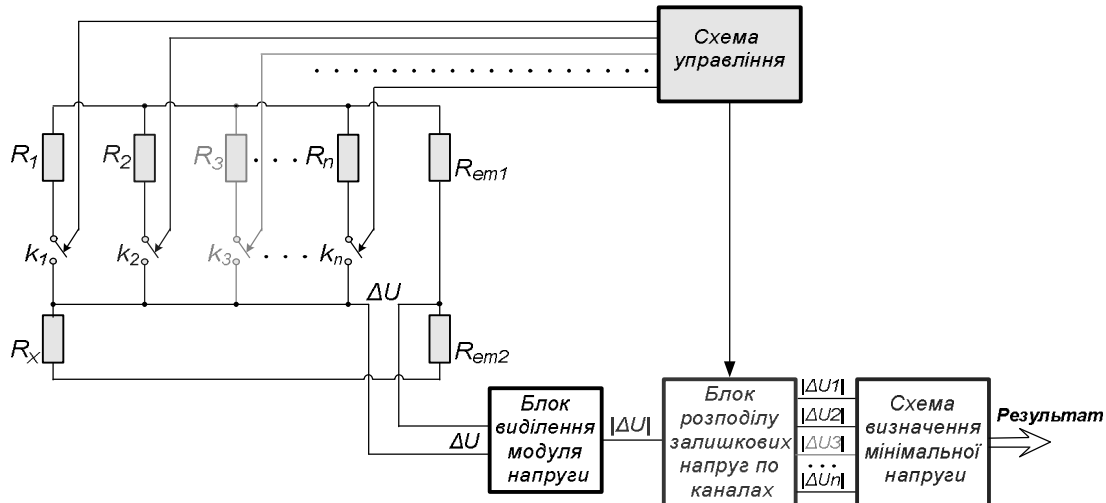


Рис. 2. Діапазони величини електричного опору

У результаті почергового замикання ключів k_1-k_n підсистеми створюється залишкова напруга ΔU дизбалансу електричного моста на одного із зразкових резисторів R_i , вимірюваного опору R_x та еталонних резисторів R_{em1} і R_{em2} . Отже, серед усіх зразкових опорів R_1-R_n найближчим за величиною до вимірюваного опору R_x буде той зразковий електричний опір R_j , у разі ввімкнення якого абсолютне значення залишкової напруги $|\Delta U_j|$ дизбалансу електричного моста буде мінімальним. Звідси випливає, що значення величини вимірюваного електричного опору R_x знаходиться в околі значення величини зразкового електричного опору R_j .

Розроблення алгоритму функціонування методу визначення діапазону значень величини вимірюваного електричного опору

Нижче на рис. 3 зображена блок-схема алгоритму функціонування методу.

Алгоритм працює так. Відбувається встановлення індексу діапазону величини електричного опору таким, що дорівнює одиниці ($i:=1$), після чого виявляється залишкова напруга ΔU дизбалансу електричного моста, обчислюється її абсолютне значення ($\Delta U := |\Delta U|$), яке зберігається у відповідній комірці ΔU_i . Після цього відбувається збільшення індексу діапазону величини електричного опору ($i := i + 1$), виявлення, обробка та збереження залишкової напруги поточного діапазону величини електричного опору аж до тих пір, поки не будуть проаналізовані усі діапазони ($i > n$). Наступним етапом роботи алгоритму є процес розподілу залишкових напруг $\Delta U_1-\Delta U_n$ по каналах, після чого відбувається визначення мінімальної залишкової напруги ΔU_j та вивід результатів роботи підсистеми.

Розроблення моделі функціонування методу визначення діапазону значень величини вимірюваного електричного опору на основі кольорових мереж Петрі

На рис. 4 зображена розроблена модель функціонування методу визначення діапазону значень величини вимірюваного електричного опору на основі кольорових мереж Петрі [7–11].

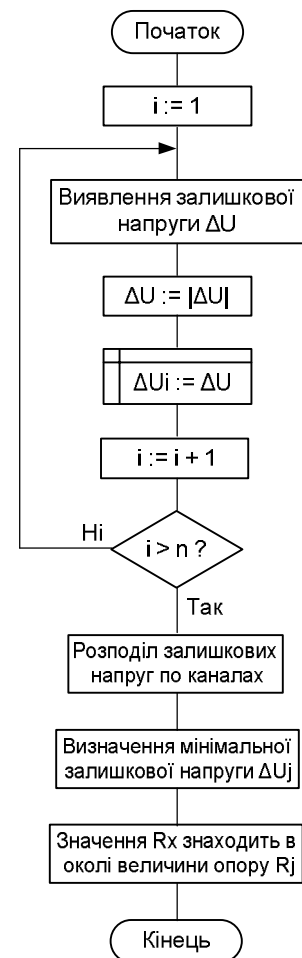


Рис. 3. Блок-схема алгоритму

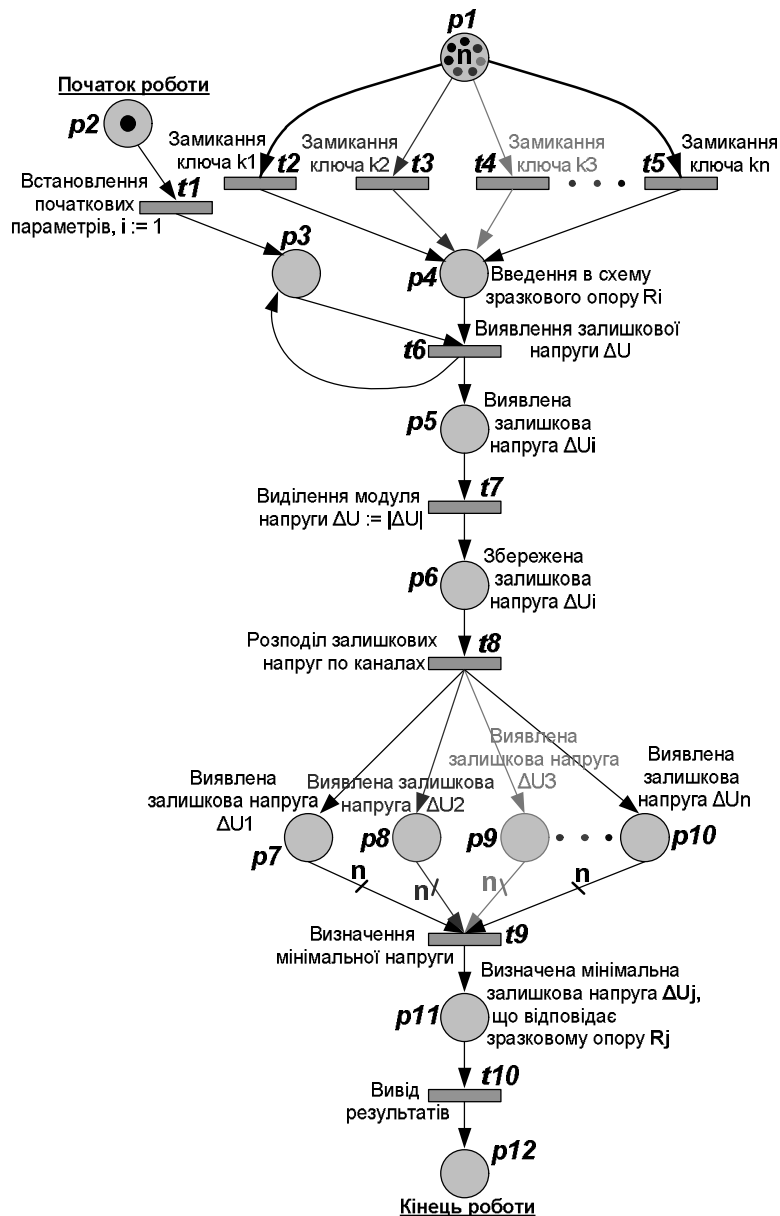


Рис. 4. Модель функціонування на основі кольорових мереж Петрі

Побудована модель працює так. Перед безпосереднім початком роботи відбувається встановлення початкових параметрів, тобто задання кількості робочих діапазонів величини електричного опору, в позиції **p1**. У разі зміни величини діапазонів (n) потрібно змінити також кількість відповідних переходів замикання ключів та індивідуальних позицій виявлених залишкових напруг (див. рис. 4). Після здійснення усіх необхідних модифікацій моделі потрібно помістити маркер в позицію початку роботи **p2**. У результаті спрацювання переходу **t1** відбувається встановлення початкових параметрів підсистеми, зокрема індексів діапазонів величини електричного опору присвоюється значення одиниці ($i := 1$), і маркер переходить в позицію **p3**. Після цього спрацьовує перехід **t2**, тобто, відбувається замикання ключа k_1 підсистеми, маркер першого діапазону величини електричного опору з позиції **p1** переходить в позицію **p4**, що свідчить про введення в схему зразкового опору R_i активного діапазону величини електричного опору (у цьому випадку R_1). Спрацьовує перехід **t6**, тобто відбувається виявлення залишкової напруги ΔU , що відповідає поточному зразковому резистору R_i , маркер переходить в позиції **p3** та **p5**. Маркер в позиції **p5** активізує перехід **t7**, котрий запускає процес виділення модуля залишкової напруги ($\Delta U = |\Delta U|$) і маркер переходить в позицію **p6**. Активізацією переходу **t8** маркером з позиції **p6** запускається механізм розподілу залишкових напруг по каналах і збереження поточної залишкової напруги ΔU у

відповідній комірці ΔU_i , Маркер переходить в позиції виявлених залишкових напруг **p7–p10**, збільшуючи кількість маркерів у відповідних позиціях на 1 (кількість маркерів в цих позиціях відповідає поточному діапазону величини електричного опору), що свідчить про завершення аналізу поточного робочого діапазону величини електричного опору.

Після цього відбувається перехід до аналізу наступного робочого діапазону величини електричного опору активізацією переходу **t3** маркером другого діапазону величини електричного опору з позиції **p1**. Відбувається замикання другого ключа k_2 підсистеми, введення в схему наступного зразкового резистора (у цьому випадку R_2), а також аналіз, обробка та збереження відповідної залишкової напруги ΔU_i (у цьому випадку ΔU_2) тощо, аж доти поки не буде проаналізований останній n -й робочий діапазон величини електричного опору. В цьому випадку в позиціях **p7–p10** буде по n маркерів, тобто їх кількість буде достатньою для спрацювання переходу **t9**, що запускає механізм визначення мінімальної напруги. Маркер переходить в позицію **p11**, що свідчить про визначення діапазону значень величини вимірюваного електричного опору, активізується перехід **t10**, відбувається вивід результатів роботи підсистеми і маркер потрапляє в позицію кінця роботи **p12**.

Нижче на рис. 5 зображений граф досяжності станів розробленої кольорової мережі Петрі.

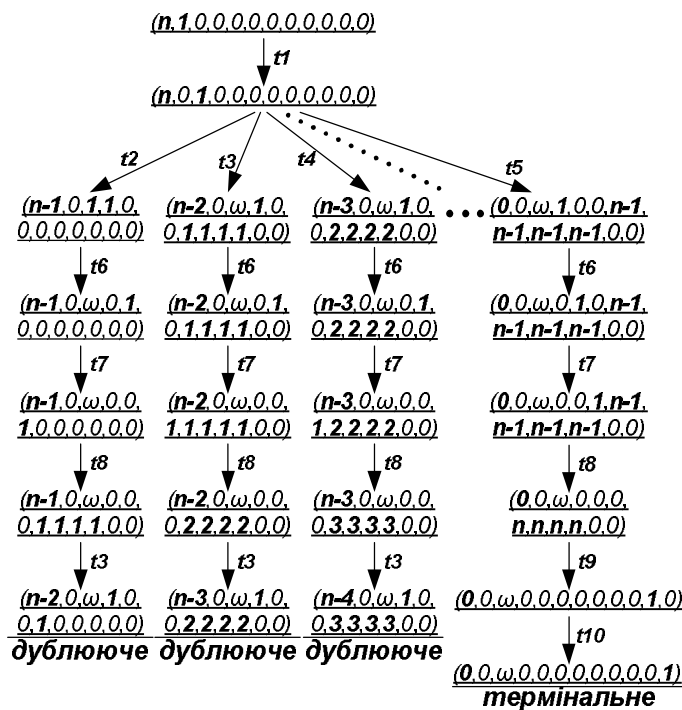


Рис. 5. Граф досяжності розробленої кольорової мережі Петрі

Побудований граф досяжності демонструє послідовно-паралельний ітераційний принцип роботи розробленої моделі, де для маркерів різного типу здійснюються одні і ті самі операції, і лише після опрацювання усіх маркерів здійснюється перехід до заключної ланки дерева. Така структура графа повністю відповідає розробленій моделі та методу, загалом.

Висновки

Розроблено метод визначення діапазону значень величини вимірюваного електричного опору та схемну модель підсистеми визначення діапазону значень величини вимірюваного електричного опору, яка виступає ефективним інструментом під час попередньої оцінки величини вимірюваного електричного опору з метою покращання точності подальшого вимірювання його точного значення. Крім того, в роботі запропоновано алгоритм і модель функціонування на основі кольорових мереж Петрі розробленого методу визначення діапазону

значень величини вимірюваного електричного опору. Розроблений метод дає змогу роз'язати задачу вимірювання малих величин електричного опору в MEMS.

1. Kruglick J. J. *EFAB Technology and Applications* / J. J. Kruglick, A. Cohen, C. Bang // *MEMS: Design and Fabrication* / [Mohamed Gad—el—Hak, ed.]. – 2nd ed. – Boca Raton: CRC Press, 2006. – 664 p.
2. Колпаков Ф. Ф. *Микроэлектромеханические устройства в радиотехнике и системах телекоммуникаций* : [учеб. пособие] / Ф. Ф. Колпаков, Н. Г. Борзяк, В. И. Кортунов. – Харьков : НАУ ХАИ, 2006. – 82 с.
3. *Minhang Bao Analysis and Design Principles of MEMS Devices*, – 1st edition: Elsevier Science, 2005 – 328 p.
4. James J. Allen *Micro Electro Mechanical System Design*, – 1st edition: CRC Press, 2005 – 496 p.
5. Marc J. Madou *Fundamentals of Microfabrication: The Science of Miniaturization*, – 2nd edition: CRC Press, 2002 – 752 p.
6. Пукач А., Іванців Р., Теслюк В. *Методи та схеми вимірювання значення невідомого опору електричного кола*. *Технічні вісті: Науково-публіцистичний часопис "Львівська політехніка"*. Львів: Видавництво НУ"Львівська політехніка". – 2010. – С. 160–163.
7. Джеймс Питерсон *Теория сетей Петри и моделирование систем: Пер. с англ.* – М.: Мир, 1984. – 264 с.
8. Котов В.Е. *Сети Петри*. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.
9. Васильев В.В., Кузьмук В.В. *Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем*. К.: Наук. думка, 1990. – 216 с.
10. James L. Peterson *A Note on Colored Petri Nets*, *Information Processing Letters*, Volume 11, Number 1, (August 1980), pages 40–43.
11. Kurt Jensen, Lars M. Kristensen. *Coloured Petri Nets: modelling and validation of concurrent systems: 1st edition* – Springer, 2009. – 395 p.

УДК 681.5 , 62-529, 62-58

А. Павельчак¹, В. Самотий^{1,2}, Р. Василів

¹ Національний університет "Львівська політехніка",
кафедра комп'ютеризованих систем автоматики

² Politechnika Krakowska, Katedra Automatyki
i Technik Informatycznych, Polska

АЛГОРИТМ КЕРУВАННЯ ДВОХОСЬОВОЮ ПЛАТФОРМОЮ

© Павельчак А., Самотий В., Василів Р., 2012

Запропоновано алгоритм для керування стабілізацією положення двохосевої платформи з використанням біполярних крокових моторів, акселерометра та керуючого мікроконтролера.

Ключові слова: кроковий двигун, акселерометр, мікроконтролер.

In this paper, the authors have proposed the algorithm to control the stabilization position 2-axial platform using bipolar stepper motors, accelerometer and microcontroller.

Key words: stepper motor, accelerometer, microcontroller.

Вступ

Завдання автоматичної стабілізації положення рухомої платформи у наперед заданій площині є давно відомим і в різні способи реалізованим. Найпоширеніший варіант – це системи з використанням гіроскопів. Потреба у стабілізації об'єктів та окремих їхніх вузлів виникає як в інерційних навігаційних системах, що встановлюються на кораблях та літальних апаратах, габаритному автомобільному транспорті, бурових платформах, так і в компактній побутовій техніці. Залежно від сфери застосування, використовуються, відповідно, різні пристрої, що виконують функції гіроскопа. Тому кожне завдання стабілізації потребує врахування своїх особливостей.

Сьогодні великої популярності набувають мікромеханічні пристрої, побудовані з використанням мікроелектромеханічної системи (MEMS – Micro Electromechanical Systems), серед них і мікромеханічні акселерометри. З швидким розвитком технологій виготовлення MEMS виникла і потреба в розробленні нових програмних інструментів для проектування та моделювання систем на основі таких пристроїв. У цій статті пропонуємо програмний алгоритм стабілізації положення двохосової платформи на основі даних, що отримуються з мікромеханічного акселерометра.

Аналіз публікацій

Акселерометри [1, 3, 5, 6] можна використовувати у будь-якому пристрої, робота якого пов'язана з переміщенням, нахилом, вібрацією. Наведемо деякі типові застосування акселерометрів. Автомобілебудування. Давачі використовуються для систем безпеки (подушки безпеки, протиугінні системи), короточасної навігації, діагностики автомобіля [2]. Авіоніка. Давачі прискорення широко використовуються у системах керування літальними апаратами, а також в космічних системах. Вимірюються вібрації і квазістатичні прискорення [4]. Побутова техніка. Використовуються для зменшення вібрацій (пральної машини тощо). Системи захисту від падіння. Типовий приклад – захист жорсткого диска в ноутбуках або інших мобільних пристроях. Під час падіння відбувається парковка магнітної головки жорсткого диска. Медицина і спорт. Акселерометри застосовуються в системах діагностики, тренажерах, крокомірах тощо. Пристрої введення інформації. Використовуються для комп'ютерних ігор, мобільних телефонів тощо.

Постановка задачі

На основі запропонованої системи (рис. 1) необхідно розробити програмний алгоритм, який підтримуватиме стабільне положення двохосової платформи для заданих кутів нахилу, незалежно від зміни нахилу основи установки. Кути нахилу при цьому можуть змінюватися програмно. Також передбачається, що осі платформи відхилятимуться лише в межах $\pm 90^\circ$.

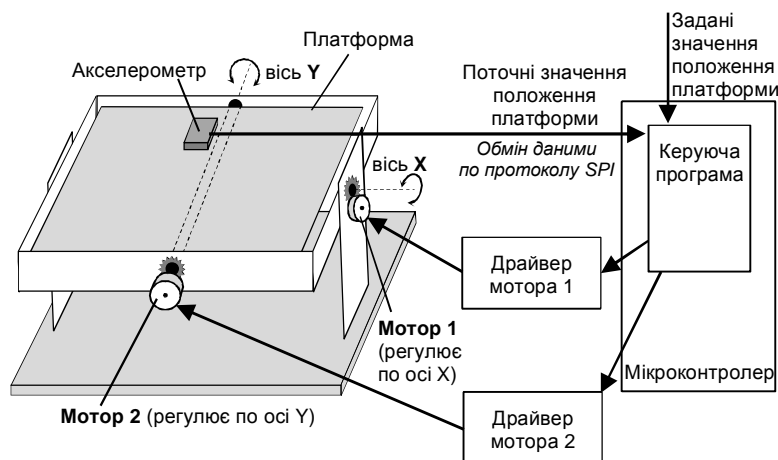


Рис. 1. Структурна схема системи керування для стабілізації положення платформи

Акселерометр генерує кути відхилення відносно горизонту по двохосях X та Y як додатні та від'ємні значення, залежно від вибраної чутливості давача. За наявності заводської похибки калібрування давача відносно горизонтального положення алгоритм повинен передбачати можливість корекції.

Драйвер біполярного крокового мотора повинен підтримувати базові режими роботи моторів: однофазний, двофазний, півкроковий, розгону, гальмування. У режимі гальмування необхідно зменшувати робочий струм обмоток за допомогою ШІМ-регулювання.

На рис. 2 показано реалізацію драйвера. Н-мости виконані на польових транзисторах IRF7105 з N- та P-каналами в одному корпусі та з керуванням логічним рівнем. Для керування

польовими транзисторами як їхні драйвери використані біполярні транзисторні ключі BC337. Ці ключі заодно виконують і функцію захисту від збоїв у програмі мікроконтролера, оскільки забезпечують однакову логіку для N- та P-каналу польової транзисторної зборки. Керування на драйвер крокового мотора подається з мікроконтролера на входи Ain, Bin, Cin та Din. Польові зборки захищені від викидів напруги при відімкненні обмоток двигуна діодами Шоткі. Апаратне ШІМ-регулювання струму реалізоване на логіці І та АБО (мікросхеми 4081, 4071). На вивід PWM драйвера подається ШІМ-сигнал з виводу таймера мікроконтролера, а на вивід Men – дозвіл на активацію ШІМ-регулювання струму обмоток мотора. Для дроблення кута нахилу платформи мотори працюють через редуктори.

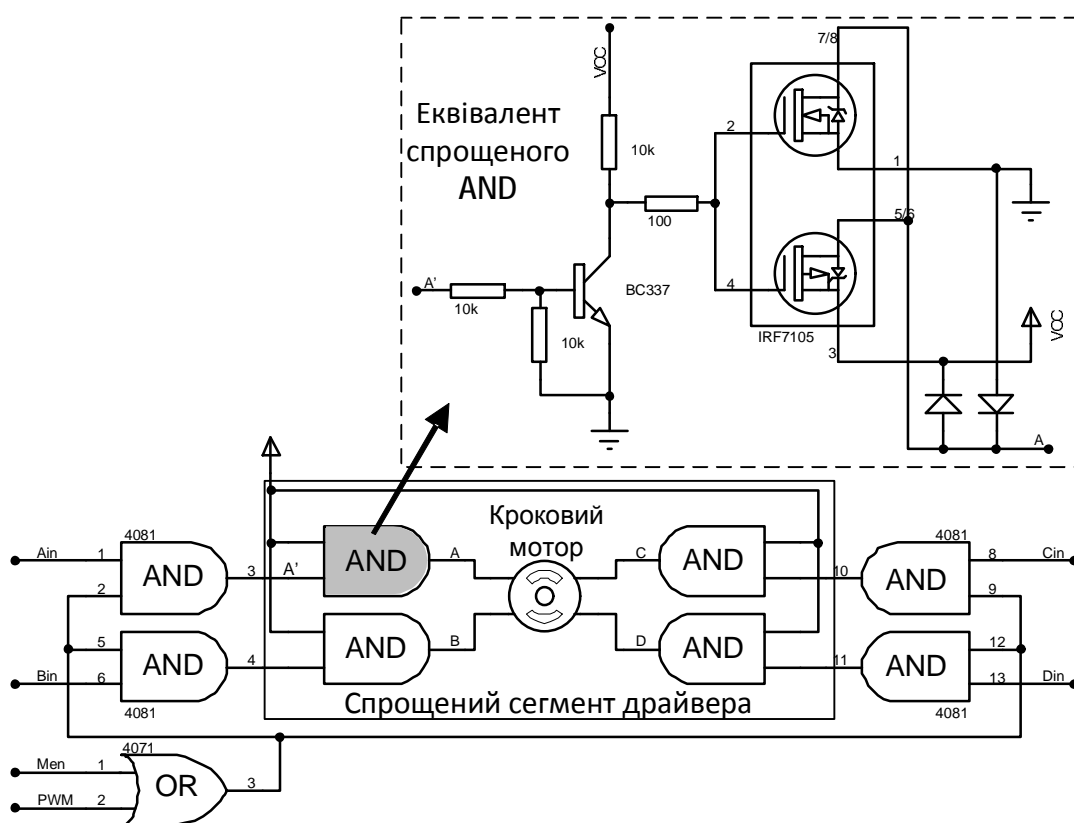


Рис. 2. Реалізація драйвера біполярного крокового мотора

На рис. 3 показані режими, які повинен відпрацьовувати розроблений алгоритм. Базовий режим (А) керування роботою мотора передбачає, що ділянки розгону та гальмування не перетинаються, тобто мотор спершу розганяється до встановленої швидкості, обертається деякий час з номінальною швидкістю, а потім за досягнення кінцевого маршруту сповільнюється. Початок ділянки гальмування визначається згідно з наперед вибраним значенням акселерометра.

Однак базовий режим не враховує деякі нюанси. У разі старту руху поблизу InhibitionZone мотор ще не встиг розігнатися, а вже потрібно гальмувати (рис. 3, Б). Може трапитися, що мотор більшу частину ділянки InhibitionZone пройде на мінімальній швидкості. Або, якщо мотор стартуватиме у самій ділянці, то, відповідно, він буде обертатися до кінцевого пункту на мінімальній швидкості. А це знижує швидкісні характеристики та динаміку цілої системи.

Інший варіант поведінки платформи – це різка зміна координат з + на -, і навпаки (рис. 3, В). Наприклад, платформа перебувала у стані спокою і отримала показ +20 з акселерометра. Згідно з алгоритмом мотор починає поступово прискорюватися до номінальної швидкості. І тут стрибком показ акселерометра змінюється на від'ємне значення, наприклад -23. Наш алгоритм попросту змінить напрям руху, хоча, логічно, він мав би поступово пригальмувати, а потім почати прискорюватися у протилежний бік.

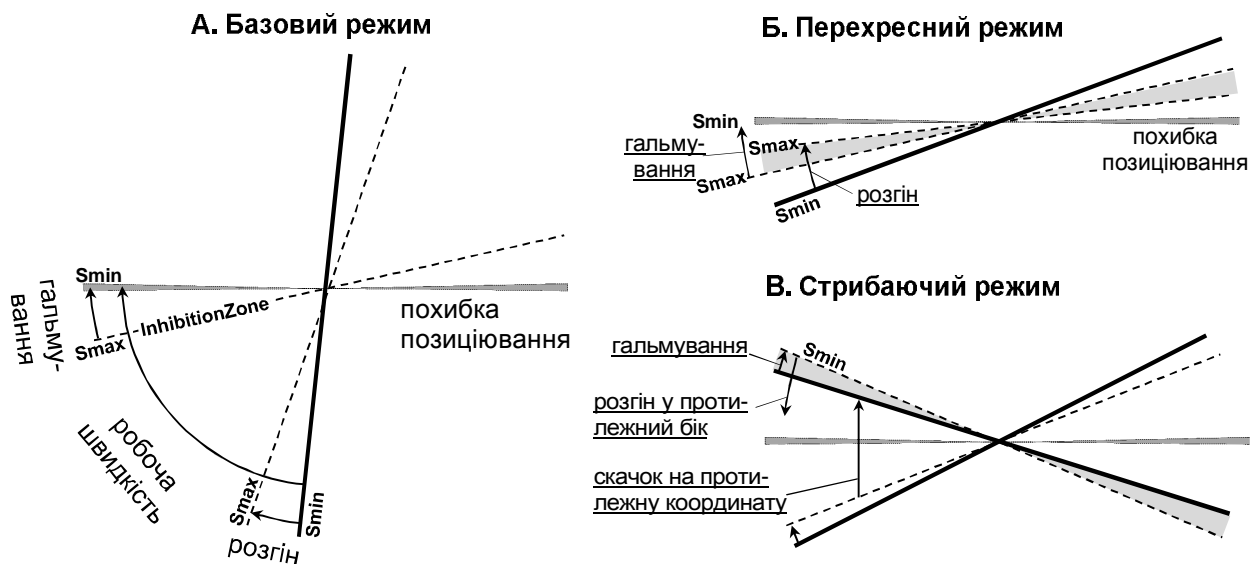


Рис. 3. Режими роботи крокового мотора

Реалізація алгоритму

Режими роботи драйвера крокового мотора (однофазний, двофазний, півкроковий) реалізуємо за допомогою масиву даних, представлених у двійковому вигляді

SM[3][8] = { {0001₂, 0100₂, 0010₂, 1000₂, 0001₂, 0100₂, 0010₂, 1000₂}, /1-фазний/
 {0101₂, 0110₂, 1010₂, 1001₂, 0101₂, 0110₂, 1010₂, 1001₂}, /2-фазний/
 {0001₂, 0101₂, 0100₂, 0110₂, 0010₂, 1010₂, 1000₂, 1001₂}}; /півкроковий/

Логіку розгону/гальмування реалізуємо на затримці за рахунок пропусків тактів таймера (сам алгоритм керування має бути реалізований у підпрограмі переривання таймера мікроконтролера). Тобто, спершу пропускаємо n кроків, потім n-1, n-2 і до 0. У алгоритмі ця логіка буде реалізована на змінних Speed та N. У разі гальмування Speed щокроку зменшуватиметься, а у разі розгону – збільшуватиметься до значення Smax. Для режиму гальмування необхідно також вказувати межі його зони (InhibitionZone), що визначатиметься з показів акселерометра. Обчислюють значення InhibitionZone за такою формулою:

$$\text{InhibitionZone} = \frac{\text{Accel}_{90^\circ} \times \text{Nstep}_X}{\text{Nstep}_{90^\circ}}, \quad (1)$$

де Accel_{90°} – статичні покази акселерометра за відхилення на 90°, Nstep_{90°} – кількість кроків, які виконує двигун у разі повороту осі платформи на 90°, Nstep_X – кількість кроків двигуна, які необхідні для зупинки у зоні стабілізації.

Напрямки руху мотором задаються за допомогою елементів переліку
enum {Stop, Forward, Back} DirM, DirMnew;

Алгоритм керування

1. Початкова ініціалізація: Xcor, Ycor, Epos, InhibitionZone, Smax, Smin=1, Speed=Smin, N=Smax, mode, sw=0, PORTC=SM[mode][7], ШІМ включений, K=0, logic=1, DirM=Stop.
2. Отримання значень кутів відхилень X та Y.
3. Xpos = X – Xcor. /корекція положення/
4. **якщо** |Xpos| > Epos, **тоді**: /мотор обертається/
 - 4.1. виключення ШІМ-регулювання струму.
 - 4.2. **якщо** N=Speed, **тоді**: /виконуємо крок двигуном/
 - 4.2.1. N=Smax. /поч. знач. для лічильн. пропуску тактів таймера/
 - 4.2.2. **якщо** DirM=Stop, **тоді**:

- 4.2.2.1. **якщо** $X_{pos} > 0$, **тоді:** DirM=Forward; DirMnew=Forward
інакше: DirM=Back; DirMnew= Back
- інакше:**
- 4.2.2.2. **якщо** $X_{pos} > 0$, **тоді:** DirMnew=Forward
інакше: DirMnew= Back
- 4.2.3. **якщо** DirM \neq DirMnew, **тоді:** /зупиняємо мотор гальмуючи/
4.2.3.1. **якщо** Speed $>$ Smin, **тоді:** Speed-- /гальмуємо/
інакше: DirM=Stop
- інакше:**
- 4.2.3.2. **якщо** $|X_{pos}| \leq \text{Inhibition Zone}$, **тоді:** /зона гальмування/
4.2.3.2.1. **якщо** logic=1, **тоді:** /1-й раз у зоні гальмування/
4.2.3.2.1.1. $K = (S_{max} - \text{Speed}) / 2$;
4.2.3.2.1.2. logic=0;
4.2.3.2.2. **якщо** $K < 0$, **тоді:**
4.2.3.2.2.1. K--;
4.2.3.2.2.2. Speed++; /прискорюємо/
інакше:
4.2.3.2.2.3. **якщо** Speed $>$ Smin, **тоді** Speed--
/гальмуємо/
інакше:
4.2.3.2.3. **якщо** Speed $<$ Smax, **тоді** Speed++
/прискорюємо/
4.2.3.2.4. logic=1;
- 4.2.4. **якщо** DirM=Forward, **тоді:** /обертання вперед/
перемикач (sw)
0: PORTC=SM[mode][0]; sw=1
1: PORTC=SM[mode][1]; sw=2
2: PORTC=SM[mode][2]; sw=3
3: PORTC=SM[mode][3]; sw=4
4: PORTC=SM[mode][4]; sw=5
5: PORTC=SM[mode][5]; sw=6
6: PORTC=SM[mode][6]; sw=7
7: PORTC=SM[mode][7]; sw=0
інакше: /обертання назад/
перемикач (sw)
0: PORTC=SM[mode][0]; sw=1
1: PORTC=SM[mode][1]; sw=2
2: PORTC=SM[mode][2]; sw=3
3: PORTC=SM[mode][3]; sw=4
4: PORTC=SM[mode][4]; sw=5
5: PORTC=SM[mode][5]; sw=6
6: PORTC=SM[mode][6]; sw=7
7: PORTC=SM[mode][7]; sw=0
- інакше (до п. 4.2):** N-- /пропускаємо такт таймера/
інакше (до п. 4): /мотор не рухається/
4.3. Speed=Smin; N=Smax; /скидання до початкових параметрів/
4.4. включення ШІМ-регулювання струму.
4.5. logic=1; DirM=Stop.
5. аналогічна модифікація пп.3-4 для координати Y.
6. при наступному такті таймера алгоритм продовжується з п.2.

У цьому алгоритмі використані змінні мають такий зміст:

Базовий режим. **X** та **Y** – значення кутів відхилень від заданого положення, які зчитуються з акселерометра по SPI-протоку; **Xcor** та **Ycor** – задані відхилення для фіксації платформи у довільному положенні, якщо ж дорівнюють 0, тоді платформа стабілізується в горизонтальній площині; **Xpos** та **Ypos** – відкалібровані величини кутів, що враховують задані значення відхилень; **Epos** – значення похибки показів акселерометра (на рис. 3 ця похибка позиціонування позначена сірою ділянкою); **InhibitionZone** – величина, що задає розмір для зони гальмування; **Smax** та **Smin** – задають межі прискорення (фактично їхня різниця відповідає кількості кроків двигуна, протягом яких має виконатися прискорення); **N** – змінна, що відповідає за прискорення (за допомогою неї реалізується логіка пропуску тактів таймера); **Speed** – визначає поточну швидкість; **mode** – вказує у якому з трьох режимів (однофазний, двофазний, півкроковий) має працювати мотор; **sw** – змінна для реалізації логіки перемикачів фазами двигуна; **SM** – двовимірний масив, у якому розміщені дані для реалізації логіки перемикачів фазами двигуна для вказаних трьох режимів; **PORTC** та **PORTD** – регістри мікроконтролера, що відповідають за вихідні рівні на його портах C та D, та до яких під'єднані драйвери крокових біполярних двигунів (для спрощення алгоритму передбачається, що решта виводів цих портів не задіюються);

Перехресний режим. **logic** – якщо змінна дорівнює 1, то це означає, що ми або ще не зайшли в зону гальмування, або перебуваємо у ній перший раз, якщо дорівнює 0 – ми виконуємо обертання двигуном у зоні гальмування; **K** – змінна, в якій зберігається кількість кроків прискорення, що необхідно здійснити мотору.

Стрибаючий режим. За допомогою змінних **DirM** та **DirMnew** фіксується напрямок руху мотора, як у попередньому такті таймера, так і у поточному. Якщо змінні містять різні значення, то це означає, що трапився різкий стрибок на протилежну за знаком координату, і необхідно плавно змінити напрямок обертання мотора. Тобто, мотор має спершу пригальмувати і зупинитися, а потім, поступово прискорюючись, почати обертатися у протилежний бік.

Для реалізації цього алгоритму на нашій фізичній установці необхідно попередньо визначити значення величин, що потребують початкової ініціалізації.

Тестування алгоритму проводилося за допомогою системи симулювання мікропроцесорних пристроїв Proteus та середовища розробника AVR Studio v.4.19. Програма керування була написана мовою Cі (компілятор AVR Toolchain). Для апробації алгоритму був виготовлений макет установки з такою конфігурацією (рис. 4):

- біполярні крокові мотори M42SP-6NKA 6.3 Ом, 7,5 град. (MITSUMI);
- передатна кількість редукторів: вісь X – 21,44; вісь Y – 4,7;
- драйвера моторів на базі мікросхем польових зборок IRF7105;
- акселерометр LIS302DL з передачею даних по SPI-протоколу;
- керуючий мікроконтролер ATmega32A;
- LCD-дисплей RAYSTAR RC1602A-GGN-CSX.

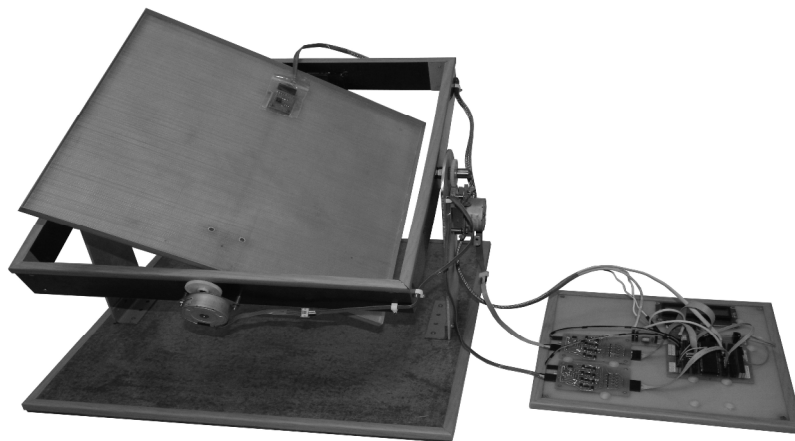


Рис. 4. Макет для натурної апробації алгоритму керування

Усі необхідні файли проектів для тестування алгоритму можна знайти на сайті автора <http://pavelchak.at.ua>.

Висновки

Розроблений програмний алгоритм керування дає можливість виконувати стабілізацію положення двохосової платформи для заданих кутів нахилу та є оптимізований для роботи у разі незначних відхилень та різких стрибків. Також його можна використати у разі певної його модифікації для подібних систем, що використовують крокові двигуни. Необхідно зазначити, що для досягнення високої точності позиціонування платформи потрібно використовувати акселерометри, що мають низькі шуми та велику роздільну здатність на виході (такі моделі доволі дорогі).

1. Волович А., Волович Г. *Интегральные акселерометры // Компоненты и технологии.* – 2002. – № 1. – С. 67–72. 2. Сысоева С. *Автомобильные акселерометры. Части 1-6 // Компоненты и технологии.* – 2005. – № 8–9; – 2006. – № 2–5. 3. Сысоева С. *Ключевые сегменты рынка МЭМС-компонентов. Акселерометры // Компоненты и технологии.* – 2010. – № 3. – С. 20–26. 4. А.Казакевич. *Акселерометры Analog Devices. Устройство, применение и непрерывное обновление // Компоненты и технологии.* – 2007. – № 5. – С. 46–50. 5. William C. Dunn. *Introduction to instrumentation, sensors, and process control.* – ARTECH HOUSE, INC. – 2006. – 331 p. 6. *Modern sensors handbook / edited by Pavel Ripka, Alois Tipek.* – ISTE Ltd. – 2007. – 518 p.

Ю. Іванов, Д. Пелешко, А. Ковальчук, М. Пелешко*
 Національний університет “Львівська політехніка”,
 кафедра інформаційних технологій видавничої справи
 *Державний університет безпеки життєдіяльності

ПОШУК ТРАЄКТОРІЙ РУХОМИХ ОБ’ЄКТІВ ДЛЯ АВТОМАТИЗОВАНИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

© Іванов Ю., Пелешко Д., Ковальчук А., Пелешко М., 2012

З розвитком інформаційних технологій до систем відеоспостереження висуваються все нові завдання, які полягають у розробці алгоритмів оброблення відеоданих для автоматизованих режимів роботи систем. Це зумовлено двома потребами: зменшення людського фактора у процесах, які забезпечують відеонагляд та безпеку. Виявлення найбільшої кількості інформативних характеристик для забезпечення безпеки чи усунення наслідків несанкціонованого порушення безпеки. Розроблено алгоритм, що дає змогу здійснювати супровід і обчислювати кількість транспортних засобів у відеопослідовності.

Ключові слова: зображення, траєкторія руху, бінарна маска, ерозія, відео послідовність, відеоряд, математична морфологія, системи відеоспостереження.

With the development of information technologies to the surveillance systems are put forward new tasks that lie in the development of algorithms for automated processing videodanyh modes of operation of systems. This is due to two needs: reducing human factor in the processes which provide videonahlyad and security. Identifying the greatest quantity informatyvnyh characteristics for security or unauthorized removal of the consequences of a security breach. In this paper, the algorithm that provides the support and calculate the number of vehicles in the video sequence.

Key words: image trajectory, binary mask erosion, video sequence video series, mathematical morphology, CCTV.

Вступ

Системи відеоспостереження є одними із ключових елементів сучасних систем відеоспостереження. Якість їх роботи безпосередньо впливає на роботу систем захисту та організації безпеки.

Серед основних вимог, які висуваються до сучасних систем відеоспостереження, є інтелектуалізація. Її суть полягає у тому, що система повинна не лише фіксувати відеопотік, а і здійснювати задану цільову обробку. Технічними вимогами до цієї обробки є її функціонування у автоматичному (або автоматизованому) режимі реального часу. При цьому вплив оператора повинен бути мінімальним.

Вочевидь розробка таких систем вимагає розроблення нових методів групових методів. Контекст групового методу тут стосується процедури декомпозиції задачі системи відеоспостереження. Залежно від цільового призначення основне завдання системи розбивається на підзадачі, кожна з яких розв’язують власними методами, зазвичай штучного інтелекту або DataMinig. Виділення і розв’язання найбільш інтелектуально ємних підзадач є основною запорукою успішного розроблення систем відеонагляду.

Аналіз і супровід рухомих об’єктів є одним з ключових завдань сучасних систем відеоспостереження. Оскільки такі системи постійно удосконалюються, то для підвищення їх рівня інтелектуалізації та адекватності важливою є побудова нових і модернізація існуючих алгоритмів, що призначені для супроводу рухомих об’єктів у полі зору камер відеоспостереження. У цій публікації реалізовано алгоритм супроводу, що базується на побудові морфологічної моделі.

Постановка задачі

Основним завданням роботи є розроблення моделі для визначення траєкторії руху об'єктів у відеопослідовностях реального часу. Об'єктом дослідження виступають процес руху у відеорядах. Основним засобом дослідження є морфологічні методи теорії цифрового оброблення зображень [2].

Математична модель системи пошуку траєкторій рухомих об'єктів. Виділення передньопланових об'єктів і побудова бінарної маски кадру відеопослідовності відбувається шляхом визначення абсолютного значення різниці збережено попереднього кадру $B(x, y)$ відеопослідовності з наступним $I(x, y)$. Отримане значення різниці порівнюється з пороговим значенням δ . Якщо різниця $|B(x, y) - I(x, y)|$ перевищує порогове значення, то піксель бінарної маски приймається як передньоплановий, інакше – фоновий (рис. 1, б).

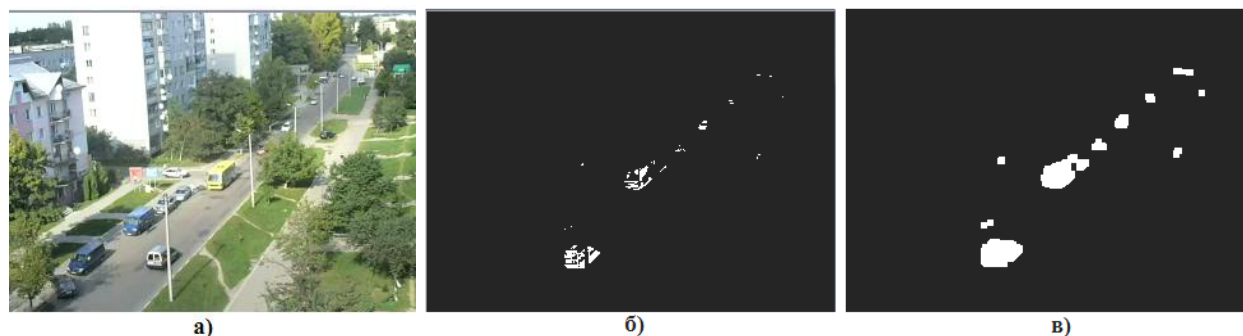


Рис. 1. Результат морфологічної обробки бінарної маски:
а – кадр відеопотоку; б – бінарна маска; в – морфологічна обробка бінарної маски

У практичній реалізації методологічна схема є такою.

Спочатку здійснюється мінімізація впливу шуму шляхом згладжування за допомогою фільтра Гауса з ядром розміром 3×3 кожного кадру відеопослідовності.

Надалі для початкового кадру відеопотоку приймаємо: $B_0 = I_0$, і для кожного наступного генеруємо бінарну маску $M_n(x)$, на якій нанесені усі пікселі, що належать рухомих об'єктам:

$$M_n(x) = \begin{cases} 1, & |I_n(x) - B_{n-1}(x)| > T; \\ 0, & |I_n(x) - B_{n-1}(x)| \leq T, \end{cases} \quad (1)$$

де T – порогове значення для виділення пікселя.

Для усіх пікселів кадру, що належать задньому плану, застосовуємо фільтр з нескінченним імпульсним відкликом:

$$B_n(x) = \begin{cases} B_{n-1}(x), & M_n(x) = 1; \\ \alpha I_n(x) + (1 - \alpha) B_{n-1}(x), & M_n(x) = 0, \end{cases} \quad (2)$$

де α – стала часу фільтра.

Пошук траєкторій рухомих об'єктів. До зображення бінарної маски послідовно застосовуються такі операції математичної морфології, як розширення та ерозія. Застосування цих операцій дає можливість згладити зображення. Структурним елементом для операцій розширення і ерозії є множина пікселів, які розташовані навколо опорного пікселя:

$$\begin{aligned} \delta_B(S) &= S \oplus B = \{s + b \mid s \in S; b \in B\}; \\ \varepsilon_B(S) &= S - B = \{B_h \subseteq S \mid h \in E\}, \end{aligned} \quad (3)$$

де $\delta_B(S)$ – розширення підмножини пікселів зображення S із структурним елементом B , $\varepsilon_B(S)$ – ерозія підмножини пікселів зображення S із структурним елементом B , E – дискретна площина, \mathbf{h} – вектор, що визначає позицію B на E , $B_h = \{b + h \mid b \in B\}$ перенесення елемента B вздовж вектора \mathbf{h} .

Коректний вибір конфігурації структурного елемента є для кожної відеопослідовності і окремих частин бінарної маски оптимізаційним завданням, і залежить від кількох умов та рекомендацій. Зокрема це такі:

- 1) бажано звільнити зображення бінарної маски від дефектів, наскільки це дають змогу зробити методи математичної морфології;
- 2) однією з основних умов є збереження топології представлених на зображенні маски об'єктів,
- 3) час опрацювання зображення повинен бути достатньо малим.

Приклад поетапного опрацювання кадру відеопослідовності за допомогою бінарної маски наведений на рис. 1.

Ініціалізація проводиться за першими двома кадрами відеопослідовності на основі принципу найближчого сусіда, тобто коли кожній точці траєкторії на наступних кадрах ставиться у відповідність найближча точка траєкторії на попередньому кадрі. Для будь-яких точок траєкторії можна визначити вектор різниці:

$$V_{i,t} = x_{i,t+1} - x_{i,t}, \quad (4)$$

де t – момент часу, коли спостерігається i -та точка траєкторії ($x_{i,t}$).

На основі різниці векторів, що входять і виходять з точки траєкторії $x_{i,t}$, можна визначити:

- гладкість напрямку у цій точці шляхом обчислення скалярного добутку цих векторів,
- гладкість швидкості на основі порівняння середньоарифметичного та середнього геометричного довжин цих векторів.

$$S_{i,k} = w \left(\frac{V_{i,t-1} \bullet V_{i,t}}{|V_{i,t-1}| |V_{i,t}|} \right) + (1-w) \left(\frac{2\sqrt{|V_{i,t-1}| |V_{i,t}|}}{|V_{i,t-1}| + |V_{i,t}|} \right), \quad (5)$$

де w – ваговий коефіцієнт, що приймається у межах від 0 до 1.

Алгоритм пошуку траєкторій. Алгоритм пошуку траєкторій на основі морфологічної моделі є таким:

КРОК 1. Створити k частин траєкторії за допомогою з'єднання найближчих сусідніх точок. Для першого кадру задати мітки вихідного масиву траєкторій. Для наступних кадрів присвоїти кожній траєкторії

$$T[i, t] = n, \quad (6)$$

де n – точка i -го об'єкта на кадрі t .

КРОК 2. Для усіх пар (i, k) визначити приріст гладкості. Циклічно виконувати обмін точок для досягнення максимального значення гладкості. Якщо обмін точок відбувся, то встановити прапорець обміну.

КРОК 3. Якщо на попередньому кроці відбувся обмін, то скинути прапорець обміну і повторити цикл.

Результати експериментів

Приклад роботи у програмній реалізації алгоритму наведено на рис.2. За еталонний приклад обрано аматорське відео. Основною мотивацією такого вибору було те, що така відеопослідовність буде не найкращої якості.

На наведеному рисунку у чорному прямокутнику визначаються траєкторії для відслідковування руху, але лише ті які є сутністю відеоспостереження. За розробленим алгоритмом відслідковуються траєкторії усіх рухомих об'єктів. Але частина їх відсівається як такі, що не потрапляють у список облікових. Цифри у цьому прямокутнику засвідчують кількість об'єктів, які рухались у зоні уваги за певною траєкторією.

Зона уваги (спостереження) визначається її прямокутними границями. Жодних обмежень на їх встановлення немає. Це означає, що зоною спостереження може бути увесь кадр відеопослідовності.

Обліковість траєкторії визначається напрямом і границями початку та кінця зони уваги. Тут допускаються будь-які варіації. В області чорного прямокутника на наведених рисунка кольоровими лініями із стрілками вказано обрані для обліку траєкторії.

Об'єкт вважається таким, який рухався за певною траєкторією, якщо він у визначеному напрямі перетнув обидві границі. Інакше він не облікується, а його траєкторія руху видаляється із списку облікових.

Траєкторія руху об'єкта, який пересік границю області спостереження, не відслідковується. І це незважаючи на те, що він продовжує рухатись.

За таких умов додатково з'являється можливість визначення ще однієї характеристики, а саме середньої швидкості руху об'єктів всередині зони спостереження. Хоча у цьому програмному рішенні визначення швидкості об'єктів не реалізоване, проте додати таку можливість не становить істотних труднощів.

Висновки

За заданих для обліку напрямів відслідковування робота алгоритму у реальному часі була безпомилковою. Експериментальне моделювання на інших аматорських відеоматеріалах також засвідчили успішне функціонування системи. Визначення залежності від освітленості та роботи алгоритму не проводили. Також не досліджували залежність ефективності роботи алгоритму від максимальної швидкості руху об'єктів всередині зони спостереження.

Програмна реалізація не містила жодних оптимізаційних процедур, що свідчить про певний запас швидкості у її фінальній реалізації.

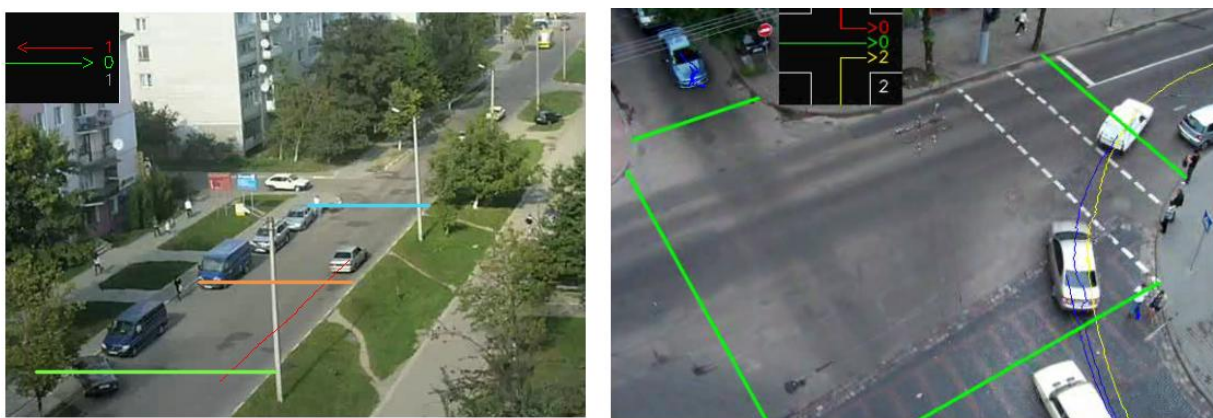


Рис. 2. Результат роботи алгоритму супроводу і обчислення кількості рухомих об'єктів

Запропонований метод визначення траєкторії рухомих об'єктів з достатньою точністю дає можливість визначати траєкторії рухомих об'єктів у відеопослідовностях. Водночас вимагається мінімальна кількість налаштувань, що спрощує використання на практиці програмно-апаратного рішення.

Достатня простота алгоритму забезпечує його повну реалізацію на апаратному рівні. Це, своєю чергою, дає можливість побудови програмно-апаратних, або повністю апаратних систем для розв'язання задач відеоспостереження у реальному часі.

1. Emilio Maggio, Andrea Cavallaro 'Object tracking. Theory and practice', A John Willey and Sons, Ltd. Publication, 2011. 2. Компьютерное зрение / Л. Шапиро, Дж. Стокман; Пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с. 3. Павлидис Т. Алгоритмы машинной графики и обработки изображений. – М.: Радио и связь, 1986. – 399 с. 4. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс ; пер. с англ. под ред. П. А. Чочиа. – М.: Техносфера, 2005. – 1072 с.

О. Гринів, Н. Притула, М. Притула

Центр математичного моделювання ІППММ ім. Я.С. Підстригача НАН України,
ТОВ “Математичний центр”

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СУМІСНОЇ РОБОТИ ГАЗОСХОВИЩ

© Гринів О., Притула Н, Притула М., 2012

Розглянута проблема оптимізації сумісної роботи газосховищ у складі газотранспортної системи України. Поставлені задачі і запропоновано алгоритми їх розв’язування. Отримані попередні оцінки потенціалу оптимізації роботи газосховищ та факторів впливу на величину таких оцінок.

Ключові слова: фільтрація, математична модель, оптимізація, газосховище, режим роботи, технологічна схема.

In this paper, we consider the problem of optimization of compatible work of gas storages as a part of the gas transport system of Ukraine. We have carried out statements and have offered the algorithms of their solving. We have also received the previous estimates of potential of optimization of work of gas storages and factors of influence on size of such estimates.

Key words: filtration, mathematical model, optimization, gas storage, operating mode, technological scheme.

Вступ

Підземні газосховища в складі газотранспортної системи (ГТС) України – фактор надійності її функціонування та гарантія стабільного забезпечення транзиту газу у Європу, особливо за умов різкого наростання споживання та появи нештатних ситуацій [1–2]. Об’єктом дослідження є група технологічно пов’язаних підземних газосховищ. Сумісна ефективна експлуатація газосховищ вимагає узгодження режимів їх роботи. Для формування оцінки потенціалу оптимізації потрібно розробити математичну модель роботи газосховищ в складі ГТС та поставити відповідні задачі оптимізації [3–6]. Технологічна схема, яка пов’язує газосховища з ГТС, дозволяє багатоваріантність їх сумісної експлуатації. Багатоваріантність роботи ще пов’язана з розподілом об’ємів газу між сховищами. Вибір варіантів ефективної експлуатації за паливно-енергетичними затратами та знаходження максимальної сумарної піковості системи ПСГ разом з ГТС вимагає сформулювати інтегральну критеріальну оцінку наявного режиму ГТС. Це дасть змогу побудувати алгоритми оптимального завантаження ПСГ на етапі нагнітання та відбирання газу.

Існує проблема розподілу між газосховищами планового об’єму газу для його зберігання, який є меншим за сумарний максимальний об’єм активного газу, за умови забезпечення максимальної ефективності функціонування ГТС.

1. Про піковість та оптимальність роботи газосховищ

Сумарний максимально можливий об’єм відбирання газу із газосховища називають його піковістю. Розрізнятимемо миттєві піковості та піковості, зараховані до інтервалу часу $[0, t]$. У загальному випадку піковість (миттєву чи за інтервалом часу) будемо пов’язувати з пластовим тиском P_p та тиском у магістральному газопроводі P_m , з яким технологічно пов’язане газосховище. Ще розрізнятимемо піковість по газозбірному пункті $Q_{\max}(P_p, P_g)$, $Q_{\max}(P_p, P_g, t)$ (g -піковість) та по магістральному газопроводу (m -піковість) $Q_{\max}(P_p, P_m)$, $Q_{\max}(P_p, P_m, t)$. m -піковість враховує параметри дотискувальної компресорної станції та тиск у магістральному газопроводі.

У пластах-колекторах газ зберігається під високим тиском. Для ефективного забезпечення контрактних умов на транспортування газу як для зовнішніх, так і для внутрішніх споживачів, в кожен момент часу, під час відбирання газу, потрібно цю енергію оптимально використовувати для підтримки роботи ГТС в оптимальному режимі. Варто очікувати, що така стратегія відбирання газу і забезпечить максимальну піковість роботи газосховищ.

2. Математична модель групи газосховищ. Постановка задач

Математична модель газосховищ, які інтегровані з ГТС, об'єднує моделі всіх технологічних об'єктів, які беруть участь у відбиранні та нагнітанні газу, зокрема набір з'єднувальних трубопроводів із запірною арматурою. Структура сукупної системи зображається граф-схемами, які поєднують технологічні схеми ПСГ з технологічною схемою газотранспортної системи.

Задана граф-схема $G_s(x, y) = G_p(x, y) \mathbf{U}(\mathbf{U}_{i=1}^5 G_i(x, y))$, де $G_i(x, y) (i=1, \dots, 5)$ – граф-схеми технологічних схем вибій-магістраль підземних газосховищ; $G_p(x, y)$ – підграф графу технологічної схеми ГТС $G(x, y)$, який технологічно пов'язаний із сховищами газу. Для всіх об'єктів відомі моделі їх газових потоків.

При постановках задач вважатимемо, що встановлення темпу використання наявного активного газу підпорядковане єдиній диспетчерській службі, основна ціль якої – забезпечення роботи ГТС в оптимальному режимі. Ця служба має інформацію про планові поставки газу на зберігання його в газосховищах та річний баланс газу (видобуток, поставки, транзит та споживання для власних потреб ГТС).

Для оцінки потенціалу сумісної роботи газосховищ варто спершу розглянути сумісну роботу двох газосховищ. За вхідні виберемо такі дані:

- Q_Σ – сумарний об'єм активного газу;
- $q_i^p = f(Q_i^a)$ – функціональний зв'язок між об'ємами активного газу в i -му сховищі та його піковістю.

Задача 1. Знайти таке представлення $Q_\Sigma = Q_1^a + Q_2^a$, щоб $f(Q_1^a) + f(Q_2^a) \rightarrow \max$.

Розв'язок такої задачі зводиться до розв'язання рівняння

$$\frac{d}{dQ_1^a} (f_1(Q_1^a) + f_2(Q_\Sigma - Q_1^a)) = 0,$$

відносно Q_1^a за умов

$$Q_{\min}^a \leq Q_i^a \leq Q_{\max}^a \quad (i=1, 2).$$

При виконанні в знайдений точці достатньої умови на максимум, сумарна піковість газосховища буде

$$q_{1,2}^p = f_1(Q_1^a) + f_2(Q_\Sigma - Q_1^a).$$

Робота газосховищ у пікових режимах, зазвичай, відбувається на окремих слабко прогнозованих інтервалах часу, які становлять незначну частину часового інтервалу роботи газосховищ. Якщо до умов попередньої задачі додається ще умова на сумарний відбір газу $q(t)$ на заданому інтервалі часу $[0, t]$, то виникає задача.

Задача 2. Знайти такий розподіл об'ємів відбирання газу між газосховищами, щоб під час відбирання залишковий буферний газ в сховищах забезпечував його піковість.

Інакше кажучи, потрібно знайти таке представлення $q(t) = q_1(t) + q_2(t)$, щоб

$$f(Q_1^a - q_1(t)) + f(Q_2^a - q_2(t)) \rightarrow \max (\forall t \in [0, t]),$$

де $q_1(t), q_2(t)$ – відбори із газосховищ.

Задача зведеться до аналізу рівняння

$$f(Q_1^a - q_1(t)) = f_1(Q_1^a - q_1(t)) + f_2(Q_\Sigma - Q_1^a - (q - q_1(t))).$$

Ми знову приходимо до розв'язування рівняння

$$\frac{d}{d(Q_1^a - q_1(t))} (f(Q_1^a - q_1(t))) = 0.$$

Розв'язавши рівняння для дискретних значень часу (вважаємо, що $q(t)$ протягом доби є стабільним), отримаємо розв'язок задачі.

Доволі часто стараються відібрати весь активний об'єм газу за заданий час. Маючи для кожного газосховища функцію піковості, досить просто знайти час, починаючи з якого сховище потрібно переводити в піковий режим роботи.

Розглянемо інший клас задач. Він стосується експлуатації газосховищ з мінімальними паливно-енергетичними ресурсами. Вважаємо, що з точки безпеки експлуатації газотранспортної системи, в загальному випадку, забезпечення піковості на певних інтервалах часу є важливішим за оптимальність роботи газосховищ. Важливо встановити зв'язок цих двох понять.

Задача 3. Побудувати на заданому часовому інтервалі часу $[0, t]$ окремо такі стратегії відбирання газу із сховищ, щоб забезпечити

- сумарний відбір $q(t)$;
- Q_i – сумарні відбори газу для кожного із i -го сховищ газу;

які б були оптимальними за паливно-енергетичними затратами.

Ця задача вимагає встановлення такого функціонального зв'язку:

$$Q_i^p = f_i(Q_i^a, q_i, P_m),$$

де Q_i^p – витрата паливного газу; q_i – відбір газу; P_m – тиск у магістральному газопроводі (у кінці відводу газу із сховища).

У загальному випадку вказана функція є дещо складнішою і залежить ще й від темпу відбирання газу протягом попереднього періоду. Ця ж функція дає змогу оцінити оптимальність за паливно-енергетичним критерієм та побудувати стратегію відбирання газу зі збереженням максимальної сумарної піковості.

Розв'язання третьої задачі (перша умова), в кожен моменту часу, за відомого сумарного об'єму відбирання газу, для двох газосховищ зведеться до знаходження розв'язку рівняння

$$\frac{d}{dq_i} (f_1(Q_i^a, q_i, P_{m1}) + f_2(Q_2^a, q - q_i, P_{m2})) = 0.$$

Якщо кількість сховищ газу є більшою ніж два, то потрібно використовувати необхідні умови екстремуму функцій з багатьма параметрами, що призведе до розв'язування систем нелінійних рівнянь.

Задача 4. Знайти таке представлення $Q_\Sigma = Q_1^a + Q_2^a$, відбори $q_1(t), q_2(t)$ та час t_m , щоб сумарна середня піковість

$$\frac{1}{t_m} \int_0^{t_m} (q_1(t) + q_2(t)) dt \rightarrow \max,$$

за умов

$$\int_0^{t_m} (q_1(t)) dt = Q_1^a, \int_0^{t_m} (q_2(t)) dt = Q_2^a.$$

Потрібно очікувати, що розв'язком задачі буде такий мінімальний час, за який обидва газосховища, працюючи в піковому режимі, відберуть активний газ одночасно. Дійсно, як легко перевірити, що у всіх інших випадках середня сумарна продуктивність газосховищ буде меншою.

Результатом задачі буде розв'язок рівняння (рівність часів відборів за пікових відборів)

$$\int_0^{Q_1^a} f_1^{-1}(q) dq = \int_0^{Q_2^a} f_2^{-1}(q) dq,$$

де f^{-1} – функція обернена до f .

Задача 5. За умов задачі 1 побудувати таку стратегію відбирання газу, щоб залишковий сумарний об'єм активного газу забезпечував максимальну піковість.

3. Розв'язування задач

Розв'язати першу задачу можна двома способами: перший – з використанням необхідної умови екстремуму, який зведеться до розв'язування систем нелінійних рівнянь, другий – алгоритмічний, який заданий об'єм активного газу розкладає по пластах газосховищ з певним кроком на об'єми газу в такий спосіб, щоб на кожному кроці газосховища досягали сумарної піковості.

Проведені числові експерименти підтвердили еквівалентність обох підходів. Запропонований алгоритм швидко працює, оскільки є лінійним за складністю відносно кількості газосховищ.

У реальних умовах експлуатації газосховищ встановлені функціональні зв'язки будуть дещо іншими. Тиск на вході в магістральний газопровід (відвід від ДКС ПСГ) формується, враховуючи оптимальність сумісної роботи ПСГ та ГТС. Часто складаються такі умови, що досягнення піковості роботи ПСГ є основною ціллю системи. У таких ситуаціях оптимальна робота системи не є пріоритетною.

Побудовані функції піковості (залежність між об'ємами активного газу в пласті-колекторі газосховища та максимальними відборами з нього) для всіх сховищ України. Розглянемо приклади, в яких використано реальні дані.

Приклад 1. Розглянута група технологічно поєднаних сховищ. На рис. 1 і 2 наведені графіки функцій піковості двох газосховищ.

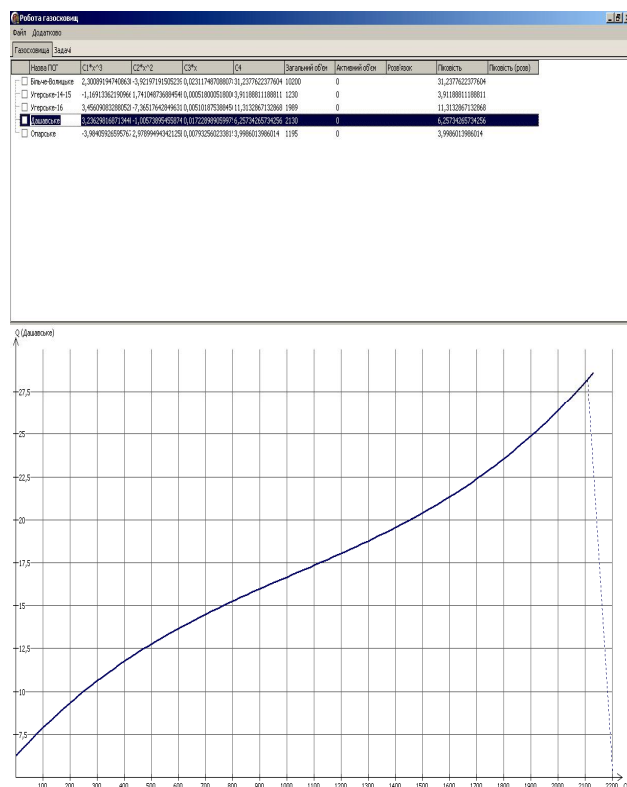


Рис. 1. Функція піковості Дашавського ПСГ

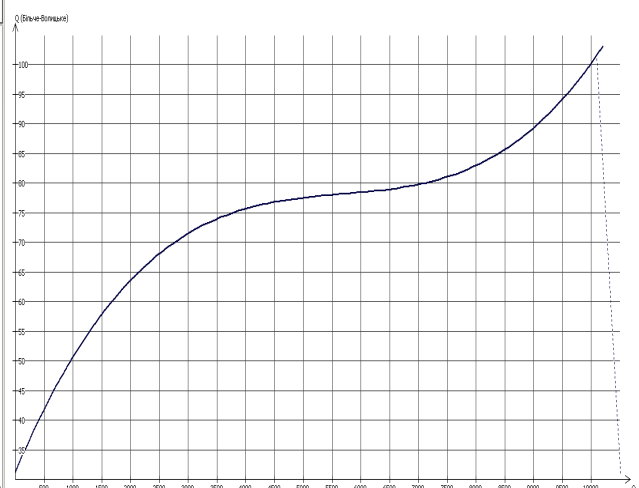


Рис. 2. Функція піковості Більче-Волицького ПСГ

У табл. 1 наведені коефіцієнти кубічної апроксимації функції піковості. У стовпці «Піковість» наведені значення піковості по кожному ПСГ у разі відсутності активного газу.

Таблиця 1

Оптимальний розподіл 12545.00 млн. м³ газу між 5 газосховищами

Назва ПСГ	C1*x^3	C2*x^2	C3*x	C4	Загальний об'єм	Активний об'єм	Розв'язок	Піковість	Піковість (розв)
<input checked="" type="checkbox"/> Більче-Волицьке	2,300891947408631	-3,92197191505235	0,02311748708807	31,2377622377604	10200	0	6386	31,2377622377604	78,8456538653109
<input checked="" type="checkbox"/> Угерське-14-15	-1,16913362190966	1,74104873688454	0,00051800051800	3,91188811188811	1230	0	980	3,91188811188811	10,1367685698438
<input checked="" type="checkbox"/> Угерське-16	3,456090832880521	-7,36517642849631	0,005101875388451	11,3132867132868	1989	0	1989	11,3132867132868	21,2666666666664
<input checked="" type="checkbox"/> Дашавське	3,236298168713441	-1,00573895455874	0,01722898905997	6,25734265734256	2130	0	2130	6,25734265734256	28,6000000000002
<input checked="" type="checkbox"/> Опарське	-3,98405926595767	2,978994943421251	0,00793256023381	3,9986013986014	1195	0	1060	3,9986013986014	11,0092356341701

Рис. 3. У виділеному стовпці «Розв'язок» наведений оптимальний розрахований розподіл активного газу між сховищами

У останньому стовпці таблиці містяться значення піковості по кожному ПСГ при знайденому розподілі газу між сховищами. Загальна піковість становитиме 149,85 млн. $\frac{м^3}{добу}$.

Приклад 2.

Таблиця 2

Розподіл об'ємів газу під час його відбирання

Назва ПСГ	C1*x^3	C2*x^2	C3*x	C4	Загальний об'єм	Активний об'єм	Розв'язок	Піковість	Піковість (розв)
<input checked="" type="checkbox"/> Більче-Волицьке	2,300891947408631	-3,92197191505235	0,02311748708807	31,2377622377604	10200	5039	3146	77,5813708029888	72,3126583944705
<input checked="" type="checkbox"/> Угерське-14-15	-1,16913362190966	1,74104873688454	0,00051800051800	3,91188811188811	1230	369	0	5,88624708624709	3,91188811188811
<input checked="" type="checkbox"/> Угерське-16	3,456090832880521	-7,36517642849631	0,005101875388451	11,3132867132868	1989	9	0	11,3591441858028	11,3132867132868
<input checked="" type="checkbox"/> Дашавське	3,236298168713441	-1,00573895455874	0,01722898905997	6,25734265734256	2130	636	2130	13,9793741641923	28,6000000000002
<input checked="" type="checkbox"/> Опарське	-3,98405926595767	2,978994943421251	0,00793256023381	3,9986013986014	1195	6	783	4,04630314326545	10,123642672993

Рис. 4. У виділених стовпцях містяться значення фактичних об'ємів акумульованого газу у кожному ПСГ на кінець відбору та значення функцій піковості.

В обох прикладах сумарний об'єм газу, який нагнітався в сховища, становив 6 059.00 млн. $\frac{м^3}{добу}$. Загальна піковість становить 112,852 млн. $\frac{м^3}{добу}$.

Таблиця 3

Розподіл об'ємів газу в процесі його відбирання

Назва ПСГ	C1*x^3	C2*x^2	C3*x	C4	Загальний об'єм	Активний об'єм	Розв'язок	Піковість	Піковість (розв)
<input checked="" type="checkbox"/> Більче-Волицьке	2,300891947408631	-3,92197191505235	0,02311748708807	31,2377622377604	10200	5039	3146	77,5813708029888	72,3126583944705
<input checked="" type="checkbox"/> Угерське-14-15	-1,16913362190966	1,74104873688454	0,00051800051800	3,91188811188811	1230	369	0	5,88624708624709	3,91188811188811
<input checked="" type="checkbox"/> Угерське-16	3,456090832880521	-7,36517642849631	0,005101875388451	11,3132867132868	1989	9	0	11,3591441858028	11,3132867132868
<input checked="" type="checkbox"/> Дашавське	3,236298168713441	-1,00573895455874	0,01722898905997	6,25734265734256	2130	636	2130	13,9793741641923	28,6000000000002
<input checked="" type="checkbox"/> Опарське	-3,98405926595767	2,978994943421251	0,00793256023381	3,9986013986014	1195	6	783	4,04630314326545	10,123642672993

Рис. 5. У виділених стовпцях містяться значення розрахованих оптимальних об'ємів акумульованого газу у кожному ПСГ на кінець відбору та значення функцій піковості

Загальна піковість становитиме 126,261 млн. $\frac{м^3}{добу}$. Зазначимо, що для досягнення максимальної піковості не потрібно використовувати пласти Угерських сховищ.

Висновки

Результати роботи запропонованого алгоритму перевірено аналітичними (точними) методами. Незначна неточність роботи алгоритму породжується його дискретністю. Міняючи крок роботи алгоритму, можемо встановити межі точності отриманого результату. У проведених числових розрахунках зміною кроку алгоритм знайшов точний розв'язок. Швидкодія алгоритму слабко залежить від кількості ПСГ, оскільки його складність є лінійною відносно кількості газосховищ.

1. Тетерев И.Г., Шешуков Н.Л., Нанивский Е.М. Управление процессами добычи газа. – М.: Недра, 1981. – 248 с. 2. Вечерік Р.Л., П'янило Я.Д., Притула М.Г., Хасецький Ю.Б. Математичне моделювання процесу руху газу в системі пласт підземного сховища газу – магістральний газопровід // *Нефть и газ*. – 2004. – № 6. – С. 83–89. 3. Вечерік Р.Л., П'янило Я.Д., Притула М.Г., Хасецький Ю.Б. Математичний аналіз акумулюючої здатності газонесних пластів ПСГ // *Нафтова і газова промисловість*. – 2005. – № 6. – С. 55–59. 4. П'янило Я.Д., Притула М.Г. Дослідження впливу параметрів пласту та привибійної області свердловини на розрахунок дебіту свердловини. // *Комп'ютерна інженерія та інформаційні технології* // *Вісник Держ. ун-ту "Львівська політехніка"*. – 2002. – № 392. – С. 45–49. 5. Вечерік Р.Л., П'янило Я.Д., Притула М.Г., Хасецький Ю.Б. Математичне моделювання процесу руху газу в системі пласт підземного сховища газу – магістральний газопровід // *Нефть и газ*. – 2004. – № 6. – С. 83–89. 6. Лопух Н.Б., П'янило Я.Д., Притула М.Г., Притула Н.М. Розрахунок початково – граничних умов в задачах фільтрації газу в пористих середовищах // *Вісник Нац. ун-ту "Львівська політехніка: Комп'ютерні науки та інформаційні технології"*. – Львів, 2009. – № 638. – С. 239–243.

УДК 519.81+519.863+004.942

О. Верес, Ю. Верес

Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

РОЗПОДІЛ ОБМЕЖЕНИХ РЕСУРСІВ ЗАСОБАМИ СППР

© Верес О., Верес Ю., 2012

Описано методології планування потреб у ресурсах. Проаналізовано програмні комплекси систем розподілу ресурсів і обґрунтовано потребу в розробленні своєї системи підтримки прийняття рішень. Проведено тестування прототипу СППР розподілу обмежених ресурсів, яка реалізує розроблені моделі та алгоритми.

Ключові слова: метод аналітичної ієрархії, обмежений ресурс, пріоритет, прийняття рішень, система підтримки прийняття рішень.

In the article describes the methodology of planning resource needs. The analysis of software systems and resource allocation systems proved the need to develop their decision support systems. Testing of the prototype DSS allocation of scarce resources, implementing the developed models and algorithms.

Key words: analytic hierarchy method, limited resources, priority, decision-making, Decision Support System.

Вступ. Загальна постановка проблеми

Без відповідних бізнес-процесів, стандартів і інструментів ефективний розподіл і постачання обмежених ресурсів є неможливим. Навіть, навпаки, неефективне управління ресурсами може значно збільшити ризик невиконання проектів. Прийняття рішень – підґрунтя будь-якого управління.

Система підтримки прийняття рішень з розподілу обмежених ресурсів забезпечує широку інформованість і свідоме прийняття рішень із розподілу і постачання ресурсів. Ефективний розподіл ресурсів є рушійною силою конкурентоспроможності підприємства.

Використання систем підтримки прийняття рішень у практиці управління виробничою фірмою виявляє її внутрішні резерви, що дає змогу досягнути кращих результатів з меншими витратами. Застосування таких систем є підґрунтям для прийняття правильних рішень.

Сучасний стан проблем, пов'язаних із розв'язанням задач автоматизації і управління в промислових системах, спонукає до розроблення і застосування нових математичних методів, комп'ютерних та інформаційних технологій. Водночас практично неможливо обійтися без наявності системного, гнучкого, багатofункціонального, універсального математичного і програмного забезпечення.

Проте в будь-якому випадку, зазвичай, системи підтримки прийняття рішень (СППР) з розподілу обмежених ресурсів мають використовувати положення математичного моделювання і програмування, враховувати багатокритерійність оптимізаційних задач, нелінійність моделей технологічних зв'язків, наявність обмежень, умови невизначеності під час прийняття рішень, неперервне оновлення інформації, унеможливити суб'єктивність і трудомісткість процесу опитування спеціалістів під час експертного оцінювання.

Розвинуті у роботі наукові положення ґрунтуються на доробку багатьох учених, зокрема О'Лірі, В.М. Буркова, Т. Сааті, А.Н. Борисова, Е.Р. Вілюмса, В.М. Глушкова, І.В. Сергієнка [1–4].

Отже, актуальним науковим завданням є розроблення математичного забезпечення процесу розподілу обмежених ресурсів та удосконалення математичної моделі системи «постачальник-споживач», розроблення методів розв'язання задачі розподілу обмежених ресурсів в управлінні ІТ-проектами та у виробничих фірмах, а також створення програмних засобів для реалізації відповідної системи підтримки прийняття рішень.

Аналіз останніх досліджень та публікацій

Більшість перебоїв у процесі виробництва пов'язана зі запізненням надходження тих чи інших комплектуючих. Водночас створення надмірних страхових запасів приводить до заморожування значних оборотних коштів, що негативно позначається на ефективності бізнесу. Крім того, за надмірних запасів важко визначити, до якої партії належить елемент, який перебуває у вже укомплектованому готовому продукті. А це буває необхідно під час виявлення причин виробництва бракованої продукції.

Для подолання таких проблем розроблено методологію планування потреби у матеріалах MRP (Material Requirements Planning). Її головна ідея полягає у плануванні постачань, що забезпечує наявність будь-якої облікової одиниці товарно-матеріальних цінностей, необхідних для виробництва виробів і/або постачань товарів споживачам у потрібний час і у потрібній кількості. Методологія MRP реалізується за допомогою комп'ютерних програм, що дають змогу скласти оптимальний план постачання комплектуючих у виробничий процес або товарів, що підлягають відвантаженню за замовленнями покупців. Оптимізацію плану постачань можна проводити за різними критеріями, але базовим принципом є контроль заданого рівня реально необхідних у кожен момент запасів [5, 7, 8].

Отже, MRP-система – це сукупність комп'ютерних програм, призначених для складання детального календарного плану постачань товарно-матеріальних цінностей, потрібних для забезпечення виробничого процесу або відвантаження товарів за замовленнями покупців, що забезпечує оптимальний рівень стану запасів у будь-який момент заданого періоду [5].

Системи підтримки прийняття рішень (СППР), які реалізують методологію MRP, забезпечують планування процесу закупівель потрібних товарно-матеріальних цінностей у постачальників, ґрунтуючись на рівні наявних запасів, вже розміщених замовлень на закупівлі, потреб виробництва (замовлень покупців, прогнозів продажу) з урахуванням нормативного рівня страхових запасів. Під час зміни у виробничій програмі, плані постачань товарів покупцям або відхиленнях від раніше прийнятих графіків виконання замовлень постачальниками, система видає рекомендації для зміни термінів обсягів закупівель та постачань для їхньої відповідності початковим графікам.

Основною вхідною інформацією MRP-системи є:

- описання стану матеріалів;
- програма виробництва (збуту) продукції (товарів);
- перелік складових компонентів остаточних продуктів.

Описання стану матеріалів (Inventory Status File) для виробничих фірм містить вичерпні відомості про всі матеріали та комплектуючі, необхідні для виробництва усіх потрібних видів продукції. Для кожної номенклатурної позиції задається наявний запас, вже зарезервована для тих або інших цілей кількість матеріалу (товару), обсяг вже розміщених у постачальників замовлень на постачання, орієнтовні терміни виконання замовлень постачальниками, мінімально необхідний обсяг страхових запасів. Вказуються ціни й інша додаткова інформація, наприклад, терміни придатності, максимальний обсяг запасу, мінімальний і максимальний розміри партії постачання тощо.

Виробнича програма (Master Production Schedule) є графіком випуску продукції або постачань товарів згідно із замовленнями покупців на планований період.

Перелік складових остаточного продукту (Bills of Material File) для виробничих підприємств є списком матеріалів і їхньої кількості, потрібної для виробництва кінцевого продукту. Для торгових фірм список складових може бути потрібним для опису комплектів товарів, що входять у різні замовлення.

Ґрунтуючись на вказаній інформації, система здійснює необхідні розрахунки і формує різноманітну звітність.

Основним звітом MRP-системи є План замовлень (Planned Order Schedule), у якому містяться дані про те, яку кількість кожного матеріалу потрібно замовити у кожен період часу впродовж терміну планування. План замовлень є основою для подальшої роботи з постачальниками. Крім того, він визначає внутрішню виробничу програму випуску напівфабрикатів і комплектуючих.

При змінах у первинному плані формується звіт змін до плану замовлень (Changes in Planned Orders). Він показує, які замовлення треба відмінити, змінити, затримати або перенести на інший період.

Звіт про «вузькі місця» планування (Exception Report) призначений для того, щоб завчасно вказати на ті періоди часу впродовж терміну планування, в які може виникнути потреба зовнішнього втручання для точнішого (ручного) управління. Такі ситуації можуть виникнути, наприклад, у разі серйозної затримки того чи іншого постачання.

Звіт про прогнози (Planning Report) містить інформацію, яка використовується для складання прогнозів про можливу майбутню зміну обсягів випуску продукції. Ці дані є результатом аналізу поточного ходу виробничого процесу і звітів про продукти. Звіт про прогнози може використовуватися для довгострокового планування потреб у матеріалах [5].

На практиці використовують ще деякі методології.

MRP II (Manufactory Resource Planning) – система виробничого планування ресурсів, що поєднує виробниче, маркетингове, фінансове планування й логістичні операції.

DRP (Distribution Requirements Planning) – система планування відправлень і запасів готової продукції у дистрибутивних каналах, зокрема й у логістичних посередників.

DRP II (Distribution Resource Planning) – це сучасна версія системи планування, що використовує сучасніші й потужніші програмні модулі, алгоритми й моделі прийняття рішень.

VMI (Vendor Managed Inventory) – удосконалена версія системи управління запасами постачальником, що ґрунтується на нових інформаційних технологіях.

ERP (Enterprise Resource Planning) – система інтегрованого планування ресурсів, що дає змогу планувати всю діяльність підприємства.

CSRP (Customer synchronized resource planning) – система планування ресурсів, синхронізована зі споживачем, що використовує функціональність ERP, переорієнтовує планування від виробництва до кінцевого покупця.

Визначивши основні технології розподілу та планування ресурсів, що застосовують в сучасних виробничих фірмах, необхідно здійснити класифікацію систем підтримки прийняття рішень, щоб визначати тип та функції системи підтримки прийняття рішень розподілу обмежених ресурсів.

Конкурентоспроможність у сучасному світі потребує розв'язання задач оптимізації бізнес-процесів, скорочення видатків, підвищення якості продукції і лояльності клієнтів. Для розв'язання цих задач призначені інформаційні системи класу ERP/ERP II [6–8].

ERP-система (англ. Enterprise Resource Planning System – система планування ресурсів підприємства) – це інтегрована система на базі ІТ для управління внутрішніми і зовнішніми ресурсами підприємства (значимі фізичні активи, фінансові, матеріально-технічні і людські ресурси). Зазвичай ERP-системи будують із використанням централізованого сховища даних. Така система формує стандартизований єдиний інформаційний простір підприємства [6].

Концепція ERP стала розвитком простіших концепцій MRP (Material Requirement Planning – Планування матеріальних потреб) і MRP II (Manufacturing Resource Planning – Планування виробничих ресурсів).

Класична архітектура ERP-систем містить такі елементи:

- модель управління інформаційними потоками (ІП) на підприємстві;
- апаратно-технічна база і засоби комунікації;
- СУБД, системне і супровідне програмне забезпечення;
- регламент використання і розвитку програмних продуктів;
- служби, що забезпечують функціонування системи;
- користувачі програмних продуктів;
- набір програмних засобів, що автоматизують управління інформаційними потоками.

До основних функцій ERP систем належать:

- впровадження конструкторських і технологічних специфікацій, що визначають склад виробів, а також матеріальні ресурси і операції, необхідні для їхнього виготовлення;
- формування планів продажу і виробництва;
- планування потреб у матеріалах і комплектуючих, строків і обсягів постачань для виконання плану виробництва продукції;
- управління запасами і закупками: ведення договорів, реалізація централізованих закупівель, забезпечення обліку і оптимізації складських і цехових запасів;
- планування виробничих потужностей від збільшеного планування до використання окремих верстатів та обладнання;
- оперативне управління фінансами, зокрема складання фінансового плану і виконання контролю його виконання, фінансовий і управлінський звіт;
- управління проектами, зокрема планування етапів і ресурсів.

У цьому розділі ми коротко розглянемо основні ERP-вирішення, що використовуються на українських виробничих фірмах та підприємствах, а саме:

1) SAP Business One – одна із перших систем SAP. Є самостійним програмним комплексом, призначеним для виробничих фірм, що працюють у сферах торгівлі і надання сервісних послуг. Система орієнтована на автоматизацію роботи відділу продажу, закупівель, ведення складського обліку і фінансів.

2) Microsoft Dynamics AX (Ахарта) – комплексне ERP-вирішення, створене для середніх і великих компаній. Є повністю інтегрованим вирішенням.

3) Microsoft Dynamics NAV (Navision) – комплексне ERP-вирішення, яке містить передові і найсучасніші технології управління підприємством.

4) ERP HansaWorld Enterprise – сучасне ERP-вирішення, призначене для інтеграції бізнес-процесів будь-якого підприємства. У цій системі інтегровані: бухгалтерія, логістика, CRM, веб-магазин, документообіг і мобільні вирішення для ведення бізнесу.

5) Epicor iScala – сучасний ефективний інструмент управління різними галузями діяльності підприємства. Повністю враховує локальні особливості. Scala є гнучкою системою, що використовується більше, ніж у 90 країнах протягом багатьох років. Модулі Scala охоплюють такі аспекти бізнесу, як фінанси, матеріально-технічне забезпечення, управління виробництвом, сервісне обслуговування, ведення проектів і управління персоналом.

6) 1С:Предприятие 8.0 – об'єднує платформу і прикладні вирішення, розроблені для автоматизації діяльності організацій і приватних підприємств.

7) Галактика Business Suite – комплекс бізнес-вирішень, що дає змогу у єдиному інформаційному просторі виконувати типові і спеціалізовані задачі з управління підприємством, холдингом, групою компаній в умовах сучасної економіки.

Оскільки до складу багатьох сучасних ERP-систем входять компоненти, які дають змогу управляти проектами та розподіляти ресурси між проектами (та між роботами проекту), розглянемо програмний продукт Microsoft Project Server 2010, який призначений для управління проектами [7].

Microsoft Project Server 2010 використовує переваги SharePoint Server 2010 і надає організаціям гнучкі рішення з управління робочим процесом. Project Server 2010 об'єднує управління проектами та портфелями, даючи можливість розподіляти наявні ресурси та інвестиції відповідно до пріоритетів організації, ефективно керувати всіма типами роботи та візуалізації її результатів за допомогою багатофункціональних панелей моніторингу.

У Microsoft Project Server 2010 з'явилася нова функціональна можливість «Стратегія» для управління портфелем проектів (раніше входила в окремий продукт Project Portfolio Server 2007), в яку входить розроблення бізнес факторів, визначення їхніх пріоритетів та аналіз портфеля.

З появою Microsoft Project Server 2010 офіс управління проектами отримав не тільки інструмент з аналізу портфеля та вибору стратегії, але й інструмент з управління ресурсними (людськими) обмеженнями проектів у портфелі.

Microsoft Project Server 2010 пропонує низку кроків з управління ресурсними обмеженнями, а саме: сценарії «що якщо»; перепризначення ресурсів; додавання або видалення. Користувач може переглянути відомості про потреби в ресурсах.

Якщо нам потрібно змоделювати замовлення ресурсів (аналіз «що якщо»), треба задати необхідну кількість ресурсу і розподіл ресурсу здійснюється залежно від розрахованого рангу проекту, який ґрунтується на визначених пріоритетах бізнес-факторів. Також можна моделювати примусове включення і виключення проектів, змінювати дати початку проектів.

Для того, щоб створити бізнес-фактор у Microsoft Project Server 2010, треба створити бібліотеку імен бізнес факторів і їхніх коротких описів. За бажанням можна вибрати відділи компанії, які будуть відповідати за певний бізнес фактор. З одним відділом рекомендується пов'язувати не більше ніж 7–9 бізнес-факторів.

Переваги та недоліки систем розподілу ресурсів наведено у табл. 1. Нашу увагу будемо звертати на ті переваги, які стосуються розподілу ресурсів, зокрема розподілу обмежених ресурсів.

Таблиця 1

Переваги та недоліки систем розподілу ресурсів

Перевага	Назва системи							
	SAP Business One	Microsoft Dynamics AX (Axapta)	Microsoft Dynamics NAV (Navision)	HansaWorld Enterprise	iScala	1C 8.0	Business Suite	Project Server
1	2	3	4	5	6	7	8	9
Управління фінансовими ресурсами	+	+	+	+	+	+	+	+
Управління основними засобами	-	+	-	+	+	+	+	+
Бюджетування	+	+	+	+	-	+	+	+
Управління персоналом	-	+	+	-	+	+	+	+
Консолідація	-	+	-	+	-	-	-	-
Виробництво	+	+	+	+	+	+	+	+
Склад	+	+	+	+	+	+	+	-
Управління проектами	-	+	+	+	+	-	-	+
Взаємодія із клієнтами	+	+	+	+	-	+	-	-
Пріоритетний розподіл ресурсів	+	+	+	+	+	-	+	+

1	2	3	4	5	6	7	8	9
Врахування обмеженості ресурсів	-/+	-	-/+	-	-	-	-	-/+
Примітка: «+» означає повну підтримку переваги програмним продуктом; «-» – перевага не підтримується; «-/» – часткова підтримка заданої функції.								

У SAP Business One, Microsoft Dynamics NAV (Navision), Microsoft Project Server є часткове врахування обмеженості ресурсів. Під час пропорційного розподілу обмежених ресурсів – розглянутими системами пропонується або збільшити плани виробництва дефіцитних ресурсів, або переглянути вимоги споживачів.

Невирішені раніше частини загальної проблеми. Оскільки жодна із розглянутих сучасних систем управління та розподілу ресурсів не враховує можливого дефіциту (обмеженості) ресурсів – доцільним є розроблення та побудова (програмна реалізація) системи підтримки прийняття рішень розподілу обмежених ресурсів, яка міститиме розроблені математичні моделі та методи.

Цілі (завдання) статті

Основним завданням статті є імітаційне моделювання розподілу обмежених ресурсів та порівняльний аналіз результатів дослідження різноманітних принципів і методів розподілу.

Основний матеріал

У результаті проведених наукових досліджень описано, засобами теорії систем, процес розподілу обмежених ресурсів. Побудовано математичний опис системи «постачальник-споживач» та сформульовано критерій ефективності її функціонування, який полягає у мінімізації сумарних видатків функціонування системи [9].

Для задоволення попиту споживачів необхідно так організувати процес постачання ресурсів від постачальників до споживачів, щоб сумарні видатки функціонування системи були мінімальними. Потрібно знайти такі значення елементів кортежу контролінгу $V = \langle S_{pu}, W_{pm}, O_{mu}, \hat{T} \rangle$, за яких отримаємо оптимальний розв'язок задачі розподілу обмежених ресурсів.

Компонентами кортежу є: $S_{pur}^t, W_{pmr}^t, O_{mur}^t$ – кількості ресурсу r -го виду, що наявні на маршрутах відповідно (pu) , (pm) і (mu) , тобто величини ресурсів, що надходять до споживачів і на проміжні пункти до початку t -го інтервалу постачань; \hat{T} – множина інтервалів функціонування системи.

Для подальшого опису стану і динаміки системи введено вектор стану системи $s^t = \{X_{pr}^t, Y_{ur}^t, Z_{mr}^t\}$. Координати вектора стану – ресурси, наявні в кінці інтервалу часу t у постачальників, споживачів та на проміжних пунктах, відповідно. Початковий і кінцевий стани системи вважаємо заданими векторами s^0 і s^T .

Стан системи описано місцезоташуванням точки в просторі станів. Координати цієї точки – поточні ресурси у постачальників, споживачів і на проміжних пунктах. Розмірність простору станів системи $P \times U \times M$. Стан системи описано у вигляді вектора

$$s^t = (X^t, Y^t, Z^t), \quad (1)$$

де X^t, Y^t, Z^t – ресурси наявні в кінці інтервалу часу t у відповідно постачальників, споживачів та на проміжних пунктах.

Очевидно, що плани постачань відповідають на три запитання: в який момент треба організувати постачання ресурсу, якою має бути його кількість і звідки вигідно його постачати.

Оптимальним вважається такий розподіл, за якого Центр мінімізує сукупні витрати, що пов'язані із функціонуванням системи, тобто витрати на розміщення замовлень, формування ресурсів, їхнього зберігання і втрати від дефіциту.

У критерій ефективності введено складову g^T , що відображає можливі втрати від недопостачання споживачам ресурсу. У результаті критерій ефективності набув вигляду

$$J = f^T + g^T + w^T + g^T \rightarrow \min, \quad (2)$$

$$g^T = \sum_{t=1}^T \sum_{u=1}^U \sum_{r=1}^R g_{ur}^t, \quad (3)$$

де f^T – сумарні видатки формування ресурсів за час T ; g^T – сумарні видатки зберігання за час T ; w^T – сумарні видатки виробництва за час T ; g^T – можливі втрати від недопостачання споживачам ресурсу за час T ; g_{ur}^t – можливі втрати від недопостачання одиниці r -го виду ресурсу.

Тут

$$g^t = \sum_{u=1}^U \sum_{r=1}^R g_{ur}^t, \quad 0 \leq t \leq T. \quad (4)$$

Значення g_{ur}^t обчислено як добуток коефіцієнта ефективності a_{ur}^t використання u -м споживачем r -го виду ресурсу в t -му періоді та різниці між його попитом та фактично отриманою кількістю ресурсу

$$g^t = \sum_{u=1}^U \sum_{r=1}^R (a_{ur}^t \cdot (d_{ur}^t - n_{ur}^t)), \quad (5)$$

де d_{ur}^t – попит u -го споживача на r -й вид ресурсу в t -му періоді; n_{ur}^t – фактично надана u -му споживачу кількість r -го виду ресурсу в t -му періоді.

Оптимізація функціонування системи «постачальник-споживач» полягає у вирішенні двох головних проблем, а саме: проблеми розподілу обмежених ресурсів між споживачами оптимальним для Центру способом; проблеми оптимальної доставки обмежених ресурсів до споживачів. У такому разі ми розв'язуємо задачу розподілу ресурсів, оскільки для розв'язання задачі доставки відомі ефективні методи та алгоритми для розв'язку задач транспортного типу (наприклад, метод гілок і границь для розв'язання задачі комівояжера).

Розроблені алгоритми подані із використанням алгебри алгоритмів В.К. Овсяка [10], яка на відміну від вербального та блок-схемного описів, забезпечує точний опис, мінімізацію за кількістю дій (унітермів) та дослідження алгоритмів.

Загальний алгоритм розподілу ресурсів передбачає секвенції

$$S_1 = \overline{a_4; a_5}; \quad S_2 = \overline{a_3; S_1}; \quad S_3 = \overline{a_2; S_2}; \quad S_4 = \overline{a_1; S_3}.$$

Тут a_1 – визначення початкових даних, потрібних для здійснення розподілу обмежених ресурсів; a_2 – опитування експертів та побудова матриць переваг для кожного виду ресурсу; a_3 – обчислення пріоритетів споживачів для кожного з видів ресурсів; пропорційний розподіл обмежених ресурсів між споживачами згідно з обчисленими пріоритетами; a_4 – визначення рівня сукупних видатків та обчислення значення критерію ефективності; a_5 – обчислення вихідних даних: значення пріоритетів споживачів для кожного виду ресурсу; видатки функціонування системи «постачальник-споживач»; кількості ресурсів, які отримає кожен із споживачів; значення критерію ефективності функціонування системи «постачальник-споживач».

Підставивши замість $S_1 - S_4$ їхні секвенції, отримуємо таку формулу загального алгоритму розподілу ресурсів:

$$\overbrace{a_1; \overbrace{a_2; \overbrace{a_3; \overbrace{a_4; a_5}}}}} \quad (6)$$

Синтез алгоритму оптимального розподілу ресурсів, який застосовується в унітермі a_3 , виконано секвентним методом, який передбачає синтез секвенцій та елімінувань.

У результаті роботи алгоритму отримуємо: обчислені вектори пріоритетів споживачів $q^j = (q_1^j, q_2^j, \dots, q_n^j)$, для кожного виду ресурсу; кількості обмежених ресурсів, що надаються споживачам (пропорційно до пріоритету).

Для обґрунтування та застосування методу обчислення пріоритетів споживачів (крок 2 алгоритму розподілу обмежених ресурсів між споживачами) використано дерево цілей та множину споживачів ресурсу [3, 9, 11]. Кореню ієрархії відповідає генеральна мета розподілу обмеженого ресурсу, а саме: оптимальний розподіл обмеженого ресурсу. Відповідно до методу аналітичної ієрархії, на другому рівні ієрархії знаходяться аспекти. На рівні листя ієрархії знаходяться споживачі. Використовуючи метод попарного порівняння елементів ієрархії, побудовано матриці парних порівнянь для всіх вершин ієрархії, окрім листя (рівню листя відповідає множина споживачів). Для обчислення вектора пріоритетів споживачів щодо альтернатив та фокуса ієрархії здійснено ієрархічний синтез.

Обчисливши пріоритети альтернатив відносно фокуса ієрархії, отримано вектор пріоритетів споживачів $r_{(1)}^{(j)} = (r_{(1)1}^{(j)}, r_{(1)2}^{(j)}, \dots, r_{(1)n}^{(j)})$ для j -го виду ресурсу. Оскільки оцінювався ступінь видатків, то споживач із найбільшим пріоритетом має найбільший рівень видатків, як фактичних, так і прогнозованих. Для застосування сформованих пріоритетів в алгоритмі розподілу ресурсу між споживачами, треба, щоб споживач із найменшим рівнем видатків мав найвищий пріоритет. Для цього змінюємо пріоритет кожного з них за такою формулою:

$$q_i^j = \frac{1 - r_{(1)i}^{(j)}}{\sum_i r_{(1)i}^{(j)}}, \quad (7)$$

де q_i^j – нове значення пріоритету i -го споживача для j -го виду ресурсу; $r_{(1)i}^{(j)}$ – пріоритет i -го споживача для j -го виду ресурсу, обчислений за допомогою методу аналітичної ієрархії.

Отримуємо $q^j = (q_1^j, q_2^j, \dots, q_n^j)$ – вектор пріоритетів споживачів для j -го виду ресурсу.

Пріоритети споживачів застосовуються в алгоритмі розподілу ресурсу між споживачами.

Грунтуючись на розробленому математичному забезпеченні процесу розподілу обмежених ресурсів та на математичній структурі системи «постачальник-споживач», розроблено алгоритми розв'язання задачі розподілу обмежених ресурсів, що забезпечують мінімізацію сукупних видатків функціонування системи «постачальник-споживач» [9].

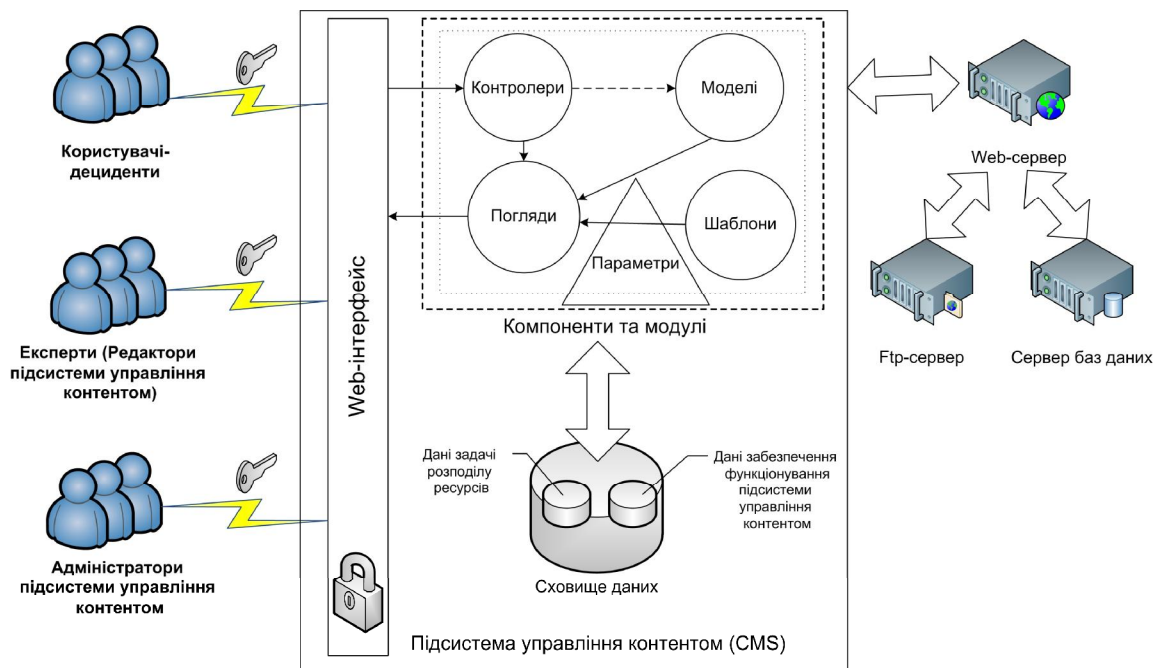


Рис. 1. Архітектура СПДР розподілу обмежених ресурсів

Розроблені алгоритми реалізовано у системі підтримки прийняття рішень. Розроблена узагальнена класифікація СППР дала змогу визначити тип СППР розподілу обмежених ресурсів [4]. Така система є комбінованою, оскільки містить у собі властивості систем різних типів.

Система підтримки прийняття рішень розподілу обмежених ресурсів складається з: підсистеми управління контентом; web-сервера; ftp-сервера; сервера баз даних (рис.1).

Користувачі системи взаємодіють з нею через веб-інтерфейс. Користувачів умовно поділяємо на три групи: користувачі, які приймають рішення (ці користувачі не змінюють ні структури системи, ні даних, якими вона оперує; вони користуються нею для прийняття рішення про надання споживачам певної кількості ресурсу); експерти або редактори підсистеми управління контентом (ці користувачі вносять у систему необхідну інформацію для прийняття рішення: заповнюють матриці переваг, необхідні для виконання алгоритму розподілу обмежених ресурсів); адміністратори підсистеми управління контентом (адміністратори мають повний доступ до системи, вони можуть змінювати дані, занесені до бази даних, змінювати (редагувати) ключові елементи системи). Кожна з груп користувачів має певний набір привілеїв (прав доступу до системи). Тому дії, які можуть здійснити адміністратори системи, не можуть здійснити користувачі (дециденти).

Опишемо призначення основних складових системи (табл. 2):

- web-сервер. Зв'язує всі компоненти системи в одне ціле. За його допомогою відбувається взаємодія користувачів із веб-інтерфейсом підсистеми управління контентом (CMS – Content Management System), та взаємодія CMS із сервером баз даних та ftp-сервером. Він здійснює компіляцію виконуваних файлів CMS.
- ftp-сервер. Призначений для зберігання файлів підсистеми управління контентом.
- сервер баз даних. На ньому розміщена база даних СППР розподілу обмежених ресурсів.
- підсистема управління контентом (CMS). Містить необхідні моделі для розв'язання задачі розподілу обмежених ресурсів. Надає користувачам (особам, що приймають рішення – ОПР) необхідні їм інтерфейси користувача. Експертам надає інтерфейси для заповнення таблиць бази даних СППР. Адміністраторам надає адміністративний інтерфейс для управління контентом та складовими частинами CMS.

CMS складається із компонентів, модулів та бази даних. У базі даних зберігаються таблиці, необхідні для забезпечення функціонування CMS, та таблиці, потрібні для зберігання даних, необхідних для розв'язання задачі розподілу ресурсів.

СППР із розподілу обмежених ресурсів є розширюваною, тобто адміністратори системи можуть розширювати функціональність CMS шляхом додавання до неї нових компонентів та модулів. Модулі та компоненти також називають розширеннями підсистеми управління контентом.

Модулі, зазвичай, використовуються для відображення інформації. Вони мінімально втручаються у функціонування системи і не впливають на її складові частини.

На відміну від модулів, компонент завантажується завжди у одну і ту ж позицію (спеціальна позиція для компонентів). На сторінці веб-інтерфейсу може бути лише один компонент. Вони використовуються для взаємодії із користувачами. Компоненти слугують, наприклад, для відображення проміжних обчислень, які отримуємо, використовуючи розроблені моделі розподілу ресурсів. Компоненти, як і модулі, мають набір різних параметрів, які необхідні для їхнього налаштування (наприклад, один і той же компонент може приймати різний вигляд, залежно від параметрів, з якими користувач завантажує веб-інтерфейс системи).

Кожен модуль та компонент підсистеми управління контентом побудований із використанням сучасної технології MVC (Model View Controller – Модель Погляд Контролер).

Прототип системи підтримки прийняття рішень розподілу обмежених ресурсів розроблений у вигляді веб-порталу. Для його функціонування потрібні такі програмні засоби: веб-сервер із можливістю виконання скриптів PHP 5 (підтримка інтерпретації файлів .php5); FTP сервер (він потрібний для завантаження файлів СППР на хостинг); POP3 / IMAP4 сервер (необхідний для відправлення повідомлень користувачам СППР), DNS сервер (потрібний для підтримання доменного імені веб-порталу), MySQL сервер із підтримкою таблиць у форматі InnoDB (це сервер управління базами даних у якому зберігається сховище даних СППР); phpMyAdmin (набір PHP скриптів, який використовується для адміністрування СУБД MySQL).

Призначення та функції розроблених компонентів СППР

Назва компоненту	Призначення	Функції
1	2	3
Перегляд бази даних	Відображення частини бази даних системи, яка призначена для зберігання даних розв'язання задачі розподілу обмежених ресурсів	Відображення споживачів обмеженого ресурсу для заданого періоду
		Відображення постачальників обмеженого ресурсу для заданого періоду
		Відображення проміжних пунктів для заданого періоду
		Відображення кількості обмеженого ресурсу певного типу спродукованого постачальниками у заданий період
		Відображення прогнозованих витраток функціонування системи у заданому періоді.
		Відображення вимог споживачів за кожним типом ресурсу у заданому періоді.
		Відображення рівня прибутковості кожного із споживачів для кожного типу ресурсу у заданому періоді.
		Відображення значення важливості аспектів у заданому періоді.
		Відображення усіх наявних періодів розподілу обмежених ресурсів та значень критеріїв якості, отриманих розробленими методами, методом прямого, оберненого та рівного розподілу обмежених ресурсів.
		Відображення усіх типів ресурсів, які підлягають розподілу.
Імітаційне моделювання	Генерування даних задачі розподілу обмежених ресурсів. Запис отриманих значень у базу даних. Очищення бази даних від значень.	Відображення типів витраток.
		Очищення частини бази даних, яке використовується для зберігання даних, необхідних для розв'язання задачі розподілу обмежених ресурсів.
		Генерування даних із наперед заданим розкидом значень.
		Заповнення бази даних числовими значеннями.
		Генерування заданої кількості періодів.
		Генерування заданої кількості споживачів для наявного періоду.
		Генерування заданої кількості постачальників для наявного періоду.
		Генерування заданої кількості проміжних пунктів для наявного періоду.
		Генерування заданої кількості типів ресурсів
		Імітаційне моделювання продукування постачальниками обмежених ресурсів.
		Імітаційне моделювання виникнення витраток (генерується значення витратку на одиницю ресурсу).
		Імітаційне моделювання виникнення у споживачів вимог на споживання обмежених ресурсів.
		Генерування прибутковості (значення прибутковості на одиницю ресурсу) споживачів для окремих типів обмежених ресурсів.
		Імітаційне моделювання оцінювання експертами аспектів, критеріїв та витраток споживачів.
Запис у базу даних типів витраток.		

1	2	3
Розв'язання задачі розподілу обмежених ресурсів	Розв'язання задачі розподілу обмежених ресурсів, розробленими та наявними методами. Відображення перебігу розв'язання задачі. Обчислення критеріїв якості функціонування системи.	Відображення перебігу розв'язання задачі для заданого періоду та типу ресурсу.
		Обчислення та відображення рівня видатків для заданого періоду.
		Розв'язання задачі розподілу обмежених ресурсів для усіх періодів.
		Запис у базу даних результатів розв'язання задачі розподілу обмежених ресурсів.
		Обчислення критерію якості функціонування системи для розв'язків отриманих розробленими методами та алгоритмами, та методами прямого, оберненого та рівного розподілу обмежених ресурсів.

Під час створення СППР використано такі засоби:

- HTML – мова розмітки гіпертексту, стандартна мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок створюються за допомогою цієї мови. Документ HTML опрацьовується браузером та відтворюється на екрані у звичному для людини вигляді.
- XML – розширювана мова розмітки, запропонована консорціумом World Wide Web (W3C). (ця мова використана для опису структури модулів та компонентів СППР).
- SQL – мова структурованих запитів, декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними базами даних, створення схеми бази даних і її модифікація, система контролю за доступом до бази даних. SQL використана для здійснення запитів і внесення змін до бази даних СППР.

Функціональним призначенням системи підтримки прийняття рішень розподілу обмежених ресурсів є допомога децидентові прийняти найефективніше рішення. Програма пропонує децидентові варіант розв'язання задачі розподілу обмежених ресурсів, а саме: пріоритети, які необхідно присвоїти споживачам для надання їм ресурсів; кількості ресурсу, які треба надати кожному із споживачів. Також програма показує хід розв'язання задачі, щоб децидент міг оцінити достовірність обчислень та за необхідності внести відповідні корективи у вхідні дані задачі.

Компоненти програми реалізовано у формі каталогу із відповідним набором файлів. Всього розроблено три компоненти: dbrevision; dbgen; mai.

Компонент dbrevision, призначений для перегляду бази даних СППР у зручному для децидента вигляді. Опишемо призначення файлів компонента dbrevision: controller.php – реалізація контролеру компонента; dbrevision.php – файл завантаження компонента; models\dbrevision.php – реалізація моделі компонента; views\dbrevision\view.html.php – реалізація погляду компонента; views\dbrevision\tmpl\default.php – реалізація шаблону погляду компонента.

Компонент dbgen призначений для здійснення імітаційного моделювання, а саме для генерування вхідних даних задачі розподілу обмежених ресурсів. Своєю чергою mai – містить алгоритми та моделі розв'язання задачі розподілу обмежених ресурсів. Структура каталогів та призначення файлів компонентів є аналогічними до компонента dbrevision.

Для виклику програми, треба у браузері завантажити веб-портал за такою адресою: <http://resource>. Після завантаження, децидент отримує головну сторінку порталу. На головній сторінці відображено меню системи, використовуючи яке, користувач має змогу перейти на потрібну йому сторінку (завантажити потрібний компонент) (рис. 2).

Для здійснення аналізу роботи програмної реалізації системи підтримки прийняття рішень розподілу обмежених ресурсів здійснено низку експериментів, використовуючи розроблений компонент імітаційного моделювання (КІМ). Він дає змогу отримати розв'язки задачі розподілу обмежених ресурсів за різних вхідних даних та порівняти між собою критерії якості функціонування системи «постачальник-споживач», отримані в результаті розв'язання задачі різними методами, як розробленими, так і наявними (прямого, оберненого та рівного розподілу).

Щоб здійснити імітаційне моделювання, в розробленій системі користувачу треба ввести вхідні дані задачі.

- Кількість періодів. Вводимо кількість періодів функціонування системи «постачальник-споживач». Їх може бути один і більше.
- Кількість споживачів. Ця змінна означає кількість споживачів обмежених ресурсів у кожному із періодів функціонування системи «постачальник-споживач».
- Кількість постачальників. Значення цієї змінної відповідає кількості постачальників обмежених ресурсів у кожному із періодів функціонування системи «постачальник-споживач». Постачальники продукують ресурси, потрібні споживачам.
- Кількість проміжних пунктів. Вводимо кількість проміжних пунктів, які існуватимуть у кожному із періодів. Проміжні пункти є одночасно і постачальниками ресурсу (продукують ресурси) і їхніми споживачами.
- Кількість типів ресурсів. Змінна означає кількість типів обмежених ресурсів, які будуть продукуватися постачальниками та споживатися споживачами.
- Мінімальна та максимальна кількість ресурсу, яка може бути вироблена постачальником. По суті це мінімальне та максимальне значення виробничих потужностей постачальників. Кожен із постачальників продукуватиме обмежені ресурси у кількості, яка більша за мінімальне значення змінної та менша за максимальне.
- Мінімальна та максимальна величина видатку (рівень видатку на одиницю ресурсу). Задаємо рівень видатків на одиницю виробленого ресурсу. Значення видатків генеруються окремо для кожного із споживачів за кожним із видів ресурсів, у межах від мінімального до максимального заданого значення.
- Максимальний рівень дефіциту. Рівень дефіциту обчислюється як відношення суми всіх вимог на ресурс до його сумарного виробленого обсягу. Вводиться максимальне значення рівня дефіциту, яке може бути на певний вид ресурсу. Розроблена система генерує вимоги споживачів на ресурси залежно від заданого рівня дефіциту у межах від 0 (дефіциту немає) до значення введеного користувачем.
- Мінімальний та максимальний рівень прибутковості на одиницю ресурсу. Споживачі, використовуючи обмежені ресурси, дають певний рівень прибутку. У системі генеруються значення прибутковості кожного зі споживачів на одиницю певного виду ресурсу. Це значення генерується в межах від мінімального до максимального значень, заданих користувачем.

Виконаємо низку експериментів та проаналізуємо отримані вихідні дані (табл. 3).

Ввівши у систему вхідні дані, даємо команду провести імітаційне моделювання та записати результати експерименту у базу даних. Після цього за допомогою розробленого компонента перегляду бази даних отримуємо результати – обчислені критерії якості функціонування системи «постачальник-споживач» для кожного із періодів, використовуючи різні методи (розроблені та наявні). У таблиці також подамо відсоток покращення результату розробленими методами (значення обчислюється, як відношення результатів, отриманих наявними методами, до результатів, отриманих розробленими методами). Рівень дефіциту у вхідних даних задається як відношення сумарних вимог на ресурс до його сумарної виробленої



Рис. 2. Зовнішній вигляд меню графічного інтерфейсу

кількості. Проведемо п'ять експериментів з різними вхідними даними. Причому у кожному подальшому експерименті підвищуватимемо дефіцитність ресурсу.

Таблиця 3

Вхідні дані для експериментів

Назва змінної	Числове значення	
Кількість періодів	10	
Кількість споживачів	10	
Кількість постачальників	10	
Кількість проміжних пунктів	10	
Кількість типів ресурсів	10	
Назва змінної	Мінімальне значення	Максимальне значення
Кількість ресурсу, виробленого постачальниками	1	10
Величина видатку	1	10
Прибутковість	1	3
Номер експерименту	Назва змінної	Числове значення
1	Рівень дефіциту	1,1
2	Рівень дефіциту	1,25
3	Рівень дефіциту	1,5
4	Рівень дефіциту	1,75
5	Рівень дефіциту	2

Обчислені середні значення критерію якості для розробленого методу, а також методів рівномірного, прямого і оберненого розподілів, наведено у табл. 4.

Таблиця 4

Середнє значення критерію якості та відсоток покращення результату

Порядковий номер експерименту	Метод обчислення критерію якості							
	Розроблені методи		Рівний розподіл		Прямий розподіл		Обернений розподіл	
	значення	%	значення	%	значення	%	значення	%
1	20763,45	-	20888,65	0,603	20889,14	0,605	20887,42	0,597
2	22119,59	-	22333,41	0,967	22342,72	1,009	22337,42	0,985
3	30609,65	-	30902,69	0,957	30946,94	1,102	30918,68	1,010
4	23501,56	-	23804,84	1,290	23740,62	1,017	23816,68	1,341
5	21102,33	-	21496,57	1,868	21486,89	1,822	21500,93	1,889

За результатами проведених експериментів (табл. 5) будуємо графік залежності рівня дефіциту від відсотка покращення результату (рис. 3).

Таблиця 5

Підсумкові результати експериментів

Порядковий номер експерименту	Відсоток покращення результату	Середній рівень дефіциту (%)
1	0,602	5
2	0,987	12,5
3	1,023	25
4	1,216	37,5
5	1,860	50



Рис. 3. Залежність покращення результату розробленими методами від рівня дефіциту

Проведемо ще один експеримент, у якому проаналізуємо пріоритети споживачів, обчислені різними методами (розробленими, методом прямого, оберненого та рівного розподілів).

За допомогою розробленої системи підтримки прийняття рішень згенеруємо вхідні дані задачі із такими параметрами:

- кількість періодів розподілу обмежених ресурсів – 3;
- кількість споживачів – 5;
- постачальників – 3;
- проміжних пунктів – 1;
- максимальний рівень дефіциту – 2;
- кількість виробленого ресурсу у межах від 1 до 10;
- величина витратків на одиницю ресурсу у межах від 1 до 10;
- рівень прибутковості на одиницю ресурсу у межах від 1 до 10;
- згенеровані вимоги споживачів на ресурси наведено у табл. 6

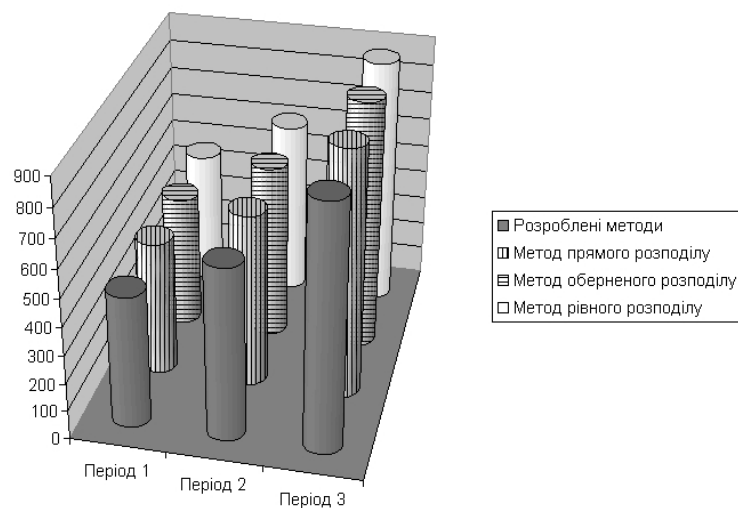
Розв'яжемо задачу розподілу обмежених ресурсів для кожного періоду та кожного виду ресурсу. У результаті отримаємо пріоритети споживачів, кількості ресурсів, які їм будуть надані та обчислені сумарні витатки (витатки у певному періоді для певного типу ресурсу). Порівнюємо результати експериментів, використовуючи різні методи обчислення пріоритетів, а саме: розроблені методи; методи прямого, оберненого та рівного розподілу.

Таблиця 6

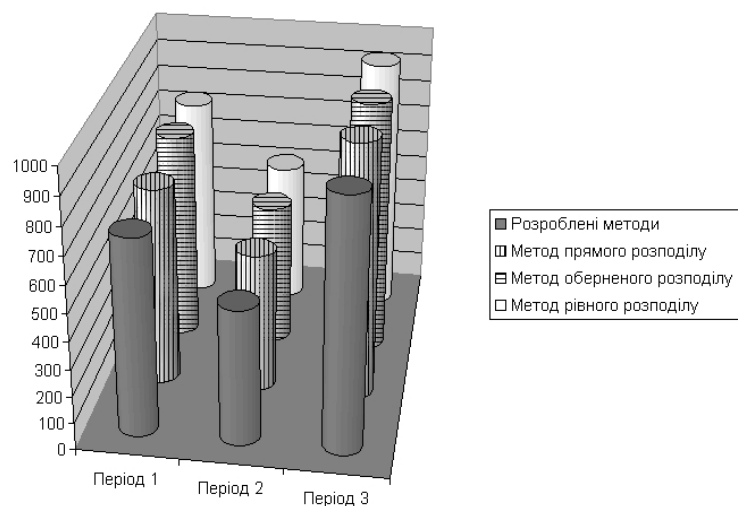
Вимоги споживачів на ресурси

	Споживач 1	Споживач 2	Споживач 3	Споживач 4	Споживач 5
Період 1					
Тип ресурсу 1	4,01	4,97	4,41	3,46	3,39
Тип ресурсу 2	6,7	6,31	4,76	4,62	5,64
Період 2					
Тип ресурсу 1	4,76	5,25	3,32	4,48	5,06
Тип ресурсу 2	4,59	4,61	4,45	2,95	4,66
Період 3					
Тип ресурсу 1	7,75	5,25	6,79	6,65	6,87
Тип ресурсу 2	8,62	5,19	6,05	6	5,34

Зобразимо графічно рівні видатків, обчислених у кожному із періодів для двох видів ресурсів (рис. 4).



a



б

Рис. 5. Рівні видатків для першого (а) і другого (б) типів ресурсу

Здійснивши експерименти, переконаємося у тому, що розроблені методи надають кращий результат за наявні у кожному із експериментів, що свідчить про їхню вищу ефективність порівняно із іншими.

За допомогою розроблених методів, отримуємо кращий результат, як за високих, так і за низьких рівнів дефіциту. Отже, вони застосовні для різних рівнів обмеженості ресурсів.

Висновки

Для досягнення поставленої мети здійснено порівняльний аналіз задач, моделей та методів розподілу обмежених ресурсів у різноманітних галузях народного господарства. Розроблено узагальнений формальний опис процесу прийняття рішення з розподілу обмежених ресурсів, що дало змогу побудувати адекватний формалізований опис основних етапів процесу прийняття рішень. Одержала подальший розвиток математична структура системи «постачальник-споживач».

Розроблено й досліджено модель розподілу обмежених ресурсів, що ґрунтується на побудові дерева цілей та споживачів ресурсів і обчислення значень їхніх пріоритетів засобами методу аналітичної ієрархії, тим самим підвищено ефективність прийнятих рішень системи «постачальник-споживач». Удосконалено критерій ефективності функціонування дворівневої системи «постачальник-

споживач», що дало змогу розв'язати задачу розподілу обмежених ресурсів з урахуванням можливих втрат від недопостачання ресурсу.

Розроблено архітектуру СППР розподілу обмежених ресурсів, яка ґрунтується на запропонованій узагальненій класифікації СППР. Описано принципи її функціонування. Основними складовими системи є: підсистема управління контентом; web-сервер; ftp-сервер; сервер баз даних. Створене прикладне програмне забезпечення реалізує розроблені математичне забезпечення та алгоритми. Програмна реалізація дала змогу здійснити імітаційне моделювання процесу розподілу обмежених ресурсів, що підтвердило ефективність розроблених методів та алгоритмів.

Подальші дослідження будуть присвячені розробленню інтелектуальної складової системи, ґрунтуючись на методах штучного інтелекту.

1. Бурков В.Н. Модели и механизмы распределения затрат и доходов в рыночной экономике / В.Н. Бурков, И.И. Горюхи, Д.А. Новиков, Б.С. Юсупов. – М. : Институт проблем управления, 1996. – 61 с. – (Препринт / Институт проблем управления).
2. Баркалов П.С. Задачи распределения ресурсов в управлении проектами / П.С. Баркалов, И.В. Буркова, А. В. Глаголев. – М.: ИПУ РАН, 2002. – 65с.
3. Саати Т. Принятие решений. Метод анализа иерархий. / Т. Саати. – М.: Радио и связь, 1993. – 278 с.
4. Верес О.М., Верес Ю.О., Катренко А.В. СППР з керування розподілом обмежених ресурсів // Інформаційні системи та мережі. Вісник НУ"Львівська політехніка", 2008, № 610. – С. 52–62.
5. Darryl V. Landvater, and Christopher D. Gray. MRP II Standard System. A handbook for Manufacturing Software Survival. John Wiley & Sons, Inc., 1989.
6. О'Лири Д. ERP системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение, эксплуатация / Д. О'Лири / Пер. с англ. Ю. И. Водяновой. – М. : ООО «Вершина», 2004. – 272 с.
7. Верес Ю. О. Розв'язання задачі управління запасами в системах розподілу обмежених ресурсів // Системний аналіз та інформаційні технології: Матеріали XI Міжнародної науково-технічної конференції (26–30 травня 2009 р., Київ). – К. : ННК "ІПСА" НТУУ "КПІ". – 2009. – С. 274.
8. Бородулин А. Н. Внутрифирменное управление, учет и информационные технологии. Учебное пособие. / А. Н. Бородулин, А. Ю. Заложнев, Е.Л. Шуремов. – М.: ЗАО «ПМСОФТ», 2006. – 340 с.
9. Верес Ю.О. Алгоритм розподілу обмежених ресурсів / Ю.О. Верес // Наукові праці Чорноморського державного університету ім. Петра Могили. Серія : Комп'ютерні технології. – 2010. – Вип. 130. – Т. 143. – С. 57–62.
10. Owsiak W. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych / Owsiak W., Owsiak A., Owsiak J. – Opole: Politechnika Opolska, 2005. – 275 s.
11. Катренко А. В. Системний аналіз / А. В. Катренко. – Львів: Новий світ, 2009. – 396 с.

Р. Кутельмах, Н. Павич, Х. Андрухів
 Національний університет “Львівська політехніка”,
 кафедра програмного забезпечення

АЛГОРИТМІЧНІ ТА ПРОГРАМНІ ЗАСОБИ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ В ОПЕРАЦІЙНІЙ СИСТЕМІ MAC OS X

© Кутельмах Р., Павич Н., Андрухів Х., 2012

Запропоновано алгоритмічні та програмні засоби опрацювання зображень в операційній системі Mac OS X. Забезпечено можливість коригування колірних характеристик зображення, змінюючи контрастність, чіткість, насиченість тіней, інтенсивність теплих відтінків, а також усунення ефекту червоних очей. Показані результати тестування запропонованих засобів, які підтверджують їх ефективність.

Ключові слова: алгоритмічні засоби, програмні засоби, опрацювання зображень, операційна система Mac OS X.

The paper describes image processing software and algorithms approaches and for Mac OS X. The solution provides the opportunity to adjust image color characteristics by changing the contrast, sharpness, shadows, warmth and also “red eyes” effect removal. There are proposed techniques testing results, which proves these techniques approaches efficiency.

Key words: algorithmic approaches, software approaches, image processing, Mac OS X

Вступ

Сьогодні доволі часто у багатьох користувачів комп'ютерної техніки виникає задача опрацювання зображень, фотографій, передусім для покращення їх якості, для виправлення дефектів, для надання їм особливого стилю та вигляду. Тому актуальними є дослідження та конкретні розробки щодо створення алгоритмічних та програмних засобів опрацювання зображень.

Під час опрацювання зображень важливо, щоб програмний засіб ефективно розв'язував поставлену задачу та був простим у використанні. Зокрема такі задачі можна розв'язувати в операційній системі Mac OS X. Оскільки програмний інтерфейс самої операційної системи Mac OS X для більшості користувачів є зовнішньо привабливим, зручним, зрозумілим, легким в освоєнні та користуванні, то актуальність розробки програмного продукту для опрацювання зображень в цій операційній системі є цілком обґрунтованою.

Огляд літературних джерел та публікацій

Операційна система Mac OS X є доволі популярною у користувачів і, згідно із статистикою [1], продовжує набувати все більшої популярності. Компанія-виробник Apple надає можливість розробникам представляти свої програмні продукти мільйонам користувачів по всьому світу. Компанія Apple пропонує безкоштовне програмне середовище розробки програм Xcode, що об'єднує потужний набір інструментів для створення сучасних високопродуктивних програм, які є доступними в App Store [2]. Користувачам потрібно лише ввести ключові слова, обрати зі списку запропонованих продуктів бажаний і завантажити та автоматично встановити його на свій пристрій, використовуючи онлайн-магазин App Store.

Основними компонентами Mac OS X є [3]:

- підсистема з відкритим кодом – Darwin (ядро Mach, набір утиліт BSD);
- середовище програмування Core Foundation (Carbon API, Cocoa API і Java API);
- графічне середовище Aqua (QuickTime, Quartz Extreme і OpenGL);
- технології CoreImage, CoreAudio і CoreData.

Нині відомо низку програмних продуктів, що дають змогу обробляти зображення в Mac OS X. Найвідоміші програми : InPaint, uPhotos, iSplash Colors, Snapheal, ACDSSee Photo Flash [2].

InPaint – програмний продукт, призначений для видалення з фотографії небажаних об’єктів. Програма достатньо швидко обробляє фотографію.

uPhotos – програмний продукт, з максимально спрощеним інтерфейсом, застосовує різні до завантаженого зображення. Стили встановлюються за допомогою відповідних елементів управління на панелі інструментів. Засіб є достатньо простим, швидким у обробці фото, проте має дуже обмежений набір можливостей та недостатньо привабливий інтерфейс користувача.

iSplash Colors – програмний продукт, що дозволяє редагувати фотографії. Його основною особливістю є те, що він дає змогу застосовувати різні стилі до фотографії та певним способом зафарбовувати окремі ділянки фотографії.

ACDSSee Photo Flash – програмний продукт, який призначений для зміни освітленості та контрастності зображення. Продукт є простим у використанні. Для зміни контрастності фотографії достатньо за допомогою повзунка встановити потрібне значення і одразу отримати результат.

Sharpeal – один з найвідоміших програмних продуктів обробки зображень для Mac OS X, дозволяє застосовувати різні стилі, ефекти до зображення, наприклад, чорно-білий, відтінки сірого, яскравість, чіткість, а також видаляти небажані об’єкти з певної ділянки.

Кожен із розглянутих програмних продуктів здебільшого не забезпечує вимоги пересічного користувача, вимагає спеціальних навиків для роботи з зображеннями, деякі з них мають складний інтерфейс користувача.

Постановка задачі

Постановка задачі: розробити максимально просте у використанні програмне забезпечення для опрацювання зображень в Mac OS X з нестандартними компонентами інтерфейсу користувача. Програма повинна вміти застосовувати такі ефекти до зображення, як: зміна чіткості, яскравості, контрастності, інтенсивності тіней, інтенсивності світла, інтенсивності «тепліх» відтінків, насиченості кольорів, зміна кута нахилу, інверсний поворот зображення, видалення об’єктів з певної області зображення.

Виклад розв’язання поставленої задачі

Компанія Apple пропонує безкоштовне програмне забезпечення з достатнім набором інструментів для створення якісних програмних додатків для Mac OS X. Водночас вона пропонує платформи, на базі яких можна створити програми різного характеру та призначення. Деякі платформи є вбудованими в саму операційну систему.

Carbon – це процедурний прикладний програмний інтерфейс (API), виданий компанією Apple Inc. для операційної системи Mac OS. Середовище Carbon засновано на галузевих стандартних мовах програмування C та C++. Ззовні майже неможливо знайти відмінності між застосунками Carbon та Cocoa.

Cocoa [4] – рідний об’єктно-орієнтований прикладний програмний інтерфейс (API) для операційної системи Mac OS X виробництва компанії Apple. Це один із п’яти основних API, доступних для Mac OS X (Cocoa, Carbon, Toolbox (для роботи старих додатків Mac OS 9), POSIX і Java).

Cocoa складається в основному з двох бібліотек об’єктів Objective-C, фреймворків :

- Foundation Kit, часто просто Foundation, уперше з’явився в OpenStep. В Mac OS X він заснований на основі Core Foundation. Foundation являє собою об’єктно-орієнтовану бібліотеку загального призначення, яка забезпечує роботу з рядками та значеннями, контейнери та ітерацію по них, розподілені обчислення, цикли опрацювання повідомлень та інші функції, не прив’язані безпосередньо до графічного інтерфейсу.
- Application Kit або AppKit походить по напряму від NeXTSTEP Application Kit. Він містить код, за допомогою якого програми можуть створювати графічний інтерфейс та взаємодіяти з ним. AppKit побудований на основі Foundation.

Ключовий елемент архітектури Cocoa – це модель переглядів (views). Зовнішньо вона організована як звичайний фреймворк, але реалізована з використанням PDF для усіх операцій малювання, схожого з PostScript. Крім того, це автоматично надає можливість виведення будь-якого перегляду на друк. Оскільки Cocoa обробляє обрізку, прокрутку, масштабування та інші типові задачі відображення графіки, програміст звільняється від необхідності реалізовувати базову інфраструктуру і може зосередитися на унікальних аспектах розроблюваної програми.

Програми, що використовують Cocoa, зазвичай розробляють за допомогою середовища розробки Apple Xcode (у минулому називався Project Builder) і Interface Builder з використанням мови Objective-C.

Середовище розробки Cocoa також доступне і під час розроблення іншими мовами, таких як Ruby, Python і Perl за допомогою сполучних бібліотек (RubyCocoa, PyObjC і CamelBones відповідно). Також можна писати Cocoa-програми на Objective-C у звичайному текстовому редакторі і вручну компілювати їх за допомогою GCC або make-сценаріїв для GNUstep. Cocoa-програми зазвичай мають характерний зовнішній вигляд, оскільки це середовище багато в чому спрощує підтримку принципів «дружнього інтерфейсу» Apple (Apple Human Interface Guidelines).

Середовище Cocoa суворо дотримується архітектурного шаблону Модель – Відображення – Контролер (Model – View – Controller, MVC). В Mac OS X компанія Apple представила сімейство класів MVC, що забезпечує стандартну функціональність поведінки – NSController. Ці класи вважаються частиною системи Cocoa Bindings, яка широко використовує такі протоколи, як Key-Value Coding (KVC) та Key-Value Observing (KVO). KVC дозволяє звертатися до елемента даних або властивості об'єкта, а також змінювати його під час виконання програми за ім'ям – ім'я властивості виступає ключем до його значення. KVC приводить до надзвичайної гнучкості дизайну – тип об'єкта знати необов'язково, але будь-яка його властивість може бути отримана за допомогою KVC. Крім того, завдяки технології KVO забезпечується автоматична синхронізація властивостей об'єктів, пов'язаних між собою [5].

За рахунок використання Key-Value Observing можна легко реалізувати відслідковування змін у полях моделі програмного забезпечення, а використання Key-Value Coding надає універсальний спосіб доступу до значень. Отже, можна просто пов'язати поля моделі з візуальними компонентами інтерфейсу користувача так, що зміни в одному місці будуть автоматично передаватись у всі решта.

Отже, вибір програмної технології Cocoa для розв'язання поставленої задачі є обґрунтованим та доцільним.

Для того, щоб програма виконувала своє основне призначення, у ній було реалізовано низку алгоритмів для опрацювання зображень.

Зокрема реалізовано базові методи регулювання кольірних характеристик зображення, таких як: яскравість, контрастність, насиченість, інтенсивність тіней. Базуючись на HSV моделі, пропонуємо встановлювати:

- значення насиченості – це інтенсивність кольору, що визначається як відсоток кольору вмісту певного тону відносно вмісту сірого і може змінюватися від 0 до 100 % (нульова насиченість відповідає абсолютно сірому кольору);
- значення яскравості – характеристики, що визначає, наскільки світлим чи темним може бути певний колір; цей параметр може змінюватися в діапазоні від 0 до 100 %;
- значення контрастності – максимальне відношення яскравості або щільності найсвітлішої і найтемнішої точки зображення.

Для усунення ефекту червоних очей реалізовано такий алгоритм [6]:

- розпізнавання шкіри об'єктів на ділянки зображення алгоритмом швидкої сегментації зображення;
- опрацювання ділянок зображення, на яких зображена шкіра, методами математичної морфології;
- перетворення зображення в напівтонове, виділяючи тим самим ділянки червоних очей;
- виявлення і сегментація частин зображення, на яких зображена шкіра, та присутній ефект червоних очей;
- заміна червоного кольору очей для отримання природного вигляду обличчя та очей.

Також реалізовано ефект розмиття та підвищення чіткості зображення. Розмивання зображення реалізовано методом Гаусса [7]. Цей ефект широко використовується для зменшення шуму в зображенні та зниження деталізації. Розмивання Гаусса – це тип фільтра розмивання зображення, що використовує функцію Гаусса для розрахунку трансформації кожного пікселя у зображенні. Коли метод застосовується у двох вимірах, то отримується поверхня, контури якої є концентричними колами розподілу Гаусса з центральної точки. Значення з цього розподілу використовують для створення матриці згортки. Для кожного нового значення пікселя визначається середнє зважене в околі пікселя. Значення поточного оригінального пікселя має більшу вагу (найвище значення розподілу Гаусса), а сусідні пікселі отримують все меншу вагу залежно від того наскільки далеко вони перебувають від поточного оригінального пікселя. Це надає ефект розмитості, яка зберігає межі та краї краще ніж інші, аналогічні фільтри розмиття [7].

Програмне забезпечення спроектовано та розроблено так, щоб графічний інтерфейс користувача не перешкоджав роботі основних методів опрацювання зображення у разі його модифікацій. На рис. 1 наведено діаграму компонент розробленого програмного забезпечення.

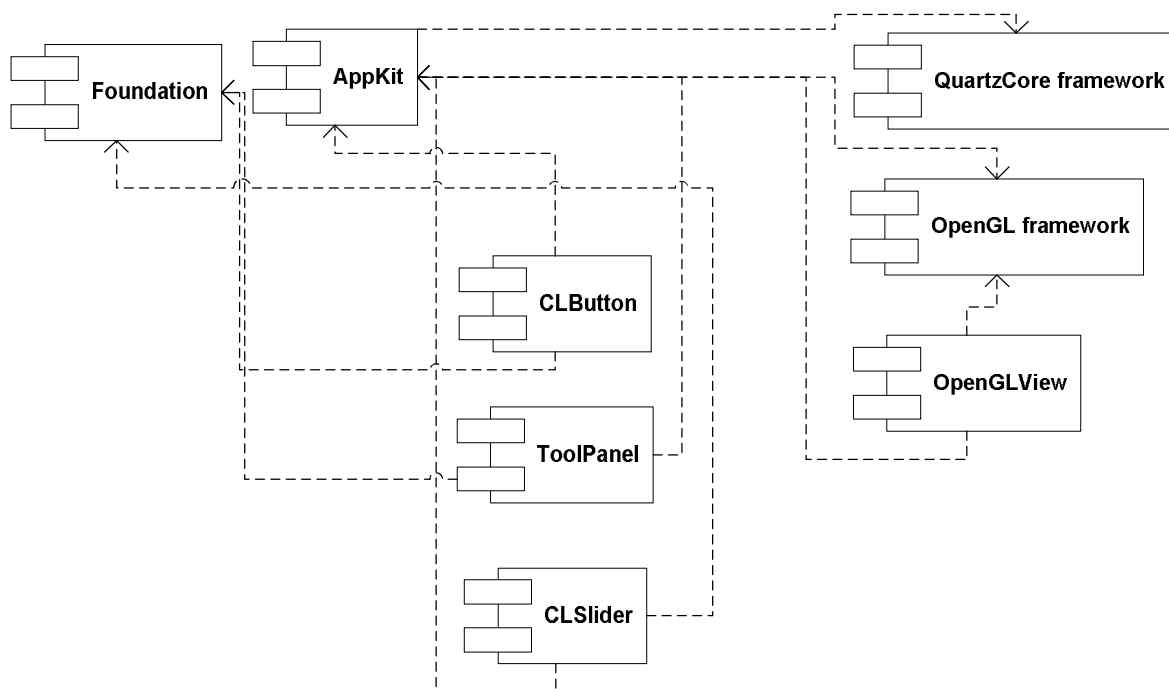


Рис. 1. Діаграма компонент програмного забезпечення

У графічному інтерфейсі користувача передбачено нестандартні візуальні компоненти, тому для кожної категорії об'єктів створено свій клас. За допомогою Interface Builder сконструйовано головне вікно програми, три елементи відображення – панелі з інструментами для редагування, ретушування зображення, інформаційної панелі; елемент відображення типу OpenGLView; панель з базовими елементами (повноекранний режим, повернути/повторити дію, вказівник).

Для реалізації розсувних панелей інструментів використано три види анімацій: анімація зміни розміру панелі, анімація руху панелі, анімація заміни однієї панелі відображення іншою. На рис. 2 наведено діаграму класів для реалізації інтерфейсу користувача.

Функціональна логіка графічного інтерфейсу програми зосереджена в трьох окремих класах-контролерах. Перший з них відповідає за управління в графічній ділянці редагування зображення. Другий клас-контролер відповідає за панель ретушування зображення. Третій клас-контролер відповідає за управління головним вікном програми. На ньому розміщується навігаційна панель управління програми з елементами управління, які відповідають за завантаження зображень, збереження зображень, зміну розмірів зображення в робочій ділянці, повернення/повтор дії, увімкнення/вимкнення повноекранного режиму.

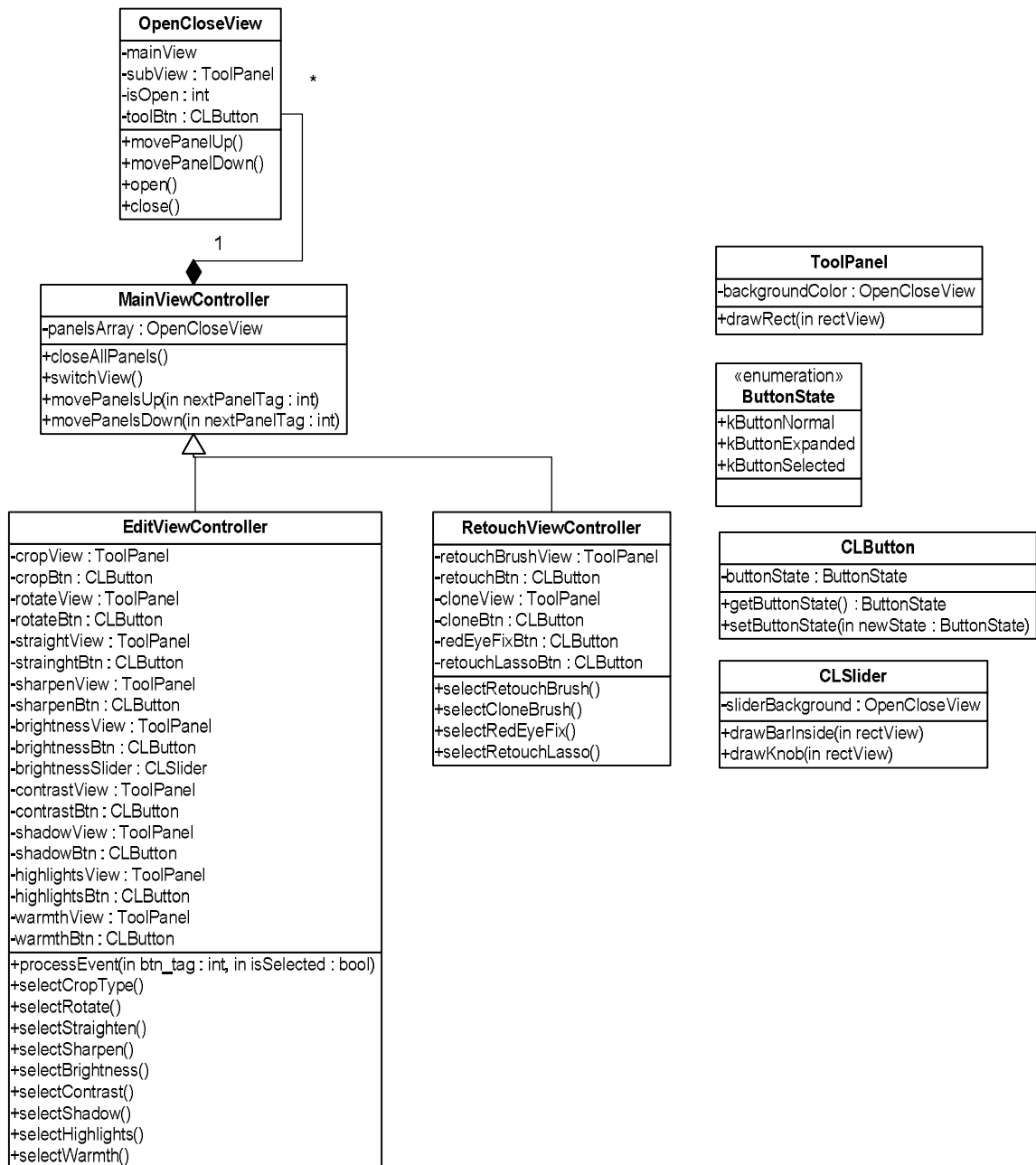


Рис. 2. Діаграма класів інтерфейсу користувача

Основою робочої ділянки зображення є панель з елементом OpenGLView. Цей клас дозволяє здійснювати маніпуляції над окремими пікселями зображення, на основі яких базуються алгоритми основних функцій програми.

У програмному продукті реалізовано логіку елемента управління TabView, який забезпечує перемикання між різними сторінками-табами навігаційної панелі, оскільки базовий елемент NSTabView не дозволяє реалізувати можливості, зокрема, встановлення зображень на кнопки-таби. Саме тому було реалізовано три окремі панелі, що завантажуються за викликом конкретної таб-кнопки, які, своєю чергою, реалізовані як звичайні кнопки.

Кожна окрема функція навігаційної панелі візуально складається з кнопки та панелі з піделементами управління. У програмі реалізована логіка відкриття, закриття панелі, зсув інших панелей догори/донизу відповідно до операцій над панеллю.

В інтерфейсі користувача надано особливого вигляду елементам прокрутки, кнопок, панелей відображення. Також реалізована зміна курсора відповідно до вибраного інструмента в конкретний момент часу.

Інтерфейс користувача розділений на логічні ділянки:

- панель з кнопками : відкрити, зберегти зображення;
- панель, що відображає назву, розміри та масштаб зображення, а також кнопки скасувати/повернути дію, вказівник, увімкнути/вимкнути повноекранний режим;
- ділянка роботи із зображення;
- ділянка інструментів для обробки зображення, представлена трьома вкладками: інструменти для ретушування зображення, редагування та інформація про зображення.

На рис. 3 зображено приклад роботи програми на вкладці редагування зображення.

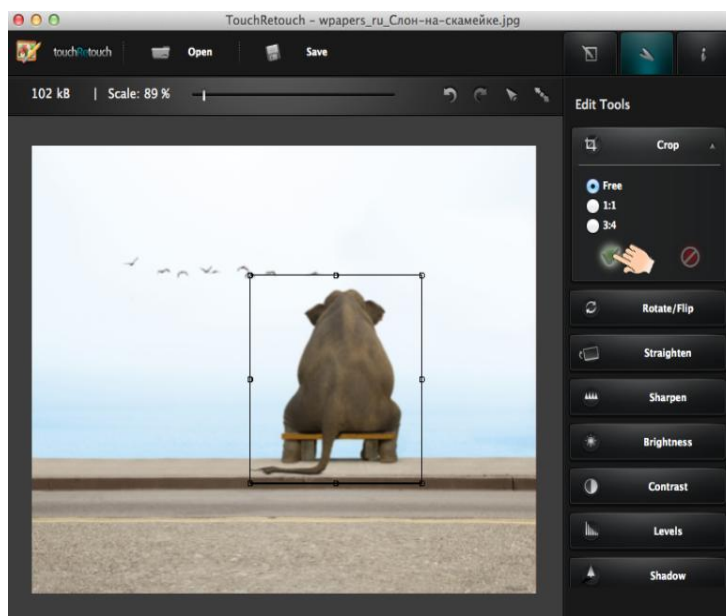


Рис. 3. Інтерфейс користувача розробленої програми

Кожна з вкладок містить різну кількість інструментів, які грамотно впорядковані і дають можливість користувачу зручно вибирати інструмент. Інтерфейс спроектований так, щоб усі решта інструментів не відволікали і не заважали користувачу, але й враховано те, що користувач може мати доступ до решти вкладок за потребою. Для цього створено панель, яка містить елементи управління конкретним ефектом. Така панель плавно відкривається і закривається за потребою користувача. На кожній з двох вкладок (ретушування, редагування) розміщено необхідну кількість таких панелей залежно від потреби.

Панель представляється класом з набором властивостей елементів управління та панеллю відображення, яка має нестандартний вигляд, тому в програмній реалізації для неї створено окремий клас. Для кожної з двох вкладок передбачено масив таких елементів, якими керує контролер вікна програми.

Одночасно користувач може працювати лише з однією вкладкою, тому передбачено автоматичне закривання відкритої панелі під час відкривання іншої. На діаграмі послідовності (рис. 4) виконання продемонстровано один з випадків – відкривання панелі, у той час як вже є відкрита інша.

В інтерфейсі також передбачено зміну вигляду курсору, оскільки користувачу повинно бути зрозуміло, яку операцію він тепер здійснює. Вигляд курсора може залежати не лише від вибраного інструмента, але від таких елементів управління, як повзунок (наприклад, зміна радіуса кисті). Для встановлення значення контрастності, чіткості, інтенсивності тіней та інших ефектів використовується компонент «повзунок», який дозволяє користувачу з легкістю встановити потрібне значення того чи іншого параметра.

Запропоновані алгоритмічно-програмні засоби перевірено на ефективність через тестування зручності використання програмного забезпечення набором з 14 окремих тестових випадків.

Акцент був зроблений на швидкості та ефективності навігації користувача по головному меню програми: панелі, кнопки, та інші елементи управління. У результаті виконання всіх тестових випадків було отримано очікувані результати.

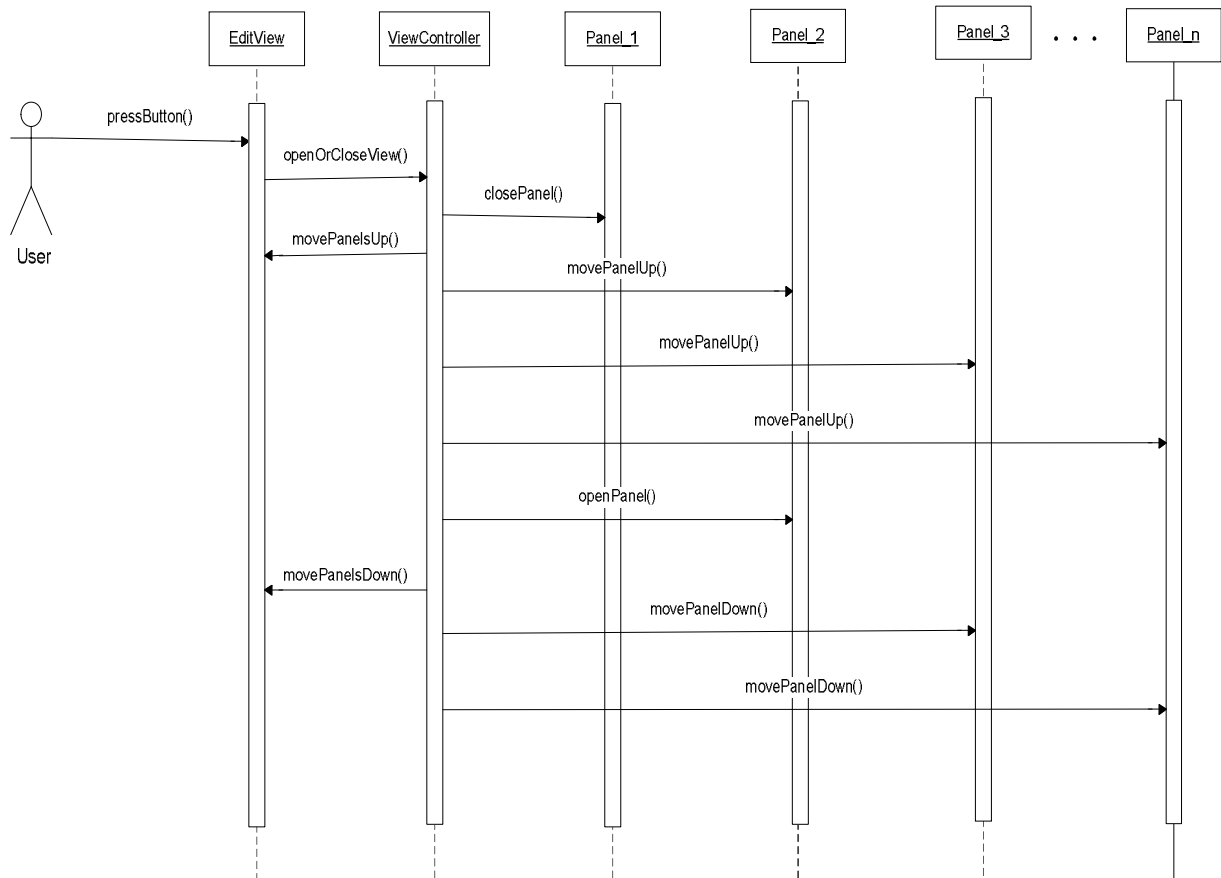


Рис. 4. Діаграма послідовності роботи користувача з нестандартними панелями

Висновки

Запропоновані алгоритмічні та програмні засоби опрацювання зображень в операційній системі Mac OS X забезпечують виконання таких операцій: видалення об'єкта; клонування об'єкта; зміну яскравості, контрастності, чіткості зображення; зміну інтенсивності тіней та інтенсивності теплих відтінків; усунення ефекту червоних очей. Результати тестування підтвердили ефективність розроблених засобів опрацювання зображень. Розроблене програмне забезпечення орієнтоване на пересічного користувача, має привабливий графічний інтерфейс.

1. Статистика операційних систем [Електронний ресурс] – Режим доступу: http://uk.wikipedia.org/wiki/Статистика_популярності_операційних_систем.
2. Офіційний сайт компанії Apple Mac App Store [Електронний ресурс]. – Режим доступу: <http://www.apple.com/mac/app-store>.
3. Kernel Architecture Overview [Електронний ресурс] – Режим доступу: <https://developer.apple.com/library/mac/#documentation/Darwin/Conceptual/KernelProgramming/Architecture/Architecture.html>.
4. Aaron Hillegass Cocoa Programming for Mac OS X / Aaron Hillegass, Adam Preble – 4th ed, 2011. – 400 p.
5. Cocoa Design Patterns / Erik M. Buck, Donald A. Yacktman – 1th end, 2009. – 456 pages.
6. Towards automatic redeye effect removal [Електронний ресурс] – Режим доступу : www.comm.toronto.edu/~kostas/Publications2008/pub.
7. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.

УМОВИ, СТРАТЕГІЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

© Бойко Н., 2012

Розкрито сучасне розуміння феномена інформаційного суспільства, подані показники, тенденції й прогнози впровадження інформаційних технологій у світі та його вплив на глобальний розвиток. У цьому контексті розглянуто соціокультурні зрушення в Україні, безпосередньо пов'язані зі впровадженням і використанням інформаційних технологій, зокрема: поширення нових форм масової комунікації, трансформація та розвиток систем «цифрової» демократії та електронного урядування, сучасних технологій. Розкрито стан проблеми розвитку інформаційних технологій. Запропоновані відповідні прогнози та рекомендації.

Ключові слова: інформаційні технології, сховища даних, WAP-технології, IP-телефонія, системи штучного інтелекту, комунікації, Інтернет, інформаційні системи, інформаційно-комунікаційні технології.

The article opened modern understanding of the Information Society, presented indicators, trends and forecasts implementation of information technology in the world and its impact on global development. In this context, consider the sociocultural developments in Ukraine is directly related to the implementation and use of information technologies, in particular the spread of new forms of mass communication, transformation and development systems "digital" democracy and e-governance and modern technology. Solved problem state of information technology. Appropriate forecasts and recommendations.

Key words: information technology, data storage, WAR technology, IP telephony, artificial intelligence, communications, Internet, information systems, information and communication technology.

Постановка проблеми

Внаслідок глобальної інформатизації суспільства активізуються нові геополітичні процеси, такі як: глобалізація економіки, що проявляється у створенні транснаціональних корпорацій, міжнародному розподілі праці та ринків збуту продукції; глобалізація науки, що активізує створення розподілених міжнародних творчих колективів учених, які працюють над спільними науковими проектами, а також процес інтенсифікації міжнародного обміну науковою інформацією, проведення міжнародних телеконференцій; глобалізація освіти, що активізує процес розвитку систем дистанційного навчання; глобалізація культури, яка проявляється у створенні електронних бібліотек, картинних галерей та інших творів мистецтва і літератури [1–3, 11].

Усе це активізує розвиток таких інформаційних технологій, як робота із сховищами даних, WAP-технології, цифрові фото та носії інформації, IP-телефонія, створення кишенькових комп'ютерів, нові принципи побудови дисплеїв, робота над створенням систем штучного інтелекту, виникнення дата-центрів, дистанційна освіта [10, 13].

Тому закономірним є те, що жодна галузь людської діяльності не зазнала такого розвитку, як інформаційні технології. Саме вони були покликані збільшити ефективність та зручність використання різноманітних видів інформації.

Аналіз останніх досліджень і публікацій

Сьогодні відомо безліч публікацій на цю тематику [2, 5, 11, 13], але зазвичай розглядаються окремо архітектура та технології. Проте, загалом, система функціонує і потрібно розглядати на різних рівнях її функціонування та взаємодію. Така увага дозволить визначитись з вимогами як до програмної, так апаратної частини інформаційної системи.

Питання щодо підходу до автоматизації управління підприємством розглядали у наукових публікаціях вітчизняних і закордонних учених та спеціалістів з інформаційних технологій: Ф.Ф. Бутинець [1], П.В.Іванюта [4], О.В. Кліменко [7], В.Д. Шквір [13] та ін.

Але, не зважаючи на таку різноманітність вивчення і розроблення підходів щодо автоматизації управління підприємством, недостатня кількість теоретиків та практиків розглядають питання сутності управлінських інформаційних систем та уніфікації їх класифікаційних ознак.

Формулювання цілей статті

Основним завданням розвитку інформаційного суспільства в Україні є сприяння кожній людині на засадах широкого використання сучасних інформаційно-комунікативних технологій (ІКТ) можливостей створювати інформацію і знання, користуватися та обмінюватися ними, виробляти товари та надавати послуги, повною мірою реалізуючи свій потенціал, підвищуючи якість свого життя. Тому розвиток інформаційного суспільства в Україні та впровадження новітніх ІКТ в усі сфери суспільного життя і визначає актуальність цієї статті.

Виклад основного матеріалу

У сучасному світі інформаційні технології (ІТ) дедалі активніше застосовуються у різних сферах діяльності, видозмінюючи їх і надаючи соціальному та людському розвитку нових якостей, сенсів та вимірів. Інформаційна складова має також істотне значення у формуванні конкурентного потенціалу держав, розвитку міжнародних відносин [2, 5].

Безпрецедентна за масштабами та якісними характеристиками глобальна інфраструктура інформаційних та інформаційно-комунікативних технологій перетворилася на технологічне підґрунтя дедалі ґрунтовніших трансформацій у повсякденному житті [6, 7].

Окрім небачених раніше можливостей для підвищення матеріального добробуту людства, сучасні ІТ спричинили появу та інтенсивне поширення принципово нових моделей соціальної інтеграції, комунікації, суспільно-політичної активності, способу життя, освіти тощо.

Особливо резонансні зміни пов'язані з "новими медіа", які функціонують у Web 2.0, що приводить до глибоких змін у структурах інформаційного споживання населення й до появи багатомільйонних глобальних "мережових спільнот" [3, 5].

У зв'язку із зазначеним у науково-експертному середовищі дедалі частіше дискутується проблема кристалізації специфічних "цифрових" (не локалізованих територіально й соціально) індивідуальних і групових ідентичностей, а також низької прогнозованості, "біфуркаційності" напрямів подальшого соціокультурного (а можливо, і суспільно-політичного) розвитку людства [11, 13].

Величезний потенціал – як позитивний, так і негативний – соціокультурної складової процесів "інформаційної технологізації" вже давно усвідомлений у розвинених країнах, причому не тільки на науково-експертному, а й на політичному рівнях. "Інформаційне суспільство" у цих країнах "першого світу" сприймається й розцінюється передусім як гуманітарна категорія, що описує якісні суспільні трансформації, зміщення акцентів з виробничої до невиробничої сфер, зміну характеру інформаційних потоків, групових та індивідуальних ідентичностей [11].

Відповідно, турбуючись про належний розвиток ІТ, інформаційно розвинені країни у своїй державній політиці впродовж тривалого часу (від початку 90-х років ХХ ст.) приділяють основну увагу доктринальним підходам, які дозволили б відслідковувати і, за можливості, контролювати суспільні зміни, спричинені ІТ.

Основним завданням розвитку інформаційного суспільства в Україні є сприяння кожній людині на засадах широкого використання сучасних ІКТ можливостей створювати інформацію і знання, користуватися та обмінюватися ними, виробляти товари та надавати послуги, повною

мірою реалізуючи свій потенціал, підвищуючи якість свого життя. Розвиток інформаційного суспільства в Україні та впровадження новітніх ІКТ в усі сфери суспільного життя і в діяльність органів державної влади та органів місцевого самоврядування визначається одним з пріоритетних напрямів державної політики [11].

Національна політика розвитку інформаційного суспільства в Україні ґрунтується на засадах: пріоритетності науково-технічного та інноваційного розвитку держави; формування необхідних для цього законодавчих і сприятливих економічних умов; всебічного розвитку загальнодоступної інформаційної інфраструктури, інформаційних ресурсів та забезпечення повсюдного доступу до телекомунікаційних послуг та ІКТ; сприяння збільшенню різноманітності та кількості електронних послуг, забезпеченню створення загальнодоступних електронних інформаційних ресурсів; поліпшення кадрового потенціалу; посилення мотивації щодо використання ІКТ; широкого впровадження ІКТ в науку, освіту, культуру, охорону здоров'я, охорону навколишнього середовища; забезпечення інформаційної безпеки [2, 10].

Сучасний світ характеризується стрімким прогресом у сфері поширення та розвитку інформаційно-комунікаційних технологій. Нові засоби телекомунікації, зокрема Інтернет як потужний глобальний інформаційний ресурс, приваблює широкі верстви населення незалежно від віку, освіти та соціального статусу.

Визначні характеристики інформаційного суспільства як моделі, є по суті, ідентичними характеристикам постіндустріального суспільства. Головним чинником суспільних змін стає виробництво та використання інформації; теоретичне знання як вища цінність і основний товар стає чинником формування нової соціальної структури суспільства, а також нових моделей керування. Якщо в індустріальну епоху в умовах капіталістичної економіки вирішальне значення мали власність і капітал, то в інформаційному суспільстві відбувається перехід до обслуговуючої економіки, коли домінує значення починає мати сфера послуг, причому послуг передусім, інформаційних [7, 9].

Процес визначається вибраною людиною стратегією і реалізовується за допомогою сукупності різноманітних засобів та методів. Оскільки інформація являє собою один із найцінніших ресурсів суспільства, вона є не менш важливою, ніж традиційні матеріальні види ресурсів. Процес переробки інформації за аналогією з процесами переробки матеріальних ресурсів можна сприймати як технологію. Тоді справедливим буде таке означення, як: інформаційна технологія являє собою комплекс збору, передачі, обробки, збереження і доведення до користувача інформації, що реалізована сучасними засобами. Ці принципово нові засоби і методи обробки даних об'єднуються в цілісні технологічні системи і забезпечують практично всі функції ІТ [7–9].

Оскільки на вході та виході ІТ є не матерія, і не енергія, а інформація, то: інформаційна технологія – це сукупність процесів, що використовує засоби та методи накопичення, обробки і передачі первинної інформації для отримання інформації нової якості про стан об'єкта, процесу або явища [4, 8].

Ця інформація нової якості називається інформаційним продуктом. Схематично процес перетворення інформації в інформаційний, а пізніше і в програмний продукт, можна проілюструвати за допомогою рис. 1. Під загрозами будемо розуміти сукупність факторів, які створюють небезпеку для цінної інформації, а саме: можливість несанкціонованого доступу і/або розповсюдження [6, 8].

Кожна інформаційна технологія орієнтована на опрацювання інформації певних видів, наприклад, статистичної (електронні таблиці, СУБД), текстової (текстові редактори), графіки (графічні редактори). Набори пакетів прикладних програм для математичних розрахунків і моделювання, експертні системи і бази знань використовуються в інформаційних системах для розв'язання формалізованих і неформалізованих задач.

Графічне і табличне подання даних часто застосовується як зручний інструмент економічного аналізу під час вивчення стану ринку (зокрема, за допомогою так званих графічних інформаційних систем), а також під час планування й прийняття рішень [7–11].

Системи мультимедіа забезпечують роботу з багатьма інформаційними середовищами: нерухомим зображенням і рухомим відео, анімованою комп'ютерною графікою, текстом і звуком.

Гіпертекстові технології створюють нові, якісно відмінні від традиційних, можливості засвоєння інформації. Вони передбачають переміщення від одних об'єктів інформації до інших з урахуванням їх змістової та семантичної взаємопов'язаності [7–11].

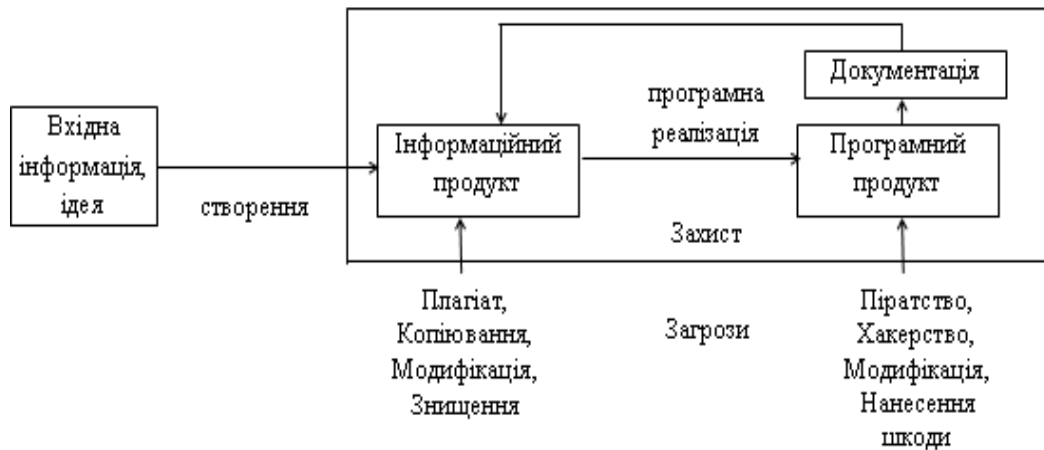


Рис. 1. Місце інформаційного та програмного продукту в системі інформаційного кругообігу

Цифрові технології дедалі ширше завойовують і сучасну видавничу справу. Кількість електронних видань поступово наближається до кількості друкованих. Спостерігається стійка тенденція до зростання кількості електронних видань, які розповсюджуються через мережу Інтернет. На основі сіткових технологій реалізовано видавничу діяльність під назвою Print-on-Demand (друкування за вимогою), де поєднується традиційне та електронне видавництво [7–11].

Для інформаційних технологій є цілком природним те, що вони застарівають і замінюються новими. Так, наприклад, на зміну технології пакетного опрацювання програм на великий ЕОМ в обчислювальному центрі прийшла технологія роботи на персональному комп'ютері на робочому місці користувача. Телеграф передав усі свої функції телефону. Телефон поступово витісняється службою експрес-доставки. Телекс передав більшість своїх функцій факсу й електронній пошті.

Тому виникнення нових гіпертекстових технологій стало можливим завдяки здешевленню вартості персональних комп'ютерів (ПК) і широкому їх охопленню глобальними комп'ютерними мережами. Інформаційний обмін почав будуватись на основі розподілених баз даних. Програмне забезпечення передбачило вже не лише індивідуальні засоби, а й системи колективного користування, підтримку мультимедіа і тривимірну графіку [3, 5].

Основними завданнями сучасних ІТ є:

- досягнення універсальності методів комунікацій;
- підтримка систем мультимедіа;
- максимальне спрощення засобів спілкування в системі "людина – ПК".

Під час впровадження нової інформаційної технології в організації необхідно оцінити ризик відставання від конкурентів у результаті її неминучого старіння, тому що інформаційні продукти, як ніякі інші види матеріальних товарів, мають надзвичайно високу здатність замінюватись новими видами або версіями. Періоди змінюваності коливаються від декількох місяців до одного року. Якщо під час впровадження нової інформаційної технології цьому фактору не приділяти належної уваги, цілком можливо, що до моменту завершення переходу фірми на нову інформаційну технологію вона вже застаріє і прийдеться вживати заходів щодо її модернізації. Такі невдачі з впровадженням інформаційних технологій звичайно пов'язані з недосконалістю технічних засобів, у той час як основною причиною невдач є відсутність або слабка пропрацьованість методології використання інформаційної технології [7].

Необхідно розуміти, що освоєння інформаційної технології і подальше її використання повинні бути зведені до того, щоб було потрібно спочатку добре оволодіти набором елементарних операцій, кількість яких обмежена. З цієї обмеженої кількості елементарних операцій у різних комбінаціях складається дія, а з дій, також у різних комбінаціях, складаються операції, що визначають той або інший технологічний етап. Сукупність технологічних етапів утворює технологічний процес (технологію). Він може починатися з будь-якого рівня і не передбачати,

наприклад, етапи або операції, а складатися тільки з дій. Для реалізації етапів технологічного процесу можна використовувати різні програмні середовища.

Інформаційна технологія, як і будь-яка інша, повинна відповідати таким вимогам [4, 6]:

- забезпечувати високий рівень розчленування всього процесу опрацювання інформації на етапи (фази), операції, дії;
- об'єднувати весь набір елементів, необхідних для досягнення поставленої мети;
- мати регулярний характер. Етапи, дії, операції технологічного процесу можуть бути стандартизовані й уніфіковані, що дозволить ефективніше здійснювати цілеспрямоване керування інформаційними процесами.

Технічними засобами виробництва інформації являється апаратне, програмне і математичне забезпечення цього процесу. За їхньою допомогою переробляють первинну інформацію в інформацію нової якості. Виділимо окремо з цих засобів програмні продукти і назвемо їх інструментарієм, а для більшої чіткості можна його конкретизувати, назвавши програмним інструментарієм інформаційної технології [10].

Інструментарій інформаційної технології – один або декілька взаємозалежних програмних продуктів для певного типу комп'ютера, технологія роботи в яких дозволяє досягти поставленої користувачем мети.

Інструментарієм можна використовувати такі поширені види програмних продуктів для персонального комп'ютера, як текстовий процесор (редактор), настільні видавничі системи, електронні таблиці, системи керування базами даних, електронні записні книжки, електронні календарі, інформаційні системи функціонального призначення (фінансові, бухгалтерські, для маркетингу тощо), експертні системи тощо [9, 10].

У загальній концепції інформаційної технології є діалектичний взаємозв'язок – розвиток інформаційної технології вимагає вдосконалення засобів, які її забезпечують і, навпаки, поява певних засобів вимагає створення нових прийомів, методів і способів.

Однією з найважливіших вимог до методології проектування інформаційної технології є забезпечення динамічності її структури і функцій. Тому значний інтерес становить визначення інваріантів технологій, тобто опорних структур даних і процесів, які можуть бути доволі стабільними властивостями і характеристиками гнучкої інформаційної технології. Очевидно, що це вимагає глибокого вивчення можливостей, як сучасних і перспективних засобів обчислювальної техніки і програмного забезпечення, так і технологічних процесів в певній предметній області та їх технічного оснащення [1, 4, 7].

Основною метою інформаційної технології вважається повне і своєчасне задоволення інформаційних потреб користувачів, де користувачі – це не окремі працівники, які обслуговують певну інформаційну технологію, а вся система, загалом. Зайняті в ній люди розглядаються лише як учасники колективних зусиль, підпорядкованих загальній меті функціонування системи. Окремих працівників можуть задовольнити фрагментарні відомості, що стосуються сфери їх компетентності і лише система ІТ, загалом, має потребу в єдиній інформаційній моделі об'єкта, в якій цілісність відображає поєднання з досягнутою детальністю. Лише така модель може стати інструментом перевірки узгодженості численних рішень, що приймаються в межах ІТ, а отже, організації інформаційної взаємодії різних підрозділів, колективів, осіб [2, 3].

Для конструктивного аналізу цілей під час побудови ІТ необхідно, щоб загальні цілі створення ІТ узгоджувались з цілями процесів керування в певній предметній області. Наприклад, для ІТ в промисловості визначені такі цілі:

- усунення наявних проблемних моментів;
- інтеграція управління і виробництва;
- стандартизація і уніфікація засобів автоматизації.

Практика використання інформаційних технологій для моделювання та автоматизації підтримки прийняття рішень в управлінні соціально-економічними процесами тісно пов'язана із постійним розв'язанням задач аналізу значних обсягів інформації. Особливої актуальності аналіз

інформації набуває за потреби прийняття рішень для управління різними процесами. Прикладом може слугувати аналіз значних обсягів економічної інформації [11].

Для його проведення можна використовувати методи, що ґрунтуються на основі інструментальних засобів та успішно застосовуються для складного аналізу статистичних даних шляхом прямого програмування інструментальними мовами. Спочатку будуються економіко-математичні моделі, а потім здійснюється їх комп'ютерна реалізація з використанням інструментальних засобів – мов програмування.

Інша група методів аналізу значних обсягів економічних даних ґрунтується на використанні пакетів прикладного програмного забезпечення. Так, наприклад, для систематизованого опрацювання значних обсягів табличних даних на ПК під управлінням операційних систем сімейства Microsoft Windows використовуються процесори електронних таблиць (ЕТ) Microsoft Excel, спеціалізовані програмні пакети StatSoft Statistica, StatGraphics Plus, SPSS SigmaPlot тощо. Зазначені програмні засоби мають потужні можливості опрацювання та аналізу даних, проте їх спільними недоліками є орієнтованість винятково на роботу під управлінням операційних систем сімейства MS Windows та порівняно висока вартість ліцензій на офіційне використання.

Під час створення будь-якої системи потрібно приділяти увагу вибору архітектури та технології, які використовуватимуть. Проблема вибору web-технологій для інформаційної системи орієнтованої на web-доступ (Web-Based) є актуальною. Адже оптимальний вибір забезпечить розширений термін життєвого циклу системи, просту підтримку та зручність у її використанні в короткостроковій та довгостроковій перспективі [12].

Розвиток інформаційних технологій актуалізував деякі проблеми, що постають перед узагальненою моделлю розподілених інформаційних систем. Зокрема, це формалізація відкритих систем та принципів створення автономних інформаційних систем. Отже, інформаційна технологія на практиці може бути успішно реалізована на основі наведених програмних засобів для автоматизації процесів аналізу інформації.

Висновки та перспективи подальших досліджень

Інформаційна технологія – сукупність процесів, що використовує засоби та методи накопичення, опрацювання і передачі первинної інформації для отримання інформаційного продукту (інформації нової якості про стан об'єкта, процесу або явища). Вона повинна давати можливість оцінити вплив рішень, які приймаються, на підвищення ефективності процесів, які вона обслуговує. Адже основною метою інформаційної технології вважається повне і своєчасне задоволення інформаційних потреб користувачів.

Практика використання інформаційних технологій для моделювання та автоматизації підтримки прийняття рішень в управлінні соціально-економічними процесами тісно пов'язана із постійним розв'язанням задач аналізу значних обсягів інформації. Застосування сучасних інформаційних технологій в управлінні є одним з головних інструментів ефективності бізнесу. Мета застосування управлінських інформаційних систем полягає в тому, щоб упорядкувати інформаційні потоки, надати ефективний доступ до даних менеджерів усіх рівнів для прийняття мотивованих управлінських рішень. Очевидно, що без своєчасної і достовірної інформації про стан справ на підприємстві неможливе управління ним, а, отже, неможливо і розвиток бізнесу.

Бутинець Ф.Ф. Інформаційні системи бухгалтерського обліку [Текст] : підручник / Ф.Ф. Бутинець, С.В. Івахненко, Т.В. Давидюк, Т.В. Шахрайчук; За ред. проф. Ф.Ф. Бутинця. – 2-ге вид., перероб. та доп. – Житомир: ПП Рута, 2002. – 544 с. 2. Вовчак І. Інформаційні системи та комп'ютерні технології в менеджменті [Текст] : навч. посіб. / І.С. Вовчак; Мін-во освіти і науки України, Тернопільський держ. технічний ун-т ім. І. Пулюя. – Тернопіль : Карт-бланш, 2001. – 354 с. 3. Єдинак В.С. Розвиток інформаційних технологій в Україні // Наукові доробки молоді – вирішенню проблем європейської інтеграції: збірник наукових статей. В 2 т. Т. 1. – Харків: Континент, 2008. – С. 289–290. 4. Іванюта П.В. Управлінські інформаційні системи в аналізі та аудиті [Текст]: навч. посіб. / П.В. Іванюта; Мін-во освіти і науки України, Держ. комітет

статистики України, Держ. академія статистики, обліку та аудиту, Полтавська філія. – К.: ЦУЛ, 2007. – 180 с. 5. Івахненко С.В. Інформаційні технології в організації бухгалтерського обліку та аудиту [Текст]: Навчальний посібник / Івахненко С.В. – К. : Знання-Прес, 2003. – 349 с. 6. Какорін М.О. Інформаційні технології як фактор інновацій у глобальній фінансовій системі.// Формування ринкових відносин в Україні: зб. наук. праць. – 2008. – № 5. – С. 106–109. 7. Кліменко О.В. Інформаційні системи і технології в обліку [Текст] : навч. посіб. / О.В. Кліменко. – К.: Центр учб. літ., 2008. – 320 с. 8. Минухін С.В. Формування інформаційного забезпечення системи управління бізнес-процесами підприємства / С.В. Минухін // Актуальні проблеми економіки. – 2006. – № 10. – С. 170–178. 9. Пацай Б.Д. Роль інформаційних технологій в управлінні фінансовими ресурсами підприємств.// Фінанси України. – 2008. – № 8. – С. 82–84. 10. Романченко О. Управлінський облік і система управлінської інформації / О. Романченко // Вісник Національного банку України. – 2008. – № 6. – С. 15–18. 11. Чигасова Н.М. Місце інформаційних технологій у розвитку інформаційного суспільства в Україні.// Формування ринкових відносин в Україні: Збірник наукових праць. – 2007. – № 9. – С. 110–113. 12. Шандра В.М. Застосування інформаційних технологій в забезпеченні технологічного оновлення економіки на інноваційній основі // Актуальні проблеми економіки. – 2007. – № 10. – С. 220–223. 13. Шквір В.Д. Інформаційні системи і технології в обліку [Текст] : навч. посіб. / В.Д. Шквір, А.Г. Завгородній, О.С. Височан. – К. : Знання, 2006. – 439 с.

УДК 628.321

В. Заяць

Національний університет “Львівська політехніка”,
кафедра загальної екології та екоінформаційних систем

ПІДХОДИ ДО ПОБУДОВИ ЕКОІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

© Заяць В., 2012

Розглянуто основні підходи до описування та створення екоінформаційних систем на основі дискретних моделей та інформаційно-комп'ютерних технологій.

Ключові слова: дискретні моделі, мережі екоданих, екоінформаційна система, інформаційна система, прикладне програмне забезпечення.

The paper describes the main approaches to the development of systems based on ecoinformation discrete models and computer technology.

Key words: discrete model, data network ecology, ecologies information system, information system, application software.

Вступ

Екологічна інформаційна система (ЕІС) – це система для управління екологічною інформацією, її аналізу та подання з метою ефективного прикладного застосування за мінімальних фінансових затрат. Екологічну інформацію можна подавати у вигляді цілого набору екологічних даних, які моделюють навколишнє середовище, за допомогою узагальнених структур даних. ЕІС, що об'єднують набори інструментальних засобів для роботи з екологічними даними.

Здатність ЕІС вести пошук у базах даних, приєднувати власні бази, здійснювати просторові запити, безперервно нагромаджувати та коригувати наявні просторові й часові дані, моделювати та

відтворювати реальні ситуації, заощадити час і кошти державних та комерційних структур, унеможливити виникнення кризових та аварійних ситуацій.

При розробленні нових реальних пристроїв, дослідженні невивчених або невідомих фізичних чи екологічних явищ або процесів, побудові систем ідентифікації та розпізнавання, що мають задані характеристики інформаційного сигналу або невідомі характеристики, які підлягають вивченню, доцільно попередньо провести комп'ютерне моделювання та аналіз, створивши адекватні математичні моделі об'єкта, що розробляється чи вивчається. Такий підхід вимагає істотно менших часових і технічних засобів порівняно з фізичним експериментом, особливо на попередній стадії розробки, за відсутності достовірної апріорної інформації про навколишнє середовище та поведінку об'єктів, що в ньому перебувають.

Останнім часом в нелінійній динаміці широкого застосування набувають дискретні моделі систем [1–4], для яких дискретність закладена в природі самого об'єкта досліджень, а не є наслідком дискретизації неперервної системи. Доцільність використання дискретних за своєю природою моделей пояснюється такими їх особливостями:

- простотою математичного опису порівняно з неперервними моделями;
- наявністю значно ширшого спектра динамічних режимів порівняно з відомими моделями;
- нескінченною вимірністю, що дозволяє моделювати кожен нову гармоніку процесу шляхом її введення у вектор змінних стану, тоді як для неперервних систем для розв'язання цієї задачі необхідно підвищувати розмірність системи;
- відсутністю необхідності визначення кроку дискретизації, оцінок похибок застосованих чисельних методів, дослідження областей їх стійкості та синхронізації;
- кращою адаптованістю до постановки комп'ютерного експерименту, порівняно з неперервними моделями.

Власне моделі, дискретні за своєю природою, є застосовні як до побудови пристроїв, що мають бажані режими, так і до розпізнавання та ідентифікації ситуацій у системах зі складною динамікою і поведінкою, якими є екологічні системи, що дає змогу підвищити ефективність їх роботи.

Цілі статті

Метою статті є формування основних вимог та підходів до побудови екоінформаційних систем для забезпечення якісного та автоматизованого процесу моніторингу навколишнього середовища та прийняття рішень щодо його захисту та розвитку. Однією із основних цілей статті є формування вимог до побудови екоінформаційних систем, встановлення доцільності використання сучасних технологій та методів формування даних у галузі природоохоронної діяльності. У роботі також визначено шляхи розвитку таких систем та напрями їх прикладного застосування.

Виклад основного матеріалу

У дев'яностих роках минулого століття практично у всіх розвинених країнах світу було створено національні екоінформаційні системи, які охоплювали існуючі системи моніторингу навколишнього середовища, а також системи збору і аналізу інформації про фізичне навантаження та стан здоров'я населення. У цей же період поява нових інформаційних технологій та розвиток мережі Internet зумовила об'єднання цих систем в єдину екоінформаційну систему, на серверах якої зберігають величезні обсяги інформації про стан навколишнього середовища, отримані за допомогою систем екологічного моніторингу.

Сучасна система моніторингу довкілля повинна містити такі технічні та інтелектуальні засоби опрацювання даних:

- розпізнавання та ідентифікації об'єкта спостереження;
- збору апріорних даних про об'єкт спостереження;
- формування інформаційної моделі об'єкта дослідження;
- планування експерименту (натурного, лабораторного, числового, вимірювального);
- оцінки відповідності об'єкта спостереження його інформаційній моделі;

- прогнозу і передбачення поведінки досліджуваного об'єкта спостереження;
- подання інформації у зручній для користування формі;
- експертної оцінки достовірності отриманих даних.

Основні цілі екологічного моніторингу полягають у забезпеченні системи управління природоохоронної діяльності та екологічної безпеки своєчасною і достовірною інформацією, що дає можливість оцінити стан навколишнього середовища, виявити причини та можливі наслідки змін його стану, а також визначити необхідні захисні дії [5]. За функціональним призначенням виділяють три види моніторингу навколишнього середовища: стандартний, кризовий, науковий.

Власне екоінформаційні системи охоплюють всі різновиди систем екологічного моніторингу і забезпечують систему управління і прийняття рішення повною і достовірною інформацією для екологічно безпечного розвитку всієї території, де поширюється їх дія.

Екоінформаційна система повинна забезпечувати розв'язання цілої низки задач:

- підготовка системної інформації про стан середовища, передбачення ймовірних наслідків суспільної діяльності, рекомендацій щодо вибору варіантів безпечного розвитку регіону та розроблення інструкцій для систем підтримки і прийняття управлінських рішень;
- моделювання процесів, що відбуваються в навколишньому середовищі, та передбачуваних результатів ухвалення управлінських рішень;
- підготовка електронних карт, що відображають стан навколишнього середовища території;
- опрацювання і накопичення у базах даних результатів моніторингу та встановлення параметрів довкілля, найчутливіших до зміни його стану;
- обґрунтування доцільної мережі спостережень для регіональної системи екологічного моніторингу;
- обмін інформацією про стан навколишнього середовища з іншими екоінформаційними системами;
- надання даних для контролю за дотриманням прийнятих законів, для розвитку екологічної освіти, для засобів масової інформації тощо [6].

Отже, екоінформаційні системи повинні бути зорієнтовані на комплексне використання результатів екологічного моніторингу, забезпечуючи перетворення первинних результатів спостережень у форму, придатну для управління та прийняття рішення, які сприяють усталеному розвитку як окремих регіонів, так і всієї планети. У міру переходу від первинних результатів екологічного моніторингу до знань про стан навколишнього середовища змінюються методи роботи з інформацією. В екоінформаційній системі можна виділити три рівні, орієнтовані на розв'язання різних задач екологічного моніторингу, які різняться методами роботи з екологічною інформацією. Верхній рівень становлять програмні модулі для підтримки прийняття рішень, середній – програмне забезпечення, що дає змогу здійснити системний аналіз інформації про стан навколишнього середовища, нижній – модулі опрацювання первинної екологічної інформації [7]. На нижньому рівні екоінформаційної системи для зберігання даних про стан навколишнього середовища використовуються різні системи управління базами даних (СУБД) типу Oracle чи Microsoft SQL Server, а для опрацювання результатів спостережень застосовують різноманітні програмні продукти – електронні таблиці, пакети прикладних програм типу MathCAD, Surfer та інші. Зазначимо, що із прогресивним розвитком технологій програмування типу Silverlight, Flash, AJAX та сучасних мов програмування C⁺⁺, C[#], Java, Delphi, Turbo-Prolog, Lisp можливо швидко отримувати доступ до екоданих та візуалізувати їх із будь-якої точки Землі через зв'язок із глобальною мережею Internet. Така різноманітність програмного забезпечення зумовлена величезною кількістю різнопланових задач опрацювання результатів спостережень за станом довкілля, одержаних за допомогою тих чи інших методів екологічного моніторингу.

На середньому рівні екологічної інформаційно-комп'ютерної системи для аналізу інформації про стан навколишнього середовища використовуються географічні інформаційні системи (ГІС). Ці системи, забезпечуючи введення, зберігання, передавання, модифікацію (видалення, додавання), опрацювання, аналіз і візуалізацію всіх видів географічно прив'язаної інформації, дають змогу

систематизувати процес отримання інформації для управління природними ресурсами та їх захисту від неприродного напрямку розвитку чи бездумного знищення

Для забезпечення підтримки прийняття рішень необхідний ще один етап роботи з інформацією, який дає змогу співвіднести одержані результати зі шкалою "істинне – хибне". Очевидно системи підтримки прийняття рішень у галузі екологічної безпеки доцільно створювати, спираючись на результати математичного та комп'ютерного моделювання. У межах таких математичних чи комп'ютерних моделей стає можливим зіставлення відомостей з різних джерел та прогнозування і передбачення наслідків того або іншого управлінського рішення. Сьогодні методи математичного моделювання докілька ще недостатньо розвинуті та апробовані, щоби їх результати можна широко використовувати для підтримки прийняття рішень у галузі екології та природоохоронної діяльності. Тому сьогодні накопичення знань, необхідних для підтримки прийняття рішень, ґрунтується на різних спрощених методиках оцінки дії на навколишнє середовище, таких, як методологія оцінки дії на довкілля, індикатори стійкого розвитку та інші [8].

Особливості розроблення комп'ютерної ЕІС (КЕІС). Однією з пріоритетних задач при створенні систем екологічного моніторингу є проблема забезпечення оперативності роботи системи моніторингу. Важливість цієї проблеми очевидна, оскільки запізніле надходження інформації про стан навколишнього середовища не дасть змоги організувати його оперативний захист. Основним способом розв'язання цієї задачі є створення систем екологічного моніторингу навколишнього середовища на основі комп'ютерних технологій опрацювання даних у реальному часі і застосування ефективних і швидкодіючих систем передавання даних.

Другою проблемою, що має місце при розробленні системи моніторингу, є його комплексність. Суть цієї проблеми зводиться до того, що більшість систем моніторингу, контролює не один, а декілька компонент навколишнього середовища. Отже, питання взаємозв'язку між розташуванням і регламентом роботи різних вимірювальних складових, а також опрацювання інформації, що поставляється ними стає актуальним. Проблема комплексності може бути вирішена на уявленні про взаємозв'язок компонентів навколишнього середовища і процесів, що у них відбуваються. Тому розташування і регламент роботи різних інформаційно-вимірювальних мереж слід скоординувати.

Третьою проблемою є проблема репрезентативності отриманих результатів. Суть проблеми полягає в оптимізації вибору розташування ланок інформаційно-вимірювальної мережі і інтервалів між вимірюваннями. Рациональний вибір просторового і часового кроку дозволяє вирішити цю проблему. Просторовий аспект вирішення проблем комплексності і репрезентативності можна запропонувати у вигляді ландшафтного принципу розміщення ланок інформаційно-вимірювальної мережі. Хоча такий підхід дає ефект у межах однієї ЕІС і не означає повної ідентичності розміщення вимірювальних ланок різних інформаційно-вимірювальних комплексів різних ЕІС. Питання про вибір часового кроку для отримання достовірних вхідних параметрів вирішується на основі наявних даних про часову мінливість параметрів, що вивчаються, а також обмежень на швидкість їх зміни і частоту.

Четверта проблема – це проблема адаптивності. Під час побудови системи моніторингу доцільно скористатися принципом адаптивної структури. Його суть полягає у тому, що система моніторингу змінює свою організацію (розташування ланок інформаційно-вимірювальної мережі, регламент їхньої роботи, процедури опрацювання інформації, системи підтримки прийняття рішень) на підставі аналізу одержаних системою даних.

П'ята проблема – інтелектуалізація ЕІС для забезпечення прийняття нею достовірного рішення про стан довкілля та вироблення коректних керуючих дій. Тобто у склад КЕІС входить підсистема прийняття рішень та керування станом навколишнього середовища, динамічні бази даних, що постійно оновлюються (бази знань), та набір моделей і правил виведення логічних висновків. По суті справи, ця підсистема утворює експертну систему, поведінка якої близька до поведінки людини-експерта з певної предметної галузі. Проте, оскільки йдеться про опрацювання надзвичайно великих обсягів інформації в стислі проміжки часу, то переваги експертної системи є очевидними, порівняно з людиною експертом. До них слід зарахувати

- відсутність суб'єктивності, упереджень та поспішності при прийнятті рішень та висновків;

- системний, логічно-послідовний підхід до виконання поставленої проблеми, що забезпечує після опрацювання всієї заданої (введеної у комп'ютерну систему) інформації вибрати єдино правильне (найкраще за заданим критерієм) рішення з усіх можливих альтернатив, незалежно від їх кількості;
- база даних (статична) і знань (динамічна) KEIC є надмірно великими і постійно зберігається у пам'яті комп'ютера. До того ж людина володіє обмеженою базою даних і є фахівцем у конкретній галузі знань, які при їх довгому не використанні забуваються і можуть бути втрачені назавжди;
- експертні системи не піддаються впливу зовнішніх чинників, які безпосередньо не пов'язані з розв'язуваною задачею;
- експертні системи не можуть замінити спеціаліста конкретної предметної області, а є лише інструментальним засобом для розв'язання конкретної задачі.

У експертній системі слід передбачити коментарі, зрозумілі фахівцю у цій галузі, про спосіб розв'язання поставленої задачі з метою перевірки достовірності отриманих результатів фахівцем з математичної статистики [8]. Розроблення експертних систем, призначених для опрацювання даних, стикається зі значними складнощами. Інтелектуалізація комп'ютерного опрацювання первинної інформації про навколишнє середовище ґрунтується, з одного боку, на ідеях і методах конкретної галузі знання, для якої створюється система опрацювання даних. З іншого, у комп'ютерній системі опрацювання даних використовуються різноманітні методи прикладної математики, теорії ймовірностей і математичної статистики, теорії розв'язання обернених задач, теорії інформації, комп'ютерних систем в мереж, алгебри логіки, штучного інтелекту та інше. Очевидно, при розробленні експертних систем опрацювання даних, з одного боку, доводиться враховувати особливості методик виконання вимірювань та методів проведення розрахунків, а з іншого, обмеження математичних алгоритмів обробки, що вимагає участі доволі великого колективу професіоналів – фахівців у цій галузі, а також: математиків, програмістів, фахівців з розроблення експертних систем. Все це зумовлює високу вартість розроблення. Тому за наявності величезної кількості систем загального призначення – пакетів для статистичного опрацювання даних, електронних таблиць, існує зовсім невелика кількість експертних систем, здатних на інтелектуальному рівні розв'язувати конкретну прикладну задачу чи реалізовувати інтелектуальну обробку даних.

Підходи до вибору та побудови моделей інтелектуальної обробки даних в ЕІС. Під час розроблення екоінформаційної системи важливим є вибір методики та способу зберігання та обробки даних. При створенні моделі даних, тобто способу цифрового опису просторових об'єктів, слід врахувати усі вхідні та вихідні параметри, їх структуру, методи збору, способи збереження, кодування, передавання, обробки та візуалізації даних. Вибір способу організації даних ЕІС має важливе значення, оскільки безпосередньо визначає функціональні можливості створеної ЕІС, а також придатність тих чи інших технологій отримання та запису даних. Від вибору моделі даних залежить як просторова точність представлення графічної частини інформації, так і можливість отримання якісного картографічного матеріалу і організація контролю карт за екологічною обстановкою. Від способу організації даних ЕІС великою мірою залежить також швидкодія системи. Зазвичай, всі дані розмежовані та утворюють свого роду рівні. Загалом усі рівні подання даних утворюють ієрархію, яку можна класифікувати та виділити як окремі частини ЕІС.

Модель даних [9] у нашому випадку є загальною концепцією організації даних у екоінформаційній системі, які являють собою створені та готові до відображення та перетворення на результат дані. До таких даних можна зарахувати як окремі об'єкти візуалізації, так і статистичні дані, що містять інформацію про стан навколишнього середовища. Рівень зв'язку даних є проміжним рівнем між сховищем і моделлю даних, відповідно і призначений для зв'язку. Зазвичай це матриці перетворення, векторні дані, посилання, списки та інші специфічні для кожної з ЕІС способи структурування даних. Структури даних – це найнижчий рівень, який максимально

деталізує модель даних, та уможливує нормалізацію даних. До цього рівня слід зарахувати структури файлів, баз даних та типи даних.

Елемент кожної моделі даних повинен містити ідентифікатори, атрибути, прив'язку до просторово-часових моделей даних, а також функції перетворення та обробки даних. Ідентифікатор – це один із основних параметрів, що бере участь у структуруванні даних. Його призначення – це формування певних ознак, інформації про об'єкт, яка може динамічно змінюватись, незважаючи на статичність самого об'єкта. Кожний об'єкт має містити власні атрибути, у яких визначаються базові властивості (наприклад: температура, вологість, тиск, площа, об'єм, маса, швидкість тощо). Оскільки кожний об'єкт може бути як окремим елементом простору, так і конкретним шаром, то потрібна чітка прив'язка до простору та часу. Для моделювання складної ЕІС недостатньо мати інформацію лише про структуру та позицію об'єкта [10]. Інколи може постати гостра проблема визначення реакції груп об'єктів на певні штучно зсимульовані ситуації. Для цього слід використовувати вбудовані методи поведінки об'єктів, що для кожного із багатьох є специфічними. Для прикладу можна розглянути модель каменепаду. Кожен із каменів подається багатогранною фігурою. Кожна грань опуклого багатогранника може бути основою, якщо його поставити на горизонтальну поверхню. У правильного багатогранника центр ваги розташований всередині, так що він стійкий, якщо поставити його на будь-яку грань. Неправильні багатогранники можуть бути нестійкі, якщо встановлені на певні грані, тобто якщо їх поставити паралельно до основи, вони будуть перевертатися, а отже, неможливо застосовувати ті самі правила для складних фігур. Використання вбудованих методів поведінки дасть максимально чіткі результати при симуляції складних процесів. Завдяки вбудованим методам можна також спростити симуляцію процесів всередині ЕІС, розглядаючи її як одну із функцій ЕІС. Маючи можливість об'єднання різноманітних моделей даних [11] та їх структур у інші, можемо створювати новий тип або модель даних. Створювана модель даних може бути двох видів: кінцева та проміжна. Проміжний тип даних призначений для тимчасового утримання цих даних та подальшого формування іншої моделі даних. Відповідна кінцева модель даних – це модель, що отримана в результаті розрахунків і може бути використана для візуалізації окремих тематичних наборів об'єктів. Відповідно до кожного із типів даних, інформацію слід зберігати роздільно, дотримуючись чіткої ізоляції рівнів, щоб запобігти отриманню хибних результатів. Екологічна інформаційна система має зберігати передовсім базові дані та дані, специфічні для конкретної ЕІС. Усі дані зберігаються у базах даних, що можливо умовно розділити на такі типи:

а) база екоданих – це просторова база даних, що містить набори даних, які відображають екологічну інформацію у контексті загальної моделі даних ЕІС. До цих даних потрібно зарахувати векторні об'єкти, векторні зображення, растри, топологію, мережі, тривимірні об'єкти, а також усі об'єкти, що входять до складу тематичних наборів даних і є однією зі складових моделей даних ЕІС;

б) база ековізуалізації – це набір інтелектуальних карт, які показують просторові об'єкти та функціональну взаємодію між об'єктами на земній поверхні. У цій базі можуть бути побудовані різні види карт і можуть використовуватися як "вікна в базу даних" [12] для підтримки запитів, аналізу та редагування інформації;

в) тип екообробки ЕІС – це набір інструментів для одержання нових наборів екологічних даних з наявних наборів даних. Функції просторових даних отримують інформацію з наборів даних, застосовують до них аналітичні функції і записують одержані результати в нові похідні набори даних. Прикладом може бути специфічний тематичний шар, для створення якого було відібрано дані інших тематичних шарів.

Для прикладу, сьогодні уже створені бази даних географічної ІС (ГІС). Так в програмному забезпеченні ESRI® ArcGIS® ці три види баз даних подані каталогом (ГІС як колекція наборів геоданих), картою (ГІС як інтелектуальний картографічний вид) і набором інструментів (ГІС як набір інструментів для інтелектуальної обробки просторових даних). Всі вони є невід'ємними складовими повноцінної ГІС і більшою чи меншою мірою використовуються у всіх ГІС-додатках.

Якщо комплексно розглядати ЕІС як одне ціле, то це особливий тип бази даних про навколишній світ – екологічна база даних (база екоданих), основою якої є структурована база

даних, яка описує світ, та об'єкти, наявні в ньому, у географічному, економічному, кліматичному, ландшафтному, соціальному та інших аспектах. Створюючи дизайн бази екоданих ЕІС, визначають, як відобразатимуться різні просторові об'єкти. Наприклад, земельні ділянки зазвичай представляються як полігони, вулиці – як центральні лінії, свердловини – як точки тощо. Ці об'єкти групують у класи об'єктів, в яких кожен набір має єдине екологічне відображення. Кожен набір даних ЕІС дає просторове уявлення про якийсь аспект навколишнього світу, зокрема:

- впорядковані набори векторних об'єктів (набори точок, ліній та полігонів);
- набори растрових даних, такі як цифрові моделі рельєфу місцевості або зображення лісу;
- просторові мережі, такі як сільськогосподарські угіддя, водойми, ліси, виробничі комплекси;
- топографію місцевості та інші трьохвимірні об'єкти;
- набори даних екологічного аналізу та статистичної їх обробки;
- інші типи даних, такі як, назви вулиць, амбулаторні картки населення міст і сіл, картографічна інформація та будь-яка інша інформація, необхідна для певного тематичного шару.

Але призначення ЕІС – не лише операції зі статичними даними, але і синтез власних динамічних даних. Як приклад можна розглянути віртуально підвищений рівень опадів у певному регіоні, для перевірки міцності та надійності захисних дамб на прилеглих річках, тим самим унеможливаючи екологічні загрози. Для таких оцінок потрібно використовувати не лише програмну частину, а й апаратну у вигляді датчиків, сенсорів тощо. Моделюючи тематичні ландшафти на основі показників, отриманих під час дистанційного зондування, можна проаналізувати ситуації у регіонах. Такі тематичні ландшафти можуть відображати інформацію у різні моменти часу, оскільки мають просторово-часову прив'язку. Це дає змогу, аналізуючи дані за певні періоди, передбачати і прогнозувати ситуації.

Зазвичай дані є основою формування нових даних, які отримують через спеціальні функціональні залежності :

$$Data_n = Data + F(x),$$

де $Data_n$ – нові дані, $Data$ – існуючі дані; $F(x)$ – функціонал обробки даних.

Інструментальна база ЕІС дозволяє виконувати багатокрокові операції. Інструменти перетворення застосовують операції до вже наявної інформації з метою отримання нових даних. Середовище екообробки використовують у ЕІС для послідовного виконання серії таких операцій. Операції, з'єднані в єдиний ланцюжок, формують модель процесу обробки даних. Така єдина послідовність виконання операцій потрібна в ЕІС для автоматизації виконання багатократних завдань екообробки. Розробка і застосування подібних процедур і називається екообробкою даних. Екообробка використовується для моделювання процесів передачі даних з однієї структури в іншу з метою виконання багатьох стандартних задач ЕІС, зокрема, для імпорту даних з різних форматів, інтегрування цих даних в ЕІС, для стандартних процедур перевірки якості імпортованих даних. Можливість автоматизації та повторного виконання таких робочих процесів є перевагою ЕІС. Вона широко застосовується у численних ЕІС-додатках і сценаріях роботи з даними. Механізм, який використано для побудови робочих потоків при екообробці, повинен виконувати ряд команд в певній послідовності. Користувачі екоінформаційних систем мають мати можливість моделювати графічні процеси за допомогою інтерфейсів, що відповідають за візуалізацію. Складною вимогою до таких інтерфейсів у екоінформаційній системі є можливість інтеграції скриптів, що побудовані на певній, специфічній у межах однієї ЕІС мові програмування. Користувачі повинні мати можливість писати власні скрипти за допомогою таких інструментів програмування, як Python, VBScript і JavaScript. Екообробка широко використовується на всіх етапах роботи з ЕІС для автоматизації та компіляції даних, управління, аналізу і моделювання даних, або іншому місці, що пов'язано з кількістю вихідної інформації. Однак база даних ЕІС все одно має залишатися доступною для підтримки щоденної роботи і поточних оновлень, а користувачі повинні мати можливість звертатися до своїх версій та до версії загальної бази даних ЕІС. Розподілена сутність ЕІС передбачає широкі можливості для взаємодії між багатьма організаціями і системами опрацювання екологічної інформації. Співпраця та спільна робота користувачів дуже важливі для створення єдиної ЕІС. Багато екологічних наборів даних можуть

компілюватись і керуватись як загальний інформаційний ресурс і спільно використовуватись спільною користувачів. До того ж користувачі ЕІС мають власне бачення того, як можна здійснювати обмін популярними наборами даних через web-сайти. Ключові web-сайти, що є порталами каталогів ЕІС, надають можливість користувачам як викладати власну інформацію, так і шукати доступну для використання екологічну інформацію. У результаті ЕІС-системи все більше приєднуються до “Всесвітньої павутини” і отримують нові можливості обміну і використання інформації. Це бачення вкоренилось у свідомість людей за останнє десятиліття і знайшло відображення в таких поняттях, як Національна інфраструктура просторових даних (NSDI) і Глобальна інфраструктура просторових даних (GSDI) [12]. Ці концепції постійно розвиваються і поступово впроваджуються, причому не тільки на національному та глобальному рівнях, але також на рівні різних муніципальних утворень. В узагальненому вигляді ці концепції інтегровані в поняття інфраструктури просторових даних (SDI, Spatial Data Infrastructure) [13].

Особливості розроблення та напрями застосування ЕІС на основі інформаційно-комп'ютерних технологій. Результати проведеного аналізу вимог та підходів до побудови екоінформаційних систем засвідчують доцільність застосування комп'ютерних інструментальних засобів формування вхідних даних та їх інтелектуальної обробки з метою отримання нових даних та вироблення управлінських команд та формування рішень про стан екологічної системи та необхідних заходів для її корегування.

Одним з найвужчих місць під час розроблення KEIC є неповнота наборів достовірних і вивірених моделей опрацювання вхідних та вихідних сигналів, надійних повністю автоматизованих систем прийняття рішень [8] та універсальних інтелектуальних інтерфейсів спряження даних різного формату.

Оскільки ЕІС – це інформаційна системи, яка поєднує комплекс функцій, таких як отримання, зберігання, кодування, передавання, обробка, аналіз, захист та візуалізація територіально-орієнтованої інформації та екопросторових даних, які можуть бути використані при побудові інших інформаційних систем. У межах ЕІС провадяться дослідження не тільки над отриманою екологічною інформацією про навколишнє середовище, а і над усіма процесами та явищами на земній поверхні, економіці та у суспільстві. Проблеми, які вирішуються з використанням KEIC, з успіхом можуть бути використані для забезпечення ефективності економічного розвитку регіону, створення та ведення кадастрів природних ресурсів та нерухомості, здійснення екологічного моніторингу та природоохоронних заходів, запобігання та ліквідація наслідків надзвичайних ситуацій, забезпечення містобудівної діяльності, побудові систем розпізнавання та ідентифікації невідомих об'єктів [14–16], управління освітою, медициною, енергетикою, транспортом, житлово-комунальним господарством, сферою послуг, рекреаційно-туристичним комплексом, земельними ресурсами, сільським господарством, водними та лісовими запасами, корисними копалинами та іншою сировинною базою. Потреба в інтелектуальному опрацюванні даних є стимулом для користувачів отримувати нові дані достовірними і швидкодіючими способами, зокрема, додаючи частини баз даних для своїх ЕІС на основі даних інших розробників та користувачів.

Висновки

У статті сформульовано основні вимоги та запропоновано підходи до побудови екоінформаційних систем з метою забезпечення достовірного моніторингу навколишнього середовища та захисту його від кризових ситуацій та створення комфортних умов життя і праці населення регіону.

Запропоновано підходи до створення моделей інтелектуальної обробки даних в екологічних інформаційних системах та створення власних моделей отримання нових знань. Відзначено основні напрями подальшого розвитку та вдосконалення екоінформаційних автоматизованих систем на основі застосування сучасних інформаційно-комп'ютерних технологій та визначені сфери їх оптимального використання та експлуатації.

1. Шарковський С.Ф., Коляда А.Г., Сивак В.В., Федоренко А.Н. Динамика одномерных отображений. – К.: Наук. думка, 1989. – 216 с. 2. Заяць В.М. Построение и анализ модели дискретной колебательной системы // Кибернетика и системный анализ. – 2000. – № 4. – С. 161–165. 3. Заяць В.М. Дискретні моделі коливних систем для аналізу їх динаміки. – Львів: Вид-во УАД, 2011. –

278 с. 4. Заяць В.М. Аналіз динаміки та умов стійкості дискретних моделей коливних систем // Вісник Нац. ун-ту „Львівська політехніка”: "Інформаційні технології та мережі". – 2004. – № 519. – С. 132–142. 5. Погребенник В., Мельник М., Бойчук М. Екологічний моніторинг: концепції, принципи, системи // Вимірювальна техніка та метрологія. – Львів, 2005. – Вип. 65.– С. 164–171. 6. Экоинформатика. Теория. Практика. Методы и системы / Под ред. акад. РАН В.Е. Соколова. – СПб.: Гидрометеоздат, 1992. – 520 с. 7. www.dux.ru/eco/handbook/WIN/ / Растоскуев В.В. Информационные технологии экологической безопасности, 2000. – 225 с. 8. Цветков В.Я. Геоинформационные системы и технологии.– М.: Финансы и статистика, 1998.– 228 с. 9. Інформатизація космічного землезнавства / Під ред. О.І. Калашикова, Л.В. Сивай. – К.: Наукова думка, 2001. – 606 с. 10. Красовский Г.Я. Петросов В.А. Інформаційні технології космічного моніторингу водних екосистем і прогнозу водоспоживання міст. – К.: Наук. думка, 2003. – 224 с 11. Крета Д.Л., Перминова С.Ю. Особенности синтеза системы картографического обеспечения управления экологической безопасностью в Херсонской области // Ученые записки Таврического национального университета. – Симферополь, 2007.– Т.20 (59), № 1.– С. 90–97. 12. Постанова Кабінету Міністрів України № 1198 від 3 серпня 1998 року "Про єдину державну систему запобігання і реагування на надзвичайні ситуації техногенного та природного характеру".– 11 с. 13. Волошкіна О.С., Перминова С.Ю., Романенко Г.М. До питання розрахунку міграції забруднюючих речовин в межах зон санітарної охорони підземних водозаборів // Екологія і ресурси: Зб. наук. праць Інституту проблем національної безпеки. – К.: ІПНБ, 2007. – № 16. – С. 69–83. 14. Заяць В.М, Заяць М.М. Математичний опис системи розпізнавання користувача комп'ютера // Фізико-математичне моделювання та інформаційні технології. –2005. – № 2.– С. 146–152. 15. Фукунага К. Введение в статистическую теорию распознавания. – М.: Наука, 1979. – С. 512. 16. Zayats V., Ivanov D. Structural method of hand-written text recognition. Pros. International Conf // The experience of designing and application of CAD systems in microelectronics. – Lviv-Polyana, 12–15 may 2005. P. 493–494.

ЗАСТОСУВАННЯ ТЕОРІЇ ЕЛЕГАНТНИХ ОБЕРТОВИХ СИМЕТРІЙ-АСИМЕТРІЙ ДЛЯ СТВОРЕННЯ НОВІТНІХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

© Різник В., 2012

Розглядається загальносистемний метод створення ефективних інформаційних технологій на основі використання чудових структурних властивостей «елегантних» ансамблів обертової симетрії-асиметрії, притаманних реальному простору-часу. Виводяться основні математичні залежності параметрів одно- та багатовимірних структурних моделей для побудови нового класу багатовимірних (векторних) циклічних кодів з оптимальним розподілом вагових розрядів за критерієм кодування векторних даних у полях багатовимірних матриць.

Ключові слова: елегантна симетрія-асиметрія, структурна модель, багатовимірний монолітний код, натуральний ряд чисел, антенна решітка, гармонія.

A general systematic method for design of effectiveness information technologies using remarkable structural properties of “elegant” symmetry and asymmetry in real space -time, is suggested. There are showed basic mathematical dependences of parameters of one- and multidimensional structural models for synthesis a new classes of multidimensional (vector data) cyclic codes with optimum distributed of digit weights by encoding criterion of vector data in the multidimensional matrix field.

Key words: elegant symmetry-asymmetry, structural model, multidimensional monolithic code, natural numerical row, antenna array, harmony.

Вступ

Однією із кардинальних проблем сучасної науки є питання про всеосяжну гармонію світобудови та відповідності вимірів людини вимірам природи [1]. Ці питання в широкому контексті філософського та природничого знання перебувають у нерозривному взаємозв'язку з поняттям симетрії-асиметрії, що є фундаментальним принципом вивчення найважливіших закономірностей реального світу. Не менш актуальною є проблема виховання гармонійної людини. Ця широта проблематики свідчить про універсальність гармонії як важливої естетичної категорії. Тому актуальним постає дослідження взаємозв'язку між обертовою симетрією та асиметрією системних об'єктів для удосконалення інформаційних технологій та елементів систем з використанням методів комбінаторного аналізу, алгебричної теорії чисел та математичного апарата теорії розширених полів Галуа. Розглядається можливість створення на основі властивостей симетрії-асиметрії загальної теорії систем з мінімальною інформаційною, структурною й алгоритмічною надмірністю для системних об'єктів будь-якої фізичної природи. Загальноновизнаної науково обґрунтованої концепції подолання проблеми надмірності систем сьогодні немає. Цим визначається актуальність проблеми, для подолання якої доводиться стикатися з такими базовими поняттями теорії систем, як: симетрія та асиметрія, топологічна структура та порядок, надійність та надмірність, комбінаторна різноманітність та сумірність, гармонія пропорцій тощо, а також «інформація» як атрибут матерії, що відображає її структуру, й обіймає всю сучасну науку.

Аналітичний огляд найпростіших топологічних структур

Геометричну довершеність реального простору зручно досліджувати за допомогою порівняння двох різновидів симетричних фігур – з ланцюжковою і кільцевою структурами, кожна з яких обіймає однакову кількість n рівновіддалених між собою точок. Розглянемо два різновиди симетричних фігур – з ланцюжковою (а) та кільцевою (б) структурами (рис. 1), кожна з яких обіймає однакову кількість n рівновіддалених між собою елементів. Знайдемо кількість способів розбиття кожної з цих послідовностей на дві підмножини так, щоб усі елементи в кожній з підмножин залишалися зв'язаними.



Рис. 1. Симетричні фігури з ланцюговою (а) та кільцевою (б) структурами

За умовою постановки задачі ланцюг з n елементів не дозволяється розривати більше ніж в одному з $n - 1$ наявних зв'язків, тому кількість способів S_l розбиття цієї системи на дві підсистеми вичерпується числом:

$$S_l = (n - 1) \quad (1)$$

У кільцевій структурі з n елементами можна розривати будь-яку пару зв'язків між ними, не порушуючи умови задачі, тому кількість способів такого розбиття визначається числом неупорядкованих комбінацій «2 із n »:

$$S_k = (n - 1)/2 \quad (2)$$

Отже, система з кільцевою структурою, на відміну від ланцюгової, забезпечує можливість реалізації в $n/2$ разів більшої кількості способів декомпозиції її на підмножини без порушення зв'язків між елементами всередині обох новоутворених підсистем, а значить без додаткових дій та енергетичних втрат всередині системи, де число S_k збігається з кількістю очікуваних симетричних відстаней між цими елементами, положення кожного з яких описується t координатами в t -вимірному просторі. Ця властивість систем з кільцевою структурою відкриває шлях для розв'язування творчих задач, де потрібно мінімізувати число елементів системи без погіршення інших параметрів, і тому відповідає означенню «інтелектуальна система» [2].

Елегантний ансамбль «обертова симетрія-асиметрія». Сучасна наукова картина світобудови розглядається з позицій єдності симетрії та асиметрії. Однак в більшості наукових праць, пов'язаних з дослідженням цієї єдності, замало уваги наділяється семантиці інформаційної структури «обертова симетрія – асиметрія». Водночас, саме довершеність структурного взаємозв'язку обертової симетрії та асиметрії завдячує інформації про можливість досконало організованого розгортання нижчих циклічних вимірів простору-часу у вищі та його згортання до як завгодно *a priori* малих циклічних вимірів у вигляді скручених кільцевих струн. Слід зазначити, що на відміну від традиційного розуміння поняття «відстань», що згідно з математичним формулюванням геометрії Рімана може бути як завгодно малою, з погляду квантової геометрії і теорії струн сенс цього поняття пов'язаний з довжиною Планка і виходить за межі семантики.

Під «елегантним ансамблем» будемо розуміти систему «обертова симетрія – асиметрія», яка складається з двох «вкладених» одна в одну асиметричних фігур зі спільним центром, кожна з яких сама по собі генерує фіксовану кількість кутових відстаней, кратних числам натурального ряду з кроком квантування $\alpha_{\min} = 360^\circ / (1 - n + n^2)$. Тривіальним випадком елегантного ансамблю є фігура, яка має обертову симетрію порядку $S = (1 - n + n^2) = 3$, де $n = 2$ (рис. 2).

Елегантний ансамбль «обертова симетрія-асиметрія» (рис.2) утворює об'єднані спільною центральною точкою дві кругові шкали для відліку плоских кутів, кратних натуральному ряду з кроком квантування $(1) \cdot 360^\circ$ та $(1/3) \cdot 360^\circ$ відповідно, де множники 1 і $1/3$ – це перші члени ряду «елегантних» пропорцій: $1/1, 1/3, 1/7, 1/13, \dots, 1/S$. Суть ідеї полягає в досягненні максимальної

комбінаційної різноманітності утворених пропорцій, кратних натуральному ряду, яка притаманна, наприклад, «ідеальним» лінійкам [3] та «безнадмірним» круговим шкалам [4]. Оскільки йдеться про системний підхід, ця закономірність стосується симетричних об'єктів і процесів будь-якої фізичної, біологічної, чи іншої природи.

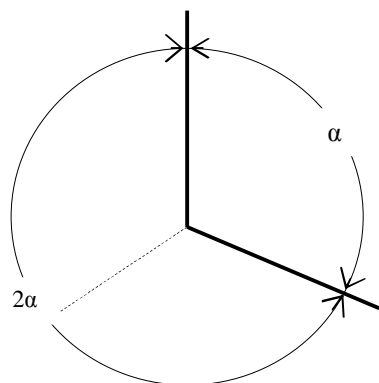


Рис. 2. Елегантний ансамбль «обертובה симетрія – асиметрія» третього ($S = 3$) порядку

Застосування «елегантних» ансамблів обертової симетрії-асиметрії.

Відомо, що значна частина методів забезпечення надійності засобів обчислювальної техніки ґрунтується на надмірності – апаратній або інформаційній. Аналіз наявних методів забезпечення надійності на основі штучного введення апаратної надмірності, показав, що більшість з них призводять до погіршення показників швидкодії, складності, енергоспоживання; порушують ітераційність, модульність апаратури та роблять її менш технологічною з точки зору сучасної елементної бази. Значний внесок у вирішення проблеми створення надійних систем керування за допомогою інформаційної надмірності, зокрема кодування, зробили Дж. Фон Нейман, К. Шеннон, Дж. Поуен, С. Голомб, С. Виноградов, М.О. Гаврилов, О.П. Стахов, Ю.Г. та ін. Аналіз існуючих кодових систем свідчить, що їх

властивості залежать від рівня надмірності. Так, негативним наслідком збільшення надмірності є підвищення інтенсивності потоку помилок, збільшення ймовірності виникнення помилок, зменшення ймовірності їх корекції. Перспективним напрямком подолання проблеми надмірності систем є глибше вивчення ролі симетричних (еквідистантних) та асиметричних (нееквідистантних) комбінаторних структур як зручних математичних моделей для створення конкурентоздатних розробок. Використанням класичної теорії комбінаторних структур в системах перетворення сигналів і комунікаційних технологіях займалися Н.Дж.А. Слоан, Р.Ч. Боуз, Д.К. Рой-Чоудхурі, Р. Дж. МакУільямс, С. Голомб, Л.Ю. Копилович, Л.Г. Содін та інші вчені. Одним із завдань сучасних інформаційних технологій є синтез кодів з підвищеною корегувальною здатністю, швидкістю обробки та передачі інформації, а також захистом від несанкціонованого доступу на основі використання нееквідистантних комбінаторних структур. Проблема зводиться до оптимального розподілу (взаємного розміщення) інформаційних розрядів в просторово-часовому вимірі носія інформації так, щоб множина усіх відстаней між парами різних розрядів коду по можливості збігалася з натуральним рядом. Це дає змогу зменшити інформаційну надмірність коду, або звести її до теоретично обґрунтованого мінімуму («циклічний ІК-код»). На рис. 3 показані порівняльні характеристики потужностей кодів, побудованих на базі стандартного двійкового коду $N=2k-1$, «ідеальної лінійки» $L = n(n-1)/2$ [3] та «ідеального кутоміра» $G=n(n-1)$ [4], де n – кількість позначок, k – кількість розрядів коду, $k = n-1$.

Ідеальний кутомір – це пристрій для відмірювання кутових відстаней, виконаний у вигляді транспортира з нееквідистантною (нерівномірною) шкалою, позначки якої утворюють багатозначну кутову міру, усі відносні відстані між якими вичерпують натуральний ряд. Ідеальні кутоміри забезпечують можливість побудови кодів вдвічі більшої потужності порівняно з теоретичними можливостями ідеальних лінійок Голомба, існування яких припиняється вже на лінійці з чотирма позначками (1,3,2). Натомість, як показують теоретичні розрахунки, існує як завгодно багато ідеальних кутомірів та «ідеальних просторомірів», тобто досконалих комбінаторних конструкцій, елементами яких можуть бути не лише числа, але й числові кортежі. Йдеться про

можливість використання нашого досвіду, здобутого за час теоретичних й прикладних досліджень алгебричних комбінаторних структур, зокрема багатовекторних нееквідистантних структур, що може стати корисним для створення конкурентоспроможних технологій на світовому ринку.

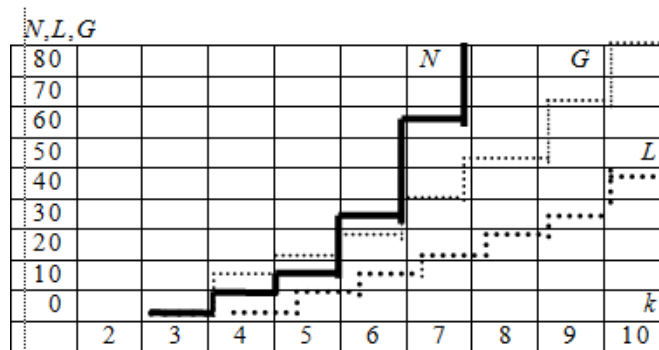


Рис. 3. Порівняльні характеристики потужності кодів: стандартного двійкового (N), побудованого за ідеальним кутоміром (G) та за ідеальною лінійкою (L) (реально не існує для $k \geq 4$).

Прикладом вдалого вирішення цієї проблеми в інформаційних та комунікаційних системах є створення нового класу кодів, названих «монолітними кодами» [5]. Для монолітного коду (МК) вводиться обмеження щодо розміщення «одиниць» і «нулів», за яким усі однойменні символи знаходяться поруч один до одного (за винятком меж, що розділяють «одиниці» і «нули»). МК набуває деяких істотних переваг за такими показниками, як швидкість формування комбінацій, завадостійкість, простота апаратної реалізації, простота перетворення форми коду в інший код тощо. Забезпечення максимальної потужності МК досягається завдяки відповідному розподілу вагових розрядів, здійснюваного аналогічно до правил розміщення позначок на шкалі ідеального кутміра. За таких умов МК вичерпує множину способів формування комбінацій, що одночасно зі збільшенням його потужності зводить до мінімуму інформаційну надлишковість. Під монолітним розуміють код, комбінації якого побудовані винятково на послідовностях інформаційних «одиниць», тому поява між ними хоча б одного «нуля» миттєво вказує на появу помилки, не потребуючи жодних додаткових дій (контрольних перевірок) і, отже, забезпечує надвисоку швидкодію щодо виявлення і виправлення помилок, збільшуючи інформаційну надійність.

Природним напрямком продовження досліджень є використання багатовимірних (векторних) монолітних кодів для створення інформаційних технологій і комп'ютерних систем на основі багатовимірної арифметики, що дозволить проектувати апаратно-програмні засоби вищого рівня порівняно з нині уживаними, а в перспективі створити високопродуктивні багатовекторні інформаційні та комунікаційні технології, що базуються на багатовекторних взаємно однозначних перетвореннях масивів даних та впровадженні векторної арифметики.

Розглянемо детальніше формування кодових комбінацій в МК та встановимо теоретичний зв'язок таких кодів з комбінаторними структурами на прикладі побудови двовимірних (2D) систем кодування.

Елементами такої системи є послідовність упорядкованих числових 2-кортежів $((k11, k21), (k12, k22), \dots, (k1i, k2i), \dots, (k1n, k2n))$, останній з яких знаходиться поруч першого. Йдеться про створення системи кодування двовимірних векторів, схема якої має вигляд петлі (рис. 4). На цьому рисунку можна побачити систему формування кодових 2-кортежів, де n – кількість 2-кортежів.

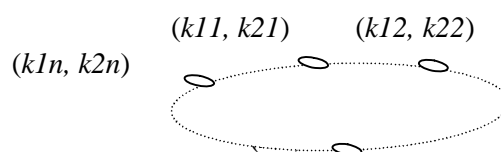


Рис. 4. Схема системи кодування двовимірних векторів

Поставимо задачу обрання цілочислових значень n -послідовності впорядкованих 2-кортежів $(k1i, k2i)$, враховуючи такі вимоги:

- 1) усі 2-кортежі не повинні повторюватися;
- 2) усі 2-модульні вектор-суми поруч розміщених 2-кортежів не повинні повторюватися;
- 3) множина усіх обраних 2-кортежів разом з усіма 2-модульними вектор-сумами поруч розміщених 2-векторів повинні заповнити координати двовимірної матриці.

З вимог 1) і 2) випливає система нерівностей:

$$(k11, k21) \neq (k12, k22) \neq \dots \neq (k1i, k2i) \neq \dots \neq (k1n, k2n) \neq (k11, k21) + (k12, k22) \neq \dots \neq (k1n, k2n) + (k11, k21) \neq \dots \neq (k11, k21) + (k12, k22) + \dots + (k1i, k2i) + \dots + (k1n, k2n).$$

Під 2-модульною вектор-сумою розуміють результат арифметичного додавання чисел, обраних від кожного з n 2-кортежів, числа яких мають однойменні порядкові номери, причому додають за відповідними модулями.

Нехай $(k11, k21) = (1,1)$, $(k12, k22) = (1,2)$, $(k13, k23) = (1,4)$, $(k14, k24) = (1,3)$. Тоді числова модель системи двовимірного кодування 4-го порядку ($n=4$) з кільцевою структурою набуває такого вигляду:

$$\begin{matrix} (1,1) \\ (1,2) & (1,4) \\ (1,3) \end{matrix}$$

Знайдемо всі 2-модульні вектор-суми на цій кільцевій послідовності :

$$\begin{aligned} (k11, k21) + (k12, k22) &= ((k11+k12), (k21+k22)) = ((1+1), (1+2)) = (2,3); \\ (k12, k22) + (k13, k23) &= ((k12+k13), (k22+k23)) = ((1+1), (2+4)) = (2,6); \\ (k13, k23) + (k14, k24) &= ((k13+k14), (k23+k24)) = ((1+1), (4+3)) = (2,7); \\ (k14, k24) + (k11, k21) &= ((k14+k11), (k24+k21)) = ((1+1), (3+1)) = (2,4); \\ (k11, k21) + (k12, k22) + (k13, k23) &= ((k11+k12+k13), (k21+k22+k23)) = ((1+1+1), (1+2+4)) = (3,7); \\ (k12, k22) + (k13, k23) + (k14, k24) &= ((k12+k13+k14), (k22+k23+k24)) = ((1+1+1), (2+4+3)) = (3,9); \\ (k13, k23) + (k14, k24) + (k11, k21) &= ((k13+k14+k11), (k23+k24+k21)) = ((1+1+1), (4+3+1)) = (3,8); \\ (k14, k24) + (k11, k21) + (k12, k22) &= ((k14+k11+k12), (k24+k21+k22)) = ((1+1+1), (3+1+2)) = (3,6); \\ (k11, k21) + (k12, k22) + (k13, k23) + (k14, k24) &= ((1+1+1+1), (1+2+4+3)) = (4,10). \end{aligned}$$

В обчисленнях слід врахувати значення модулів $m1=3$, $m2=4$:

$$\begin{aligned} (k11, k21) + (k12, k22) &= ((k11+k12), (k21+k22)) = (2,3); \\ (k12, k22) + (k13, k23) &= ((k12+k13), (k22+k23)) = (2,1); \\ (k13, k23) + (k14, k24) &= ((k13+k14), (k23+k24)) = (2,2); \\ (k14, k24) + (k11, k21) &= ((k14+k11), (k24+k21)) = (2,4); \\ (k11, k21) + (k12, k22) + (k13, k23) &= ((k11+k12+k13), (k21+k22+k23)) = (3,2); \\ (k12, k22) + (k13, k23) + (k14, k24) &= ((k12+k13+k14), (k22+k23+k24)) = (3,4); \\ (k13, k23) + (k14, k24) + (k11, k21) &= ((k13+k14+k11), (k23+k24+k21)) = (3,3); \\ (k14, k24) + (k11, k21) + (k12, k22) &= ((k14+k11+k12), (k24+k21+k22)) = (3,1); \\ (k11, k21) + (k12, k22) + (k13, k23) + (k14, k24) &= (0, 0). \end{aligned}$$

Результати обчислень показують, що кільцева послідовність $((1,1), (1,2), (1,4), (1,3))$ утворює матрицю з розмірами 3×4 , яка містить усі кільцеві вектор-суми, числові значення яких вичерпують значення її координат. Двовимірній кільцевій структурі відповідає система кодування двовимірних векторів за правилами монолітного коду. Наприклад, 2-вимірна кільцева конструкція на послідовності 2-місних кортежів $((1,1), (1,2), (1,4), (1,3))$ утворює досконалу з погляду досягнення мінімальної інформаційної надмірності монолітного коду систему кодування двовимірних векторів на матриці з розмірами 3×4 , що ілюструє табл. 1.

З табл. 1 можна бачити, що 4-розрядні кодові комбінації вичерпують цілочислові значення двовимірних векторів в межах від $(1,1)$ до $(3,4)$, закодованих у монолітному коді. Ця властивість дає змогу представляти в монолітному коді будь-який двовимірний вектор, обмежений рамками матриці 3×4 . Описаний код є оптимальним кодом за критерієм мінімізації числа розрядів з обмеженням на правила формування однойменних символів в дозволених кодових комбінаціях, де всі однойменні символи знаходяться поруч один одного. Монолітний код дає змогу отримати усі можливі значення двовимірних векторів.

Таблиця 1

Кодування двовимірних векторів у системі $((1,1),(1,2),(1,4),(1,3))$

Вектор	Кодова комбінація			
(1,1)	1	0	0	0
(1,2)	0	1	0	0
(1,3)	0	0	0	1
(1,4)	0	0	0	1
(2,1)	0	1	1	0
(2,2)	0	0	1	1
(2,3)	1	1	0	0
(2,4)	1	0	0	1
(3,1)	1	1	0	1
(3,2)	1	1	1	0
(3,3)	1	0	1	1
(3,4)	0	1	1	1

Наступний приклад демонструє різновид досконалої системи монолітного коду двовимірних векторів в діапазоні цілочислових значень від 0 до 2 – для першої, та від 0 до 3 – другої складової двовимірного вектора:

$$(0, 0) \equiv (2, 2) + (0, 2) + (1, 0),$$

$$(0, 1) \equiv (1, 1) + (2, 2) + (0, 2),$$

$$(0, 2) = (0, 2),$$

$$(0, 3) \equiv (1, 1) + (2, 2),$$

$$(1, 0) = (1, 0),$$

$$(1, 1) = (1, 1),$$

$$(1, 2) \equiv (0, 2) + (1, 0),$$

$$(1, 3) \equiv (1, 0) + (1, 1) + (2, 2),$$

$$(2, 0) \equiv (2, 2) + (0, 2),$$

$$(2, 1) \equiv (1, 0) + (1, 1),$$

$$(2, 2) = (2, 2),$$

$$(2, 3) \equiv (0, 2) + (1, 0) + (1, 1)$$

У табл. 2 наведена схема кодування двовимірних векторів на кільцевій послідовності 2-кортежів. Перша компонента вектора береться по модулю $m_1 = 3$, а друга – по $m_2 = 4$.

Таблиця 2

Монолітний код двовимірних векторів на кільцевій послідовності $((0,2),(1,0),(1,1),(2,2))$

Вектор	Кодова комбінація			
(0,0)	1	1	0	1
(0,1)	1	0	1	1
(0,2)	1	0	0	0
(0,3)	0	0	1	1
(1,0)	0	1	0	0
(1,1)	0	0	1	0
(1,2)	1	1	0	0
(1,3)	0	1	1	1
(2,0)	1	0	0	1
(2,1)	0	1	1	0
(2,2)	0	0	0	1
(2,3)	1	1	1	0

Порівнюючи між собою вищенаведені приклади кодування двовимірних векторів, можна побачити, що обидві системи перетворення форми інформації рівноцінні щодо досягнення максимальної потужності коду, оскільки кожна з них вичерпує множину координат двовимірної матриці 3×4 . Відмінність полягає лише в тому, що в системі кодування, яка базується на кільцевій послідовності векторів $((1,1),(1,2),(1,4),(1,3))$, одна з координат двовимірного вектора може набирати значень чисел натурального ряду від 1 до 3, а друга – від 1 до 4, тоді як для системи $((0,2),(1,0),(1,1),(2,2))$ – відповідно від 0 до 2 та від 0 до 3. Слід відзначити, що обидві системи є досконало сформованими інформаційними структурами, в яких віддзеркалена предвічна гармонія двовимірної «розгортки» реального простору.

Розглянемо один з варіантів системи кодування, яка базується на кільцевій послідовності векторів $((0,1,0),(0,2,3),(1,1,2),(0,2,2),(1,0,3),(1,1,1))$, де одна з координат 3D-вектора набирає значень цілих чисел $\{0,1\}$, друга – $\{0,1,2\}$, третя – $\{0,1,2,3,4\}$. В обчисленнях слід врахувати значення модулів $m_1=2, m_2=3, m_3=5$:

$$\begin{aligned} (0, 0, 0) &\equiv (0,1,0) + (0,2,3) + (1,1,2) + (0,2,2) + (1,0,3), \\ (0, 0, 1) &\equiv (0,2,2) + (1,0,3) + (1,1,1), \\ (0, 0, 2) &\equiv (1,1,2) + (0,2,2) + (1,0,3), \\ (0, 0, 3) &\equiv (0,1,0) + (0,2,3), \\ (0, 0, 4) &\equiv (0,2,2) + (1,0,3) + (1,1,1) + (0,1,0) + (0,2,3), \\ (0, 1, 0) &\equiv (0,1,0), \\ (0, 1, 1) &\equiv (0,2,2) + (1,0,3) + (1,1,1) + (0,1,0), \\ &\text{і т.д.} \end{aligned}$$

.....
 $(1, 2, 4) \equiv (1,0,3) + (1,1,1) + (0,1,0) + (0,2,3) + (1,1,2).$

Система кодування 3D-векторів на базі кільцевої послідовності $((0,1,0), (0,2,3), (1,1,2), (0,2,2), (1,0,3), (1,1,1))$ приведена в табл. 3.

Таблиця 3

Монолітний 3D- код на кільцевій послідовності $((0,1,0), (0,2,3), (1,1,2), (0,2,2), (1,0,3), (1,1,1))$

Вектор	Кодова комбінація						Вектор	Кодова комбінація					
(0,0,0)	1	1	1	1	1	0	(1,0,0)	0	1	1	0	0	0
(0,0,1)	0	0	0	1	1	1	(1,0,1)	0	1	1	1	1	1
(0,0,2)	0	0	1	1	1	0	(1,0,2)	1	1	1	1	0	0
(0,0,3)	1	1	0	0	0	0	(1,0,3)	0	0	0	0	1	0
(0,0,4)	1	1	0	1	1	1	(1,0,4)	0	0	1	1	0	0
(0,1,0)	1	0	0	0	0	0	(1,1,0)	1	1	1	0	0	0
(0,1,1)	1	1	0	0	1	1	(1,1,1)	0	0	0	0	0	1
(0,1,2)	0	0	1	1	1	1	(1,1,2)	0	0	1	0	0	0
(0,1,3)	1	1	1	1	0	1	(1,1,3)	0	0	1	1	1	1
(0,1,4)	0	0	0	0	1	1	(1,1,4)	1	1	0	0	0	1
(0,2,0)	0	1	1	1	1	0	(1,2,0)	0	0	0	1	1	0
(0,2,1)	1	1	1	0	0	1	(1,2,1)	1	0	0	0	0	1
(0,2,2)	0	0	0	1	0	0	(1,2,2)	0	1	1	1	0	0
(0,2,3)	0	1	0	0	0	0	(1,2,3)	1	0	1	1	1	1
(0,2,4)	1	0	0	0	1	1	(1,2,4)	1	1	1	0	1	1

З табл. 3 випливає, що кільцева послідовність $((0,1,0), (0,2,3), (1,1,2), (0,2,2), (1,0,3), (1,1,1))$ утворює кодову матрицю з розмірами $2 \times 3 \times 5$, що містить усі кільцеві вектор-суми, числові значення яких вичерпують значення її координат. Описана система кодування 3D-векторів є монолітним кодом.

На графіках (рис. 5, а, б) наведені діаграми направленості прямолінійних еквідистантної та нееквідистантної антенних решіток (АР). Легко побачити, що на відміну від еквідистантної АР (рис. 5, а), в

якої спостерігаються значний вплив на величину випромінювання сигналу в основному напрямі інтерференційні максимуми та мінімуми, діаграм направленості нееквідистантної АР (рис. 5, б), побудована за методом графо-аналітичних інтерпретацій для фіксованого числа випромінювачів ($n = 4$), забезпечує низький рівень енергії електромагнітного поля на бокових напрямках випромінювання та поліпшує діаграму направленості антенної решітки.

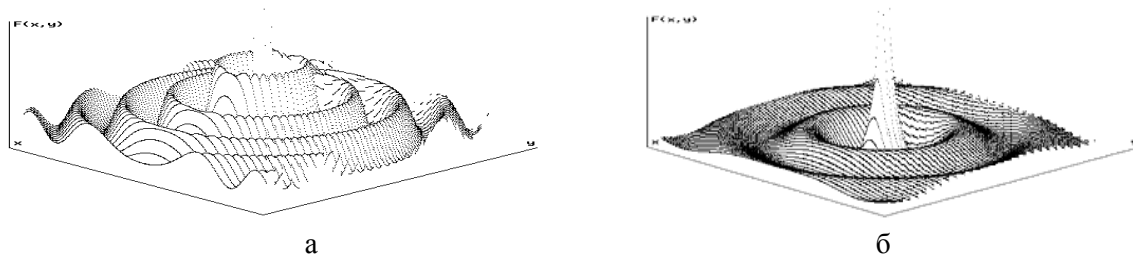


Рис. 5. Діаграми направленості прямолінійних еквідистантної (а) та нееквідистантної (б) антенних решіток для фіксованого числа елементів випромінювання ($n=4$).

Порівнюючи між собою діаграми, можемо констатувати, що на відміну від рівномірної (еквідистантної) нееквідистантна антенна решітка має нижчий рівень інтерференційних викидів і відповідно кращі характеристики за рівнем бічного випромінювання та коефіцієнтом направленої дії.

Іншим завданням є дослідження теоретичного взаємозв'язку між класичними комбінаторними конфігураціями та нестандартними комбінаторними структурами з метою досягнення максимальної простоти, наочності та практичної уживаності згаданих структур для розроблення на їхній основі інформаційних та комунікаційних технологій підвищеної надійності, завадостійкості, роздільної здатності, а також алгоритмів машинного синтезу багатоелементних комбінаторних структур.

Отримані результати передбачають розширення сфери досліджень в тих галузях науки і техніки, де впроваджуються загальносистемні принципи, що базуються на теорії комбінаторних конфігурацій: математиці (векторна алгебра, теорія груп), обчислювальній техніці, криптографії, інформаційно-вимірювальній техніці, комп'ютерних технологіях, радіофізиці, системах зв'язку та інших технічних галузях.

Висновки

Ідея застосування теорії елегантних ансамблів симетрій-асиметрій науково обґрунтована і перевірена на практиці. Вона створює нові перспективи розвитку прогресивних інформаційних і комунікаційних технологій, що є кращими за відомі в світі аналоги. Це підтверджено багатьма публікаціями у вітчизняних та зарубіжних виданнях, а також укладеною в 1996 р. угодою між Південно-каліфорнійським університетом (проф.С.Голомб) та Львівським політехнічним інститутом про співпрацю в галузі інформаційних технологій і системотехніки.

Результати проведених досліджень дають підстави стверджувати про можливість створення новітніх пристроїв та систем, які базуються на векторних інформаційних технологіях, й розроблення спеціалізованих процесорів на багатовимірній комп'ютерній арифметиці. Використання багатовимірних монолітних кодів дозволить проектувати апаратно-програмні засоби вищого рівня порівняно з нині уживаними. Опрацювання регулярних методів синтезу відкриває можливості для розгортання новітніх технологій в різних галузях науки і техніки на основі використання «закона ідеальних пропорцій». Існування цього закону є свідченням не підвладної часові всеосяжної досконалості структурної організації Всесвіту та розвитку природи за законами гармонії. Наявність численних варіантів багатовимірних комбінаторних конфігурацій свідчить про багатоманітність існуючих форм предвічної гармонії, яка притаманна геометрії реального простору-часу.

1. Вікіпедія. Естетичні категорії www.refine.org.ua/pageid-2461-1.html. 2. <http://translated.by/you/konspekt-lektsii-intellektual-nye-sistemy/original/>. 3. Martin Gardner. Golomb's Grateful Curve,

Life and Other Mathematical Amusements. W.H. Freeman and Company, 1983. 4. Бандирська О.В. Стандартизація безнадлижкових рядів методом оптимальних структурних пропорцій: Автореф. канд. техн. Наук. – Львів: ДУЛП, 2000. 5. Різник В.В. Синтез оптимальних комбінаторних систем. – Львів: Вища шк., 1989. – 168 с.

УДК 519.765:519.767:004.93

Б. Павлишенко

Львівський національний університет імені Івана Франка

КЛАСИФІКАЦІЯ ПОВІДОМЛЕНЬ ГРУП НОВИН У ВЕКТОРНОМУ ПРОСТОРІ СЕМАНТИЧНИХ ПОЛІВ

© Павлишенко Б., 2012

Розглянуто класифікацію повідомлень груп новин у просторі семантичних полів. Проаналізовано ефективність баєсівського класифікатора та класифікатора за найближчими сусідами для різних навчальних та тестових вибірок повідомлень. Показано існування підмножини груп новин, для яких використання аналізованих класифікаторів є ефективним.

Ключові слова: інтелектуальний аналіз даних, класифікація текстів, векторна модель текстів, семантичні поля.

The classification of newsgroup messages in the space of semantic fields has been considered in this work. The effectiveness of Bayesian and nearest neighbors classifier for different training and test samples of messages has been analysed. The existence of a subset of newsgroups for which the use of analyzed classifiers is effective has been shown.

Key words: data mining, text classification, vector space model of texts, semantic fields.

Вступ

У роботах [1–3] наведені результати аналізу текстових масивів на основі концепції семантичних полів. Семантичні поля розглядають як групи лексем, об'єднаних спільним поняттям. Такі групи лексем утворюють нові характеристики текстових даних, використання яких є ефективним у задачах кластеризації та класифікації текстових документів. Однією із поширених моделей в інтелектуальному аналізі текстових даних є векторна модель, в якій текстові документи представляють у вигляді векторів у деякому фазовому просторі [4]. Базис цього простору утворюють частотні характеристики лексем. У роботі [1] розглянута теоретико-множинна концепція семантичних полів в масивах текстових даних. У роботі [2] запропонована модель кластеризації текстових документів у семантичному просторі, яка дає можливість отримувати новий структурний поділ документів за семантичними ознаками у просторі істотно меншої розмірності, ніж у просторі, утвореному лексемним складом текстової вибірки. У задачах аналізу текстового змісту актуальними є теорії лексичної семантики, зокрема, вчення про семантичні поля. Спорідненими об'єктами у комп'ютерній інформатиці є семантичні мережі, в яких відображаються змістовні зв'язки між різними концептами. Одним із прикладів ієрархічно-організованої семантичної мережі можна розглядати систему WordNet, яка розроблена у Принстонському університеті [5]. Лексемний склад в цій системі організований у вигляді синсетів, під якими розуміють набори лексем синонімічного ряду, які є взаємозамінними у заданих контекстах. Бази даних WordNet створили експерти-лексикографи. Іменники, дієслова, прикметники та прислівники організовані у синсети – множини синонімів. Іменники та дієслова

згруповані відповідно до семантичних полів. Семантична структурна організація лексемного складу словника може бути використана у відповідних алгоритмах класифікації та кластеризації текстових об'єктів з точки зору зменшення розмірності задач аналізу та виявлення нових семантичних зв'язків в онтології предметної області, до якої відносять аналізований масив текстів. У роботі [6] введено поняття семантичного домена, який описує деяку семантичну область розгляду тої чи іншої теми обговорень, наприклад, економіка, політика, фізика, програмування тощо. Для розгляду алгоритмів текстової кластеризації часто використовують стандартизовані масиви текстових документів. Однією із таких колекцій є 20-NewsGroups [http://qwone.com/~jason/20NewsGroups/], яка включає у себе колекцію приблизно 20 тисяч документів близько 20 груп новин. Цю колекцію використовують у тестових задачах інтелектуального аналізу текстових масивів, зокрема у задачах класифікації та категоризації текстових масивів.

Постановка задачі

Розглянемо класифікацію текстових документів у просторі семантичних полів. Для порівняння розглянемо байєсівський класифікатор та класифікатор за найближчими к сусідами. Як навчальну та тестову вибірки використаємо повідомлення груп новин стандартизованої текстової бази даних 20NewsGroups. Проаналізуємо кількісні характеристики ефективності класифікаторів із різними параметрами текстових вибірок.

Класифікатори текстових документів у просторі семантичних полів. Сукупність текстових документів опишемо такою множиною:

$$D = \{d_j / j = 0, 1, 2, \dots, N_d\}. \quad (1)$$

Введемо множину семантичних полів

$$S = \{s_k / k = 1, 2, \dots, N_s\}. \quad (2)$$

Під семантичним полем розуміють таку множину лексем, у якій вони об'єднані певним спільним поняттям [1, 5, 6]. Прикладом семантичних полів може бути поле руху, поле комунікації, поле сприйняття та інші. Нехай існує певний словник лексем, які зустрічаються у текстових масивах $W = \{w_i / i = 1, 2, \dots, N_w\}$. Лексемний склад семантичного поля s_k визначимо як

$$W_k^s = \left\{ w_i / w_i \xrightarrow{U_{ws}} s_k, i = 1, 2, \dots, N_w \right\}. \quad (3)$$

Введемо частоту семантичного поля p_{kj}^{sd} за такою формулою:

$$p_{kj}^{sd} = \sum_{i=1}^{N_w} p_{ij}^{wd} f_s(w_i, s_k), \quad f_s(w_i, s_k) = \begin{cases} 1, & w_i \in W_k^s \\ 0, & w_i \notin W_k^s \end{cases}, \quad (4)$$

де p_{ij}^{wd} – текстова частота лексеми w_i в документі d_j , яка визначається відношенням наявної кількості лексеми w_i до загальної кількості лексем у документі d_j . Сукупність значень p_{kj}^{sd} утворюють матрицю ознака-документ, у якій ознаками виступають частоти семантичних полів у документах:

$$M_{sd} = \left(p_{kj}^{sd} \right)_{k=1, j=1}^{N_s, N_d}. \quad (5)$$

Вектор

$$V_j^s = \left(p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd} \right) \quad (6)$$

відображає документ d_j в N_s -мірному просторі текстових документів із базисом, утвореним семантичними полями.

Нехай існують деякі категорії текстових документів. Ці категорії можуть мати різну природу, наприклад, можуть визначати авторський ідеолокт, дискурс, характеризувати різні об'єкти, явища, події тощо. У нашому аналізі такі категорії утворюють групи новин. Множину цих категорій позначимо

$$Categories = \{ Ctg_m / m = 1, 2, \dots, N_{ctg} \}, \quad (7)$$

де $N_{ctg} = |Categories|$ визначає розмір множини категорій. За даними категоріями розподілені текстові документи множини D (1). Завдання полягає у пошуку цільової функції, яка описується відображенням

$$F_{d \rightarrow ctg} : Categories \times D \rightarrow \{0,1\} \quad (8)$$

Розглянемо наївний байесівський класифікатор текстових документів. У відомих методах текстової класифікації на основі наївної байесівського класифікатора використовують представлення документів за допомогою частот відповідних ключових слів [7, 8]. Підхід, який базується на представленні документів частотними характеристиками семантичних полів, є перспективним внаслідок меншої розмірності семантичного фазового простору. Знайдемо апостеріорну ймовірність того, що за деяким набором частот семантичних полів документ d_j належить до категорії ctg_m . За теоремою Байеса визначимо

$$P(ctg_m | p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd}) = \frac{P(ctg_m) P(p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd} | ctg_m)}{P(p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd})} \quad (9)$$

У реалізації наївного байесівського класифікатора роблять істотне припущення про умовну незалежність ознак об'єктів [7,8]. У такому випадку умовну ймовірність $P(p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd} | ctg_m)$ апроксимують добутком умовних ймовірностей $P(p_{ij}^{sd} | ctg_m)$. Неперервні розподіли $P(p_{ij}^{sd} | ctg_m)$ часто апроксимують нормальним гауссовим розподілом. Параметрами цього розподілу розглядають математичне сподівання та дисперсію семантичних полів. Доповненням до розрахунку наївного байесівського класифікатора є правило прийняття рішень про віднесення аналізованого документа до тієї чи іншої категорії [7,8]. У найпростішому випадку таке правило може приймати рішення про належність документа до заданої категорії, якщо розрахована апостеріорна ймовірність для такої категорії при заданих частотах семантичних полів є найбільшою, тобто

$$\begin{aligned} Category(d_j) &= ctg_m : P(ctg_m | p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd}) = \\ &= \max \left\{ P(ctg_k | p_{1j}^{sd}, p_{2j}^{sd}, \dots, p_{N_s j}^{sd}) \mid k = 1, 2, \dots, N_{ctg} \right\} \end{aligned} \quad (10)$$

Ймовірності $P(p_{ij}^{sd} | ctg_m)$ формують на деякому навчальному категоризованому масиві текстових документів.

Розглянемо класифікацію за найближчими к сусідами, яку називають kNN класифікацією [7, 8, 9]. Цей метод зараховують до векторних класифікаторів. В основі векторних методів класифікації лежить гіпотеза компактності. Згідно із цією гіпотезою, документи, які належать одному і тому ж класу утворюють компактну область, а області, які належать різним класам не перетинаються. Як міру близькості між документами виберемо евклідову відстань. Поряд із евклідовою відстанню часто використовують значення косинуса кута між двома векторами. У kNN класифікації межі категорій визначають локально. Деякий документ зараховують до категорії, яка є домінуючою для k його сусідів. У випадку $k=1$ документу приписують категорію його найближчого сусіда. Згідно із гіпотезою компактності тестовий документ d має ту категорію, яку мають більшість документів навчальної вибірки у деякому просторовому локальному околі документа d .

Розглянемо оцінки точності класифікації документів. Прийняття рішення класифікатором про належність документа d_i до категорії ctg_j позначимо $Class(d_i) = Ctg_j$. Множина документів, які визначені класифікатором як належні до категорії ctg_j і які дійсно належать цій категорії згідно з експертною оцінкою має вигляд

$$Set_1^{ctg_j} = \{d_i \mid Class(d_i) = Ctg_j \wedge d_i \in Ctg_j\}. \quad (11)$$

Множина документів, які визначені класифікатором як належні до категорії ctg_j , має вигляд

$$Set_2^{ctg_j} = \{d_i \mid Class(d_i) = Ctg_j\}. \quad (12)$$

Множина документів, які належать до категорії ctg_j має вигляд

$$Set_3^{tclass} = \{d_i \mid d_i \in Ctg_j\}. \quad (13)$$

Кількості елементів у множинах $Set_1^{tclass}, Set_2^{tclass}, Set_3^{tclass}$ визначаються кардинальними числами цих множин $|Set_1^{tclass}|, |Set_2^{tclass}|, |Set_3^{tclass}|$. Для характеристики класифікаторів використовують поняття точності (precision) та повноти (recall) [8,9]. Точність класифікатора визначають як відношення кількості елементів множини Set_1^{tclass} до елементів множини Set_2^{tclass}

$$Pr_j^{tclass} = \frac{|Set_1^{tclass}|}{|Set_2^{tclass}|} = \frac{|\{d_i \mid Class(d_i) = Ctg_j \wedge d_i \in Ctg_j\}|}{|\{d_i \mid Class(d_i) = Ctg_j\}|} \quad (14)$$

Повноту визначають як відношення кількості елементів множини Set_1^{tclass} до елементів множини Set_3^{tclass}

$$Rc_j^{tclass} = \frac{|Set_1^{tclass}|}{|Set_3^{tclass}|} = \frac{|\{d_i \mid Class(d_i) = Ctg_j \wedge d_i \in Ctg_j\}|}{|\{d_i \mid d_i \in Ctg_j\}|} \quad (15)$$

Індекс j у характеристиках $Pr_j^{tclass}, Rc_j^{tclass}$ визначає категорію, а індекс $tclass$ визначає тип класифікатора. У наших дослідженнях

$$tclass = \{NB, nKNN\} \quad (16)$$

Кожна категорія документів характеризується своїми значеннями $Pr_j^{tclass}, Rc_j^{tclass}$. Для загальної характеристики класифікатора знайдемо макроусереднення показників $Pr_j^{tclass}, Rc_j^{tclass}$ за усіма категоріями

$$Pr_{mean}^{tclass} = \frac{1}{N_{ctg}} \sum_{i=1}^{N_{ctg}} Pr_i^{tclass}, \quad (17)$$

$$Rc_{mean}^{tclass} = \frac{1}{N_{ctg}} \sum_{i=1}^{N_{ctg}} Rc_i^{tclass}. \quad (18)$$

Розкид показників $Pr_j^{tclass}, Rc_j^{tclass}$ за категоріями охарактеризуємо середньоквадратичним відхиленням

$$Pr_{std}^{tclass} = \sqrt{\frac{1}{N_{ctg}} \sum_{i=1}^{N_{ctg}} (Pr_i^{tclass} - Pr_{mean}^{tclass})^2}, \quad (19)$$

$$Rc_{std}^{tclass} = \sqrt{\frac{1}{N_{ctg}} \sum_{i=1}^{N_{ctg}} (Rc_i^{tclass} - Rc_{mean}^{tclass})^2}. \quad (20)$$

Очевидно, що чим більша кількість можливих категорій кластеризації, тим складніше завдання є перед класифікатором документів і тим більша ймовірність похибки, тому необхідно ввести деяку кількісну характеристику, яка б показувала ефективність класифікатора, враховуючи кількість категорій. Як одну із таких характеристик розглянемо покращення (*improvement*), яку визначимо як відношення повноти до частоти випадкового визначення правильної категорії

$$Impr^{tclass} = \frac{Rc_{mean}^{tclass}}{P_{ctg}^{prb}}. \quad (21)$$

Покращення $Impr^{tclass}$ характеризує ефективність класифікатора порівняно із випадковим вибором категорії. Для простоти розрахунків вважаємо, що ймовірність правильного випадкового вибору категорії документа є однакою для всіх категорій

$$P_{ctg}^{prb} = \frac{1}{|Categories|}. \quad (22)$$

Тоді отримаємо

$$Impr^{tclass} = |Categories| \cdot Rc_{mean}^{tclass}. \quad (23)$$

Експериментальна частина

Для експериментального вивчення класифікації текстових документів у просторі семантичних полів ми вибрали стандартизовану текстову базу повідомлень 20NewsGroups [http://qwone.com/~jason/20Newsgroups/]. Ця база містить близько 20000 повідомлень, які рівномірно розподілені по 20 групах новин. Для формування семантичного простору вибрано лексеми, згруповані за семантичними полями іменників та дієслів у семантичній мережі WordNet [5]. Семантичні полі у мережі WordNet (http://wordnet.princeton.edu) представлені лексикографічними файлами. У наших дослідженнях ми використали семантичні поля іменників та дієслів. Семантичні поля іменників складаються із 26 лексикографічних файлів, із яких ми відібрали 54464 лексеми. Семантичні поля дієслів містять 15 лексикографічних файлів, у які ми відібрали 9097 лексем. У семантичні поля також ввійшли похідні форми лексем. За допомогою розробленого програмного забезпечення здійснена початкова обробка текстового масиву, вилучено допоміжні символи та текстові елементи, які не несуть семантичної інформації. Для кожного документа та вибірки в цілому, обраховано частотні словники, на основі яких розраховано матрицю M_{sd} (5) типу документ-частота_семантичного_поля. На основі цієї матриці ми досліджували два типи класифікаторів – наївний байесівський класифікатор та класифікатор за найближчими сусідами. У дослідженнях розглядалися різні параметри класифікації, зокрема, досліджувалась класифікація, у якій навчальна та тестова вибірки документів збігалися та випадки, коли вони були різні. Також досліджувався випадок об'єднаної множини семантичних полів та випадки класифікації лише за полями іменників чи полями дієслів. Досліджувались випадки класифікації текстової вибірки із вилученими високочастотними лексемами, які разом становлять 50% текстового наповнення масиву документів. Також проаналізовані випадки класифікації документів окремих категорій, які в результаті класифікаційного аналізу виявились найбільш категоріально виразними у просторі семантичних полів. Розглянемо основні отримані результати. Точність та повнота для категорій у випадку байесівської класифікації при збігу навчальної та тестової вибірки об'ємом 20000 повідомлень множин для 40 семантичних полів та 20 категорій наведена на рис. 1. Такі ж розрахунки для класифікатора за найближчими k сусідами при $k=3$ наведено на рис. 2. На цих рисунках відображені категоріальні розподіли точності та повноти. У випадку байесівського класифікатора для деяких категорій показники точності та повноти є задовільними, а для деяких є низькими. Метод класифікації за найближчими сусідами показує високу ефективність. При класифікації тестового документа, який одночасно належить навчальній вибірці відбувається попадання цього документу у його індивідуальний семантичний окіл. При розгляді семантичного підпростору, утвореного лише семантичними полями іменників чи дієслів, точність розрахунків зменшується. При збільшенні кількості сусідів втрачається точність класифікатора. Це свідчить про високу щільність текстових документів різних категорій у семантичному просторі і впливає на точність визначення категорії документа. При $k=1$ точність та повнота зростають. Це означає, що кожний документ у семантичному просторі заходить у окремому місці і не відбувається перекриття семантичних околів для різних документів.

Результати макроусереднення показників класифікації по групах новин при різних умовах наведені у таблиці. Низькі значення точності та повноти байесівської класифікації повідомлень для деяких груп новин можуть бути зумовлені тим, що хоч повідомлення і належать деякій групі, однак можуть містити нейтральну текстову інформацію щодо основної тематики групи.

Розглянемо випадок розділення текстової вибірки на навчальну та тестову так, щоб документи із цих двох вибірок не збігались. Документи у вибірці розміщувались у випадковому порядку. Слід відмітити, що для одних і тих самих навчальних та тестових вибірок та категорій спостерігаються різні значення точності та повноти при використанні байесівського класифікатора та класифікатора за найближчими сусідами. Метод найближчих сусідів дає високі результати

точності та повноти класифікації у випадку коли тестова вибірка є підмножиною навчальної вибірки. Коли ці вибірки є відмінні, точність класифікатора за найближчими сусідами є суттєво меншою, однак є більшою за точність байесівського класифікатора у цих самих умовах. Невисокі значення точності та повноти при відмінності навчальної та тестової вибірки можуть пояснюватись специфікою текстової бази груп новин.

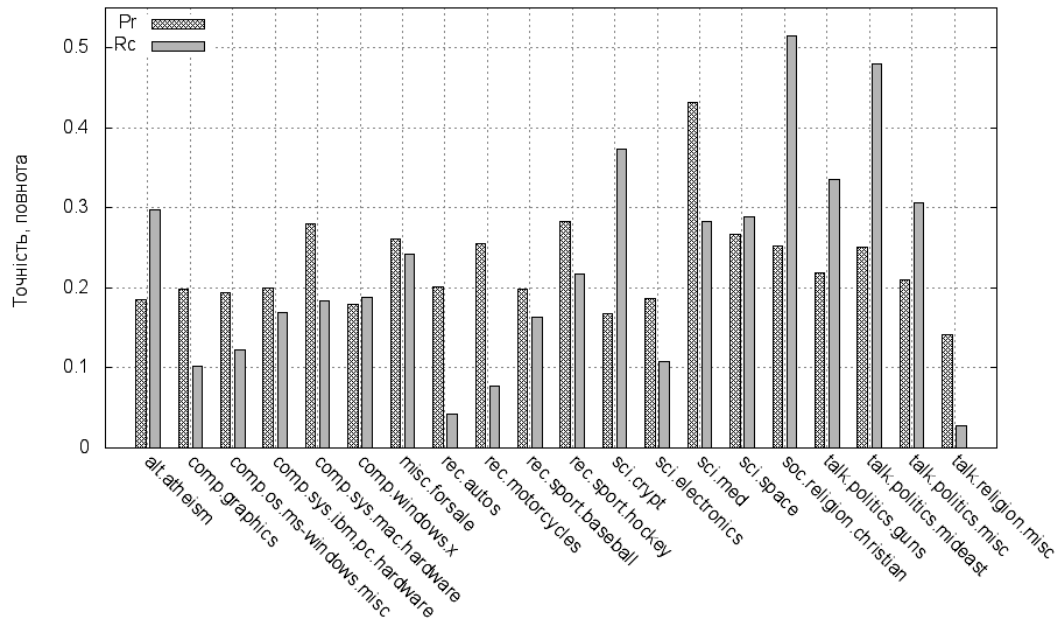


Рис. 1. Точність та повнота байесівського класифікатора для груп новин

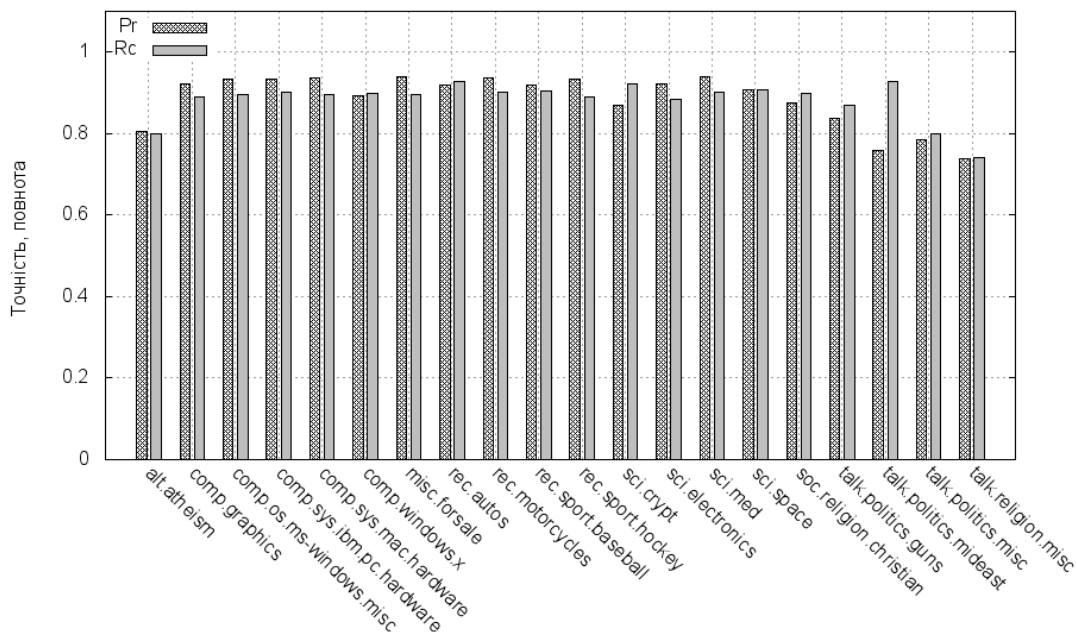


Рис. 2. Точність та повнота класифікатора за найближчими k сусідами (k=3) для груп новин

Насамперед це малий об'єм текстової інформації у кожному повідомленні, яка у середньому становить близько 1Кбайт. Це впливає на якість представлення текстових повідомлень у просторі семантичних полів. Наявність повідомлення у групі новин відображає членство у цій групі автора повідомлення і не завжди є релевантим до тематики, тому може не належати області у просторі семантичних полів, яка відображає тематику групи. Для випадку, коли навчальна та тестова вибірки

збігаються, тоді кожний документ є представлений у деякому семантичному околі простору і при $k=1$ і цей документ попадає у свою ж область, що пояснює високу точність класифікатора kNN у цих умовах. Очевидно, що широковживані лексеми несуть мінімальну семантичну інформацію і часто виконують допоміжну роль у тексті. Розглянемо частотний словник лексем, упорядкований за частотою лексем, які сумарно становлять 50 % лексемного наповнення текстів. Для аналізованої вибірки такий список містить 100 лексем. На основі цього списку утворимо лексемний фільтр, за допомогою якого виділимо ці лексеми із текстової вибірки. Для відфільтрованого текстового масиву побудуємо матрицю семантичних векторів документів і застосуємо класифікатор за найближчими сусідами. Отримано усереднені значення точності та повноти для розділених навчальних та тестових вибірок ($|D_{set}^{training}|=15000, |D_{set}^{test}|=1/3|D_{set}^{training}|$):

$$Pr_{mean}^{tclass} = 0.31, Rc_{mean}^{tclass} = 0.30, Impr^{tclass} = 6.12.$$

Показники класифікаторів при різних параметрах класифікації

Класифікатори	Pr_{mean}^{tclass}	Pr_{std}^{tclass}	Rc_{mean}^{tclass}	Rc_{std}^{tclass}	$Impr^{tclass}$
Умови класифікації: $D_{set}^{test} = D_{set}^{training}, S = \{s_k / (\forall w_i \in s_k : w_i \in Nouns) \vee (\forall w_i \in s_k : w_i \in Verbs)\}$					
NB	0.2283	0.2262	0.0621	0.1349	4.5237
kNN (k=1)	0.9731	0.0494	0.9729	0.0482	19.4589
kNN (k=3)	0.8855	0.0653	0.8829	0.0477	17.6586
kNN (k=10)	0.5500	0.0762	0.5335	0.0856	10.6696
Умови класифікації: $D_{set}^{test} = D_{set}^{training}, S = \{s_k / (\forall w_i \in s_k : w_i \in Nouns)\}$					
NB	0.1863	0.1765	0.0701	0.1281	3.5295
kNN (k=1)	0.4957	0.0628	0.4870	0.0752	9.7395
Умови класифікації: $D_{set}^{test} = D_{set}^{training}, S = \{s_k / (\forall w_i \in s_k : w_i \in Verbs)\}$					
NB	0.1673	0.1473	0.0608	0.1337	2.9464
kNN (k=1)	0.9713	0.0496	0.9710	0.0470	19.4209
Умови класифікації: $D_{set}^{test} \notin D_{set}^{training}, D_{set}^{training} =15000, D_{set}^{test} =5000$ $S = \{s_k / (\forall w_i \in s_k : w_i \in Nouns) \vee (\forall w_i \in s_k : w_i \in Verbs)\}$					
NB	0.2177	0.0718	0.2168	0.1374	4.3497
kNN (k=1)	0.2877	0.0589	0.2840	0.0791	5.6663
kNN (k=3)	0.2883	0.0606	0.2853	0.0875	5.6863
Умови класифікації: $D_{set}^{test} \notin D_{set}^{training}, D_{set}^{training} =5000, D_{set}^{test} \approx 5000,$ $S = \{s_k / (\forall w_i \in s_k : w_i \in Nouns) \vee (\forall w_i \in s_k : w_i \in Verbs)\}$					
NB	0.1957	0.0592	0.2007	0.1269	4.0696
kNN (k=1)	0.2154	0.0484	0.2131	0.0752	4.2577
kNN (k=3)	0.2268	0.0566	0.2282	0.1012	4.5618

Як впливає із отриманих результатів вилучення широковживаних лексем, які становлять 50 % наповнення текстових масивів покращує точність та повноту класифікатора і істотно зменшує обсяг обчислень у класифікаційному аналізі внаслідок зменшення текстової вибірки наполовину лексем.

При формуванні класифікаційного простору на основі підібраних для даної вибірки ключових слів можна досягнути значно кращих результатів. Однак формування такої вибірки потребує

значних експертних ресурсів, крім того розмірність класифікаційного простору значно зростає. Така вибірка буде спеціалізовано лише для даного типу тематики текстової колекції. В той же час байесівська класифікація у просторі семантичних полів є універсальною для всіх типів текстових колекцій і не потребує додаткових налаштувань. Байесівський класифікатор характеризується великим значенням середньоквадратичного відхилення по класифікаційним категоріям. Це свідчить про те, що для деяких категорій він дає задовільні результати.

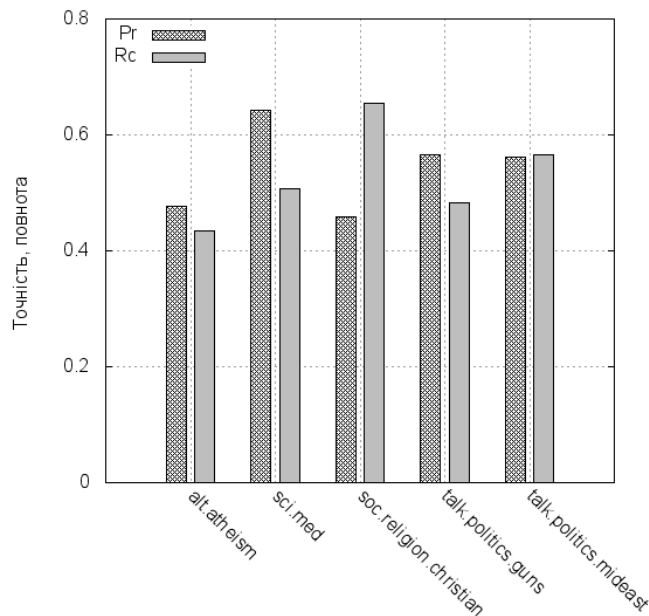


Рис. 3. Показники байесівської класифікації для вибраних семантично виразних категорій

Не зважаючи на низькі значення макроусередненої точності та повноти байесівського класифікатора при великій кількості категорій для аналізованого типу текстових документів цей метод може бути використаний у попередньому опрацюванні колекції текстових документів, зокрема тієї частки категорій, на яких цей класифікатор дає задовільні результати. Очевидно, що байесівський класифікатор може бути використаний для деякої підмножини груп новин, повідомлення яких є семантично виразними по відношенню до інших категорій. На рис. 3 зображена точність та повнота байесівського класифікатора для підмножини семантично виразних груп новин у випадку роздільної навчальної та тестової вибірки. Макроусереднені показники мають такі значення $Pr_{mean}^{tclass} = 0.5407$, $Rc_{mean}^{tclass} = 0.5290$. Отримані результати свідчать про те, що у множині груп новин існує підгрупа новин, повідомлення якої можуть бути ефективно класифіковані байесівським класифікатором.

Висновки

Проаналізовано можливість використання наївного баєсівського класифікатора (NB) та класифікатора за найближчими сусідами (kNN) у класифікаційному семантичному аналізі повідомлень груп новин. Текстові повідомлення розглянуті у векторному просторі, базис якого утворюють частотні характеристики семантичних полів іменників та дієслів. Розмірність такого базису є істотно меншою ніж у випадку широкоживаної класифікації текстів за ключовими словами. Виявлено високу ефективність kNN класифікатора у випадку збігу навчальної та тестової вибірки. Виявлено підмножину груп новин на яких NB класифікатор дає задовільні результати. Категоріальні розподіли точності та повноти можуть істотно відрізнятись для різних класифікаторів при одних і тих самих навчальних та тестових вибірках. Отримані результати свідчать про ефективність реалізації NB та kNN класифікації у просторі семантичних полів і відображають сукупність характеристик розглянутих класифікаторів та текстової вибірки заданого типу повідомлень груп новин. Для іншого типу вибірок кількісні характеристики

семантичних класифікаторів можуть істотно відрізнятись. У наступних дослідженнях ми плануємо розглянути класифікацію текстів у семантичному просторі для вибірок авторських текстів у задачах аналізу авторського ідеолекта, а також інші стандартизовані текстові вибірки.

1. Павлишенко Б. М. Використання концепції семантичного поля у векторній моделі текстових документів // *Східно-Європейський журнал передових технологій*. – 2011. – № 6/2(54). – С. 7–11.
2. Павлишенко Б. М. Ієрархічна кластеризація текстових документів у векторному просторі семантичних полів // *Електроніка та інформаційні технології*. – 2011. – Вип. 1. – С. 212–222.
3. Павлишенко Б. М. Сингулярна декомпозиція матриці семантичних ознак в алгоритмі ієрархічної кластеризації текстових масивів // *Математичні машини і системи*. – 2012. – № 1. – С. 69–76.
4. Pantel Patrick, Turney Peter D. *From Frequency to Meaning: Vector Space Models of Semantics* // *Journal of Artificial Intelligence Research*. – 2010. – vol.37. – pp.141-188.
5. Fellbaum C. *WordNet. An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998, 432p.
6. Gliozzo Alfio, Strapparava Carlo. *Semantic Domains in Computational Linguistics*. Springer, 2009 – 132 p.
7. Брасеян А.А., Куприянов М.С., Холод И.И., Тесс М.Д., Елизаров С.И. *Анализ данных и процессов: учеб. Пособие*. – СПб.: БХВ–Петербург, 2009. – 512с.:ил.
8. Sebastiani F. *Machine Learning in Automated Text Categorization* // *ACM Computing Surveys*. – 2002. – Vol. 34, № 1. – pp. 1–47.
9. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008. – 496 p.

УДК 004.421.2:517.443

І. Процько

Національний університет “Львівська політехніка”,
кафедра систем автоматизованого проектування

СИНТЕЗ ТА ОБЧИСЛЕННЯ ОСНОВНИХ ТИПІВ ДПХ НА ОСНОВІ ЦИКЛІЧНИХ ЗГОРТОК

© Процько І., 2012

Розглянуто підхід до ефективного обчислення основних чотирьох типів дискретного перетворення Хартлі (ДПХ) на основі циклічних згорток. Параметри твірної масиви базисної квадратної матриці використано для синтезу алгоритму.

Ключові слова: дискретні перетворення Хартлі, твірний масив, синтез алгоритму, циклічна згортка.

The general method of efficient computation four types discrete Hartley transform using of circular convolutions is considered. The parameters of hash array of basis square matrix for algorithm synthesis are used.

Key words: discrete Hartley transforms, hash array, algorithm synthesis, cyclic convolution.

Вступ

Для опису даних в їх спектральному гармонічному образі застосовуються високоефективні дискретні перетворення класу Фур'є. У більшості застосувань опрацьовують інформацію над послідовностями дійсних даних. Тому обробка ДПФ над дійсними даними є інформаційно надлишкова, а саме дійсна частина ДПФ є парною функцією та уявна непарною [1]. Одну з

альтернатив ДПФ запропонував американський вчений Р.В.Л. Хартлі в 1942 р. для перетворення над дійсними числами [2].

$$H(u) = \int f(t) \text{cas}(2\pi ut) dt \quad (1)$$

Було вказано на можливість виконання строгих взаємних інтегральних перетворень, в яких всі операції виконувались з дійсними числами без використання комплексних величин. Особливістю цього перетворення являється дійсна базисна функція, що в дискретній формі має вигляд

$$\text{cas}(2\pi kn/N) = (\cos(2\pi kn/N) + \sin(2\pi kn/N)). \quad (2)$$

Подальший значний внесок в розвиток і популяризацію цього напрямку опрацювання сигналів зробив Р. Брейсуелл, починаючи з 1983 р. [3], який розробив основи теорії неперервного і дискретного перетворення Хартлі, а також один з варіантів його швидкого перетворення (ШПХ) у 1984 році, що часто називають алгоритмом Хартлі-Брейсуелла [4] для обсягів $N = 2^n$ в $O(N \log_2(N))$ операцій.

Спектр Хартлі досліджується та застосовується в багатьох прикладних задачах. Відзначимо, що дійсна та уявна частини перетворення Фур'є, хоч і мають певний фізичний зміст, саме як амплітудний розподіл косинусних і синусних коливань, в якісному аналізі сигналів також застосовуються достатньо рідко. Практичніше значення для аналізу мають модуль та фаза спектра (амплітудно-частотна і фазово-частотна характеристика) і спектр густини потужності сигналу. Найбільш перспективними застосуваннями ШПХ є обчислення згорток, визначення спектральної густини та певні види цифрового опрацювання сигналів.

Подальший інтенсивний розвиток інформаційних технологій задає вищі вимоги перед ДПХ та їх алгоритмічними, програмно-апаратними засобами з швидкодії та розвитку функціональних й специфічних можливостей перетворень.

Аналіз літературних джерел

Більше трьох десятиліть досліджували ефективні обчислення одно-, двовимірних ДПХ, що назвали швидкими перетвореннями Хартлі (ШПХ). Отримано значну кількість публікацій, присвячених ефективному обчисленню [5, 6].

Звернення до перетворень Хартлі було обумовлено ситуацією, що склалась у деяких методах опрацювання інформації, де опрацювати дані бажано в області дійсних чисел. На відміну від перетворення Фур'є, що відображає дійсні функції в комплексну область і несиметричного по комплексній змінній, перетворення Хартлі здійснює відображення тільки в дійсній області.

Серед багатоваріантності ефективних обчислень для ШПХ розділяють на алгоритми з основою два, розчепленою основою, змішаною основою, непарного обсягу, складеного обсягу і алгоритм простих множників.

Для синтезу ефективних алгоритмів ДПХ використовують підходи:

- 1) прямої факторизації матриці ДХП;
- 2) непрямого обчислення через швидке перетворення Фур'є або через інші дискретні тригонометричні перетворення;
- 3) алгоритми, що базуються на алгебраїчній теорії складності.

Роботи з швидких ДПХ узагальнюються та систематизуються і завершальним кроком в цьому напрямі є теорія, що забезпечить автоматичну генерацію швидких алгоритмів [7].

Постановка проблеми

Обчислення ДПХ та ІДПХ (прямого та інверсного) належить до однієї з найбільш ємких і тривалих процедур в інформаційних технологіях, хоч пряме та інверсне перетворення Хартлі взаємно симетричні. Тобто, ця процедура в найбільшому ступені потребує вдосконалень, що дозволять пришвидшити роботу програмного та апаратного забезпечення і відповідно саме перетворення.

Одним з напрямків ефективних алгоритмів є можливість обчислення ДПХ через циклічні згортки. Цей напрям ефективного обчислення використовує наявність алгоритмів швидкої згортки [8].

Особливо поява публікацій, пов'язаних з обчисленням поширеного в застосуваннях ДПХ через циклічні згортки, належить до початку 90-х років ХХ ст. [9, 10]

Більшість досліджень використовують перехід від обчислення дискретного перетворення до циклічних згорток застосовуючи переіндексацію для простого обсягу за Рейдером [8] або розклад складеного обсягу перетворення на прості множники за Агарвалом і Кулі [8], або комбінуючи ці підходи. У роботі [11] розглянуто приведення першого типу ДПХ до циклічних структур. Використання способу обчислення на основі згорток має свої особливості для визначення узагальненого ДПХ різних типів, що потребують подальшого дослідження та аналізу.

Чотири типи ДПХ I-IV

Дискретне перетворення Хартлі відображає вхідні дані в лінійну комбінацію зважених косинусних функцій. Існує чотири типи дискретного перетворення Хартлі, що розглянуті в роботі [12]. Ці перетворення є подальшим вдосконаленням ДПФ для дійсних вхідних даних.

Це перетворення обсягу N має N вхідних значень $x(n)$ ($n=0,1,\dots,N-1$) та $X(k)$ вихідних ($n=0,1,\dots,N-1$), що визначаються за формулою

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos[(n+n_0)(k+k_0) \frac{2p}{N}], \quad k = 0,1,\dots,N-1 \quad (3)$$

Зворотне дискретного перетворення Хартлі (ЗДПХ) визначається за формулою

$$x(n) = \sum_{k=0}^{N-1} X(k) \cos[(n+n_0)(k+k_0) \frac{2p}{N}], \quad n = 0,1,\dots,N-1 \quad (4)$$

де n_0, k_0 – відповідні зміщення. Отже, на відміну ДПФ обчислення прямого і зворотного ДПХ здійснюється за формулами (3), (4) вид, яких збігається з точністю до множника $1/N$.

Для основних чотирьох типів ДПХ зміщення k_0, n_0 в загальному виразі (3), (4) можуть приймати значення $0, \frac{1}{2}$ (табл. 1).

Таблиця 1

$k_0, n_0,$	0, 0	0, $\frac{1}{2}$	$\frac{1}{2}, 0$	$\frac{1}{2}, \frac{1}{2}$
Тип ДПХ	ДПХ-I	ДПХ-II	ДПХ-III	ДПХ-IV

Інформаційні технології широко застосовують ДПХ I-IV-го видів і представлені виразами відповідного виду. ДПХ також розглядають як альтернативу для обчислення дискретних косинусних перетворень.

Ефективне обчислення ДПХ I-IV на базі циклічних згорток

ДПХ-I відповідає класичному дискретному перетворенню, яке подав Р. Брейсуелл у роботі [13]. Пряме та зворотне одновимірне ДПХ-I визначається співвідношеннями:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos [(nk) 2p/N], \quad k = 0,1,\dots,N-1 \quad (5)$$

$$x(n) = \sum_{k=0}^{N-1} X(k) \cos [nk 2p/N], \quad n = 0,1,\dots,N-1 \quad (6)$$

ДПХ-II

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos [(2n+1)k p/N], \quad k = 0,1,\dots,N-1 \quad (7)$$

$$x(n) = \sum_{k=0}^{N-1} X(k) \cos [(2n+1)k p/N], \quad n = 0,1,\dots,N-1 \quad (8)$$

ДПХ-III

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos [(2k+1)n p/N], \quad k = 0,1,\dots,N-1 \quad (9)$$

$$x(n) = \sum_{k=0}^{N-1} X(k) \text{cas} [(2k+1)n p/N], \quad n = 0, 1, \dots, N-1 \quad (10)$$

ДПХ-IV

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \text{cas} [(2k+1)(2n+1)p/2N], \quad k = 0, 1, \dots, N-1 \quad (11)$$

$$x(n) = \sum_{k=0}^{N-1} X(k) \text{cas} [(2k+1)(2n+1)p/2N], \quad n = 0, 1, \dots, N-1 \quad (12)$$

Проаналізуємо структуру матриці базису для видів ДПХ за аргументами компонентів $h_{k,n}$, що дорівнюють відповідно

для ДПХ-I

$$c_{k,n} = kn \, 2\pi/N, \quad (k,n=0,1,\dots,N); \quad (13)$$

для ДПХ-II

$$c_{k,n} = k(2n+1) \pi/N, \quad (k,n=0,1,\dots,N-1); \quad (14)$$

для ДПХ-III

$$c_{k,n} = (2k+1)n \pi/N, \quad (k,n=0,1,\dots,N-1); \quad (15)$$

для ДХП-IV

$$c_{k,n} = (2k+1)(2n+1) \pi/2N, \quad (k,n=0,1,\dots,N-1). \quad (16)$$

Базисна функція касинус періодична (2π), асиметрична відносно осі (π) на проміжку $(0, 2\pi)$, та має осі симетрії $\pi/4$ на проміжку $(0, \pi/2)$; вісь $3\pi/4$ на проміжку $(0, 3\pi/2)$; вісь $5\pi/4$ на проміжку $(\pi/2, 2\pi)$, вісь $7\pi/4$ на проміжку $(3\pi/2, 2\pi)$. Дані асиметричності та симетричності залежать від обсягу та типу перетворення і в загальному наведені в табл. 2.

Таблиця 2

Перетворення	періодичні	асиметричні	симетричні
ДПХ-I	відносно N вибірок	відносно N/2 вибірок	відносно N/8 вибірок
ДПХ-II	відносно 2N вибірок	відносно N вибірок	відносно N/4 вибірок
ДПХ-III	відносно 2N вибірок	відносно N вибірок	відносно N/4 вибірок
ДПХ-IV	відносно 4N вибірок	відносно 2N вибірок	відносно N/2 вибірок

Матриця аргументів типів перетворень Хартлі за періодичністю буде

$$H_a^I(k,n) = [(k n) \text{ mod } (N)], \quad (17)$$

$$H_a^{II}(k,n) = [k(2n+1) \text{ mod } (2N)], \quad (18)$$

$$H_a^{III}(k,n) = [(2k+1)n \text{ mod } (2N)] \quad (19)$$

$$H_a^{IV}(k,n) = [(2k+1)(2n+1) \text{ mod } (4N)], \quad (20)$$

На основі підстановки з рядків даних матриць формується твірний масив $P(n)$. На основі властивості асиметричності та часткової симетричності базисної функції косинус-елементи матриці аргументів можна спростити. Спрощені елементи матриці аргументів визначаються послідовним виконанням обчислень:

для ДПХ-I з N парним

$$\underline{h}_{k,n} = [(h_{k,n}) \text{ mod } N] - N/2, \quad \text{якщо } [(h_{k,n}) \text{ mod } N] > N/2; \quad (21)$$

для N кратне 8

$$\underline{h}_{k,n} = \{N/4 - [(h_{k,n}) \text{ mod } N - N/2]\}, \quad \text{якщо } N/8 < \{(h_{k,n}) \text{ mod } N - N/2\} < N/4, \quad (22)$$

$$\underline{h}_{k,n} = N/4 + \{N/2 - [(h_{k,n}) \text{ mod } N - N/2]\}, \quad \text{якщо } 3N/8 < \{(h_{k,n}) \text{ mod } N - N/2\} < N/2, \quad (23)$$

інакше

$$\underline{h}_{k,n} = h_{k,n}.$$

Приклад заміни значень аргументів на основі симетрії для перетворення обсягом $N=24$

$$\begin{array}{cccccccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ & & & & & & 2 & 1 & & & & & 8 & 7 & & & & & 1 & 2 & 3 & 2 & 1 & 6 & 7 & 8 & 9 & 8 & 7 \end{array}$$

Властивості симетрії та періодичності базису перетворення ДПХ приводять до ефективнішого представлення меншими значеннями елементів твірних підмасивів $P'(n)$ з доповненнями відповідних підмасивів знаків $Z(n)$. Підматриці знаків $Z(n)$ містять значення елементів $+1, -1, 0$.

Твірний масив $P(n)$ перетворення визначає особливість структури базисної матриці ДПХ приведенної до циклічних підматриць. Тому можна виділити параметри, що характеризують $P(n)$ і, відповідно, видозмінену базисну матрицю:

- k – кількість підмасивів у твірному масиві $P(n)=P_1(n_1)P_2(n_2)\dots P_k(n_k)$;
- кількість елементів кожного твірного підмасиву (L_1, L_2, \dots, L_k) ;
- перший елемент кожного твірного підмасиву n_{i1} , $i=1(1)k$.

Наступним кроком синтезу в алгоритмі обчислення ДПХ є визначення однотипових циклічних підматриць. Тобто знаходження однакових та квазіоднакових підматриць (мають однакові індекси, але протилежні значення знаків) на основі значень параметрів твірного масиву $P(n)$ і спрощеного твірного масиву індексів $P'(n)$, що доповнюється масивом знаків $Z(n)$. Значення параметрів спрощеного твірного масиву

$$P'(n)=P'(n_1)P'(n_2)\dots P'(n_k), \quad Z(n) = Z(n_1) Z(n_2)\dots Z(n_k) \quad (37)$$

для даного обсягу N і виду перетворення ДПХ визначаються відповідно за виразами (21)–(31), масив знаків $Z(n)$ – за виразами (32)–(34).

Для визначення однотипових циклічних підматриць значення елементів матриці $H_a(k,n)$ можуть бути задані попередньо, однак великий перебір всіх елементів потребує значних об'ємів пам'яті для їх збереження і відповідних часових затрат. Ефективніший шлях дасть можливість визначення тільки перших елементів підматриць в процесі аналізу структури базису за координатами розміщення підматриць. Тобто обчислюються за координатами рядка і стовпця значення перших елементів підматриці і аналізуються між собою.

Відповідність координат (i, j) елементам твірного масиву $P(n_i)$ та $P'(n_i)$:

$$\begin{array}{l} (i \setminus j) \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad \dots \quad n \\ P(n_i) \quad (n_{11}, n_{12}, n_{13}, \dots, n_{1L_1}, n_{21}, n_{22}, n_{23}, \dots, n_{2L_2}, \dots, n_{kL_1}, n_{kL_2}, \dots, n_{kL_k}) \\ P'(n_i) \quad (n'_{11}, n'_{12}, n'_{13}, \dots, n'_{1L_1}, n'_{21}, n'_{22}, n'_{23}, \dots, n'_{2L_2}, \dots, n'_{kL_1}, n'_{kL_2}, \dots, n'_{kL_k}). \end{array}$$

Координати перших елементів підматриць визначаються за $(i+L_i), (j+L_i)$, де L_i – вибирається за належності до твірного підмасиву значення перших елементів підматриць у матричній структурі. Перші елементи обчислюються за відповідними визначеними координатами (i, j) і відповідатимуть елементу $P'(n)$ спрощеного твірного масиву зі знаком, відповідно до типу перетворення ДПХ (табл. 3).

Таблиця 3

$(i+L_i, j+L_i) - n_{ij}$ (координати рядок/стовпець) – знак і значення першого елемента підматриць s_{ij} ;				
$(1, 1) - z_{\underline{h}_{ij}}$	$(1, 1+L_1) - z_{\underline{h}_{ij}}$...
				$(1, 1+L_1+L_2+ \dots +L_k) - z_{\underline{h}_{ij}}$
				$(1+L_k, 1+L_1+L_2 + \dots +L_k) - z_{\underline{h}_{ij}}$
$(1+L_1, 1) - z_{\underline{h}_{ij}}$	$(1+L_1, 1+L_1) - z_{\underline{h}_{ij}}$...
				$(1+2L_k, 1+L_1+L_2+ \dots +L_k) - z_{\underline{h}_{ij}}$
				$(1+3L_k, 1+L_1+L_2+ \dots +L_k) - z_{\underline{h}_{ij}}$
...
				...
$(1+L_1+L_2+ \dots +L_k, 1) - z_{\underline{h}_{ij}}$	$(1+L_1+L_2+ \dots +L_k, 1+L_k) - z_{\underline{h}_{ij}}$	$(1+L_1+L_2+ \dots +L_k, 1+2L_k) - z_{\underline{h}_{ij}}$	$(1+L_1+L_2+ \dots +L_k, 1+3L_k) - z_{\underline{h}_{ij}}$	$(1+L_1+L_2+ \dots +L_k, 1+3L_k) - z_{\underline{h}_{ij}}$
				...

Визначення однотипових циклічних підматриць виконується через відбір за координатами перших елементів однакових підматриць по горизонталі. Для координат рядків, кратних *простим*

множникам розкладу обсягу N в структурі, виконуємо об'єднання вхідних даних по горизонталі виконанням поелементних додавань вхідних значень.

Ці значення будуть використовуватись для обчислення циклічних згорток під час аналізу по вертикалі. Тобто, відбір за координатами перших елементів однакових підматриць по вертикалі і обчислення циклічних згорток з вхідними значеннями. Під час аналізу підматриць по вертикалі для координат стовпців кратних простим множникам розкладу обсягу N в структурі виконуємо одноразове обчислення циклічних згорток. За наявності решти незадіяних підматриць виконуються циклічні згортки за своїми параметрами на основі визначених координат.

Об'єднання результатів згорток виконується по горизонталі на основі відповідних координат перших елементів підматриць. Одержані вихідні значення перетворення відповідають порядку, відповідному до значень твірного масиву P(n).

Приклади синтезу ефективного обчислення ДПХ

Розглянемо приклад синтезу алгоритму для організації обчислення ДПХ-III обсягу N=11 на основі циклічних згорток. Враховуючи особливість базисної функції $\cos((2k+1)n \pi/N)$, твірний масив P(n) за стовпцями вдвічі більший від твірного масиву за рядками, бо складається із парних та непарних значень (n=1(1)21) аргументів на інтервалі одного періоду без врахування x(0), x(11)

$$P(21) = (1,3,9,5,15) (13,17,7,21,19) (2,6,18,10,8) (12,14,20,16,4) ;$$

$$P'(21) = (1,3,9,5,4) (2,6,7,10,8,) (2,6,7,10,8) (1,3,9,5,4) ;$$

$$Z(21) = (+,+,-,+,-) (-,-,+,+,-) (+,+,-,-) (-,-,+,+)(-1).$$

Твірний масив P(n) = P₁(5) P₂(5) P₃(5) P₄(5) за стовпцями містить непарні і парні значення n=1(1)2N-1 аргументів відповідно періоду базису. Твірний масив за рядками складається з підмасивів P(n)=P₁(5) P₂(5), що відповідають N-1 вихідним значенням. Визначення параметрів P(n): k = 4 – кількість підмасивів в твірному масиві; L₁=5, L₂=5, L₃=5, L₄=5 – кількість елементів в підмасивах P(n_i) задають обсяг циклічних згорток.

Вхідні значення перетворення відповідають порядку відповідно до значень твірного масиву за стовпцями без врахування x(0),-x(0)). Тобто переставлена вхідна послідовність повторюється двічі з протилежними знаками: x(1),x(3),x(9),x(5),-x(4),-x(2),-x(6),x(7),-x(10),- x(8), x(2), x(6),- x(7),x(10),x(8),- x(1),- x(3),- x(9),- x(5), x(4), що приведе до збільшення вдвічі 2X[i] вихідних значень за даним алгоритмом.

Визначаються та грукуються коефіцієнти функції $\cos(n_i \varphi)$ за спрощеним твірним масивом P'(21) для касинусних складових базису ДПХ-III з $\varphi=\pi/11$, що беруть участь в операціях згортки для аргументів: (1φ, 3φ, 9φ, 5φ, 4φ), (2φ, 6φ, 7φ, 10φ, 8φ).

Для визначення однотипових циклічних підматриць в структурі базису використовуємо таблицю з координатами (i, j). Табличні координати перших елементів підматриць визначаються за (i+L_i),(j+L_i), де L_i-вибирається за приналежності до твірного підмасиву значень перших елементів підматриць в матричній структурі, які обчислюються за відповідністю координат (i,j) елементам P(n) твірного масиву (n_i x n_j)mod 2N.

Відповідність координат (i, j) елементам твірного масиву P(n_i) та P'(n_i) за стовпцями:

(i, j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	(1,3,9,5,15)	(13,17,7,21,19)	(11)	(2, 6, 18, 10, 8)	(12,14,20,16,4)																

Визначені значення перших елементів підматриць спрощуються за властивістю симетрії, що відповідають симетрії до осі π на проміжку (0, 2π) функції касинуса (обсяг перетворення непарний). У табл. 4 узагальнено структуру базисної матриці аргументів розмірністю (20x11).

Визначення однотипових циклічних підматриць виконується за табл. 4 через аналіз значень за координатами перших елементів однакових підматриць по горизонталі. Для першої горизонталі підматриць виконується об'єднання вхідних даних для двох 5-точкових згорток

$$(1,1) - +1; (1) \text{ та } (1,16) - -1;(12); (1,6) - -2;(13) \text{ та } (1,11) - +2;(2),$$

і далі для наступної горизонталі підматриць.

$$(7,1) - -2; (13) \text{ та } (7,16) - +2;(2); (7,6) - -4;(15) \text{ та } (7,11) - +4;(4),$$

Виконання поелементних додавань вхідних значень будуть використовуватись для обчислення циклічних згорток. Об'єднання результатів згорток виконується по горизонталі на основі відповідних координат перших елементів підматриць. Обчислені вдвічі більші без $2x(0)$ вихідні значення перетворення відповідають порядку відповідно до значень твірного масиву за рядками: $[X(0), X(1), X(4), X(2), X(7), X(5), X(6), X(8), X(3), X(10), X(9)]$.

Таблиця 4

$(i+L_i, j+L_i) - n_{ij}$ (координати рядок/стовпець) – знак і значення першого елемента підматриць			
(1,1) – +1; (1)	(1,6) – -2;(13)	(1,11) – +2;(2)	(1,16) – -1;(12)
(6,1) – -11; (11)		(6,11) – +0;(0)	
(7,1) – -2;(13)	(7,6) – -4;(15)	(7,11) – +4;(4)	(7,16) – +2;(2)

Особливості розподілу циклічних структур в базисній матриці та їх характеристики визначають складність алгоритму обчислення кожного виду ДПХ.

Розглянемо конкретний приклад синтезу складнішого алгоритму для обчислення ДПХ-II обсягу $N=18$. Враховуючи особливість базисної функції $\cos(k(2n+1)\pi/N)$, твірний масив $P(n)$ за стовпцями вдвічі менший від твірного масиву за рядками, бо складається із непарних значень $(2n+1)$ аргументів на інтервалі одного періоду

$$P(18) = (1,5,25,17,13,29) (19,23,7,35,31,11) (3,15) (21,33)(9)(27),$$

$$P'(18) = (1,5,7,17,13,11) (1,5,7,17,13,11) (3,15) (3,15)(9)(9),$$

$$Z(18) = (+, +, -, -, +, -) \quad (-, -, +, +, -, +) (+, -) \quad (-, +) \quad (+)(-).$$

Твірний масив $P(n)$ за рядками містить непарні і парні значення $k=1(1)2N-1$ аргументів на інтервалі одного періоду за винятком $k=0$, тобто без визначення $X[0]$

$$P(18) = (1,5,25,17,13,29)(19,23,7,35,31,11) (3,15) (21,33)(9)(27) (2,10,14,34,26,22) (4,20,28,32,16,8)(6,12)$$

Враховуючи антисиметричність вихідних значень (однакові значення, але з протилежним знаком):

$$X(25) - X(7), X(29) - X(11), X(22) - X(4), X(26) - X(8), X(34) - X(16)$$

для організації обчислень через циклічні згортки використовується твірний масив $P(n)$ за рядками виду:

$$P(18) = (1,5,25,17,13,29) (3,15) (9) (2,10,14,34,26,22) (6,12) .$$

Твірний масив за стовпцями складається з підмасивів $P(18) = P_1(6) P_2(6) P_3(2) P_4(2), P_5(1), P_6(1)$.

Його параметри : $k = 6$ – кількість підмасивів в твірному масиві; $L_1=6, L_2=6, L_3=2, L_4=2, L_5=1, L_6=1$ – кількість елементів в підмасивах $P(n_i)$ задають обсяг циклічних згорток.

Вхідні значення перетворення переставляються в порядку відповідно до значень твірного масиву за стовпцями: $x(0), x(2), x(12), x(8), x(6), x(14), x(9), x(11), x(3), x(17), x(15), x(5), x(1), x(7), x(4), x(13)$.

Визначаються та групуються коефіцієнти функції $\cos(n_i \varphi)$ для касинусних складових базису ДПХ-II з $\varphi = \pi/18$, що беруть участь в операціях згортки для аргументів:

$$(1\varphi, 5\varphi, 7\varphi, 17\varphi, 13\varphi, 11\varphi), (3\varphi, 15\varphi) (9\varphi = \pi/2).$$

Визначаємо однотипові циклічні підматриці за табличними координатами перших елементів підматриць, які обчислюються за відповідністю координат (i, j) елементам $P(n)$ твірного масиву $(n_i \times n_j) \bmod 2N$.

Відповідність координат (i, j) елементам твірного масиву $P(n_i)$ та $P'(n_i)$ за стовпцями:

$$(i, j) \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \quad 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18$$

$$(1,5,25,17,13,29) (19,23,7,35,31,11) (3,15) (21,33)(9)(27),$$

Отримане значення перших елементів підматриць спрощується за властивістю симетрії, що відповідають симетрії до осі π на проміжку $(0, 2\pi)$ функції касинуса (обсяг перетворення парний). У табл. 5 узагальнено базисну матрицю аргументів розмірністю (18×17) без врахування окремих рядків.

Визначення однотипових циклічних підматриць виконується через аналіз за табл. 5 значень за координатами перших елементів однакових підматриць по горизонталі. Для першої горизонталі підматриць виконується об'єднання вхідних даних для 6- та 2-точкових згорток

$$(1,1) - +1; (1) \text{ та } (1,7) - -1;(19); (1,13) - +3;(3) \text{ та } (1,15) - -3;(21)$$

і далі для кожної наступної горизонталі підматриць. Виконання поелементних додавань вхідних значень будуть використовуватись для обчислення циклічних згорток.

Таблиця 5

(i+L_i, j+L_i) – n_{ij} (координати рядок/стовпець) – знак і значення першого елемента підматриць				
(1,1) – +1; (1)	(1,7) – -1;(19)	(1,13) – +3;(3)	(1,15) – -3;(21)	(1, 17) – +9;(9)
(7,1) – +3;(3)	(7,7) – -3;(21)	(7,13) – +3;(9)	(7,15) – -9;(27)	(7,17) – -9;(27)
(9,1) – +9; (9)		(9,13) – -9; (27)		(9,17) – +9; (9)
(10,1) – +2;(4)	(10,7) – +2;(4)	(10,13) – +6;(6)	(10,15) – +6;(6)	(10, 17) – -0;(18)
(16,1) – +6; (6)				(16,17) – -0;(18)
				(17,17) – +0;(0)

Саме виконання циклічних згорток над об'єднаними вхідними даними та згрупованими значеннями косинусної функції використовує ефективні алгоритми швидких згорток [8]. Об'єднання результатів згорток виконується по горизонталі на основі відповідних табличних координат, прив'язуючись до перших елементів підматриць. Обчислені вихідні значення перетворення без X(0) відповідають порядку відповідно до значень твірного масиву за рядками:

X(1),X(5),-X(7),X(17),X(13),-X(11),X(3),X(15),X(9),X(2),X(10),X(14),-X(16),-X(8),-X(4),X(6),X(12).

Розглянемо конкретний приклад синтезу алгоритму для обчислення ДПХ-IV обсягу N=8. Сформоване значення твірного масиву за підстановкою може бути таке:

P(16)=P(n₁)P(n₂) = (1, 3, 9, 27,17,19,25,11)(7,21,31,29,23, 5,15,13),

P'(16)=(1, 3, 9, 11,1,3,9,11)(1, 3, 9, 11,1,3,9,11), Z(n)=(+,+,+,-, -, -, -,+)(+, -, +, +, -, +, -, -).

Його параметри : k =2– кількість підмасивів у твірному масиві; L₁=8, L₂=8, – кількість елементів у підмасивах P(n_i) задають обсяг циклічних згорток; m ≥ k² загальна кількість підматриць: m=4.

Вхідні вісім значень перетворення повторюються з протилежним знаком і переставляються в порядку відповідно до значень твірного масиву:

[x(0),x(1), x(4),- x(5), -x(0),- x(1),- x(4), x(5), x(3),- x(2),- x(7),- x(6),- x(3), x(2), x(7), x(6)].

що приведе до збільшення вдвічі 2X[i] вихідних значень за даним алгоритмом.

Коефіцієнти функції cas(n_i φ) для косинусних складових базису ДПХ-IV з φ=π/2N=π/16 , що беруть участь в операціях згортки для аргументів: (1φ, 3φ, 9φ,11φ).

Для визначення однотипових циклічних підматриць в структурі базису використовуємо табличні координати перших елементів підматриць.

Відповідність координат (i, j) елементам твірного масиву P(n_i) та P'(n_i):

(i, j) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
 (1, 3, 9, 27,17,19,25,11) (7,21,31,29,23, 5,15,13)
 (1, 3, 9, 11, 1, 3, 9, 11) (1, 3, 9, 11, 1, 3, 9,11)

Значення перших елементів підматриць в матричній структурі, які обчислюються за відповідністю координат (i,j) елементам P(n) твірного масиву, дорівнюють (n_i x n_j)mod 4N. Подані в табл. 6 знак і спрощене значення першого елемента n_{ij} підматриці та в дужках (елемент аргументу з P(n_i)), який у разі значення, більшого за 2N, спрощується за виразом () на основі симетричності.

Таблиця 6

(i+L_i, j+L_i) – s n_{ij} (координати рядок/стовпець) – s знак і спрощене значення першого елемента n_{ij} (елемент аргументу з P(n_i)) підматриць	
(1,1) – +1; (1);	(1,9) – +1; (7);
(9,1) – +1; (7);	(9,9) – -1; (17);

Визначення за табл. 6 однотипових циклічних підматриць виконується через відбір за координатами перших елементів однакових підматриць по горизонталі:

(1,1) – +1 (1) та (1,9) – +1 (7) для $P(n_1) = P(n_2) = (1, 3, 9, 11, 1, 3, 9, 11)$,

(9,1) – +1 (7) та (9,9) – +1 (17) для $P(n_1) = P(n_2) = (1, 3, 9, 11, 1, 3, 9, 11)$,

однак ці підматриці містять різний у дужках елемент аргументу $P(n_i)$ 1 і 7; 7 і 17, що відповідає різним підмасивам знаків. Тому ці підматриці не є однотиповими і об'єднання вхідних даних не виконується.

Відбір за координатами перших елементів однотипових підматриць по вертикалі :

(1,1) – +1 (1) та (9,1) – +1 (7) для $P(n_1) = P(n_2) = (1, 3, 9, 11, 1, 3, 9, 11)$,

(1,9) – +1 (7) та (9,9) – +1 (17) для $P(n_1) = P(n_2) = (1, 3, 9, 11, 1, 3, 9, 11)$,

однак ці підматриці містять різний у дужках елемент аргументу $P(n_i)$, що відповідає різним підмасивам знаків. Тому ці підматриці не є однотиповими.

В результаті аналізу структури необхідно виконати чотири 8-точкові симетричні циклічні згортки. Особливістю виконання циклічної згортки є інверсне повторення групи послідовностей як вхідних даних, так і значень функції косинус. Результати згорток визначаємо тільки для чотирьох значень 1, 3, 9, 11 та 7, 5, 15, 13 горизонталей.

Об'єднання результатів згорток виконується по горизонталі на основі координат перших елементів підматриць:

- результати 8-точкової згортки з координатами (1,1) додаються до 8-точкової згортки з (1,9);
- результат з протилежним знаком 8-точкової згортки з координатами (9,9) додається до результату 8-точкової згортки з (1,9). Одержані вихідні значення перетворення відповідають порядку відповідно до значень твірного масиву: $X(1), X(3), X(9), X(11), X(7), X(5), X(15), X(13)$.

Висновки

Ефективне обчислення кожного включає синтез алгоритму для конкретного обсягу та виду ДПХ та виконання алгоритму. При синтезі виконується визначення твірного масиву та його спрощеного представлення, аналіз структури базисної матриці перетворення. На основі відповідного переставлення елементів вхідної послідовності з подальшим використанням швидких алгоритмів циклічної згортки проходить виконання алгоритму. Визначення твірного масиву, за яким відбувається переставлення, не потребує спеціальних обчислень і задається підстановкою на основі двох рядків матриці аргументів базисів ДПХ. Використання твірного масиву $P(n)$ приводить до однотипового підходу проведення організації обчислення різних типів ДПХ послідовностей даних довільного обсягу. Окреме проведення обчислень циклічних згорток, на який структуровано базис типів ДПХ, так і подальше об'єднання одержаних результатів дозволяє ефективно організувати процес обчислення, зменшуючи обчислювальну складність опрацювання інформаційних даних.

1. Опенгейм А., Шафер Р., *Цифровая обработка сигналов*. – М.: Техносфера, 2006. 2. Hartley R.V.L. *A more symmetrical! Fourier analysis applied to transmission problems, Proc.IRE*, vol. 30, pp. 144-150, Mar. 1942. 3. Bracewell, R. N., «The Discrete Hartley Transform», *J. Optical Society of America*, Vol. 73, pp. 1832–1835, Dec. 1983. 4. Bracewell, R. N., «The Fast Hartley Transform», *Proc. IEEE*, Vol. 72, No. 8, pp. 1010–1018, Aug. 1984. 5. Chen G. Bi, Y. *Fast generalized DFT and DHT algorithms, Signal Process.* 65 (1998) 383-390. 6. Guoan Bi, Shou tian Lian: *Fast Algorithms for Generalized Discrete Hartley Transform. Journal of Circuits, Systems, and Computers* 10(1-2): 77-84 (2000). 7. Egner S. and Pueschel M., «Automatic generation of fast discrete signal transforms», *IEEE Transactions on Signal Processing*, Vol. 49, No. 9, September 2001, pp. 1992–2002. 8. Макклеллан Дж., Рейдер Ч. *Применение теории чисел в цифровой обработке сигналов*. – М.: Радио и связь, 1983. 9. Yang D. *Prime Factor Fast Hartley Transform, Elect. Letters*, Jan., vol.26, n.2, pp. 119-121, 1990. 10. Lun D.P.-K., Wan-Chi Siu, *On prime factor mapping for the discrete Hartley Transform. IEEE Trans. On Signal Processing*, vol. 40, N 6, June 1992. 11. Процько І.О., *Ефективне обчислення дискретного перетворення Хартілі на основі*

циклічних згорток // Вісник Нац. ун-ту "Львівська політехніка". Комп'ютерні системи та мережі. – 2010. – № 688. – С. 190–196. 12. Hu N.C., Chang H.I, and Ersoy O.K., "Generalized discrete Hartley transforms," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 2931–2940, 1992. 13. R. N. Bracewell, *The Hartley Transform*, Oxford Univ. Press, Oxford, 1986. 14. Ihor Prots'ko, *The Efficient Computation DHT using Cyclic Convolutions*. // *Proceeding of the XI International Conference CADSM'2011, Poljana, 24–28 february, 2011.* – P. 85–86. 15. Патент 96540 Україна, G06F 17/16 (2006.01), H03M 7/30 (2006.01). Спосіб приведення дискретних гармонічних складових цифрових сигналів до циклічних згорток. /Процько І.О. / Бюл. № 21.

УДК 004.032.26

Є. Бодяньський, О. Тищенко, Д. Копаліані
Харківський національний університет радіоелектроніки,
Проблемна науково-дослідна лабораторія АСУ

ПРОГНОЗУВАННЯ БАГАТОВИМІРНИХ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ НА ОСНОВІ АДАПТИВНОЇ НЕО-ФАЗЗИ-МОДЕЛІ

© Бодяньський Є., Тищенко О., Копаліані Д., 2012

Введено структуру адаптивного нео-фаззи-предиктора та багатовимірного нео-фаззи-нейрона, а також метод навчання останнього. Запропонований алгоритм навчання має підвищену швидкість збіжності та забезпечує фільтруючі властивості. Завдяки введеній нейромережевій архітектурі, вузлами якої є нео-фаззи-нейрони, можна розв'язувати задачі короткострокового прогнозування у реальному часі за умов короткої навчальної вибірки.

Ключові слова: адаптивне прогнозування, нео-фаззи-нейрон, метод навчання, прогнозування, часовий ряд.

The architectures of the adaptive neo-fuzzy predictor and a multidimensional neo-fuzzy neuron are introduced. The proposed learning algorithm increases convergence rate and provides improved filter properties. The short-term prediction tasks may be fulfilled in an online mode with the help of proposed neuro-fuzzy architectures when a data set is short.

Key words: adaptive prediction, neo-fuzzy-neuron, learning algorithm, prediction, time series.

Вступ

Сьогодні штучні нейронні мережі набули широкого розповсюдження для розв'язання великого класу задач опрацювання інформації і, насамперед, для ідентифікації, емуляції, інтелектуального керування, прогнозування часових рядів довільної природи за умов структурної та параметричної невизначеності.

Задача прогнозування багатовимірних часових рядів доволі часто виникає у багатьох технічних, медико-біологічних та інших дослідженнях, де якість прийнятих рішень істотно залежить від точності синтезованих прогнозів. У багатьох реальних задачах часові ряди характеризуються високим рівнем нелінійності та нестационарності своїх параметрів, наявністю аномальних викидів. Зрозуміло, що традиційні методи аналізу часових рядів, засновані на регресійному, кореляційному та інших подібних підходах, що мають на меті апіорну наявність доволі великої вибірки спостережень, є неефективними. Альтернативою традиційним статистичним

методам може слугувати математичний апарат обчислювального інтелекту, а також штучні нейронні мережі [1, 2] та нейро-фаззі-системи [3], завдяки своїм універсальним апроксимувальним властивостям. Водночас з апроксимувальних властивостей зовсім не витікають екстраполуючі, оскільки врахування давньої передісторії для побудови прогнозувальної моделі може погіршити якість прогнозу. У зв'язку з цим під час оброблення нестационарних процесів треба відмовитися від процедур навчання, що базуються на зворотному поширенні помилок (багатосарові перцептрони, рекурентні нейронні мережі, адаптивні нейромережеві системи нечіткого виведення – ANFIS) або методі найменших квадратів (радіально-базисні та функціонально пов'язані нейронні мережі) та скористатися процедурами на основі локальних критеріїв та «короткої» пам'яті типу алгоритма Качмажа-Уїдроу-Хоффа. При цьому використані алгоритми навчання мусять забезпечувати не лише високу швидкодію, але й фільтруючі якості для придушення стохастичної «шумової» компоненти в оброблюваному сигналі. У зв'язку з цим синтез спеціалізованих гібридних систем обчислювального інтелекту для розв'язання задач прогнозування істотно нестационарних часових рядів за умов невизначеності, що забезпечують разом з високою швидкістю навчання і фільтрацію завад, є досить цікавою та перспективною задачею.

Архітектура адаптивного нео-фаззі-предиктора

Архітектуру адаптивного нео-фаззі-предиктора багатовимірного нелінійного істотно нестационарного часового ряду $x(k) = (x_1(k), x_2(k), \mathbf{K}, x_n(k))^T$, де $k = 1, 2, \mathbf{K}$ – поточний дискретний час, наведено на рис. 1.

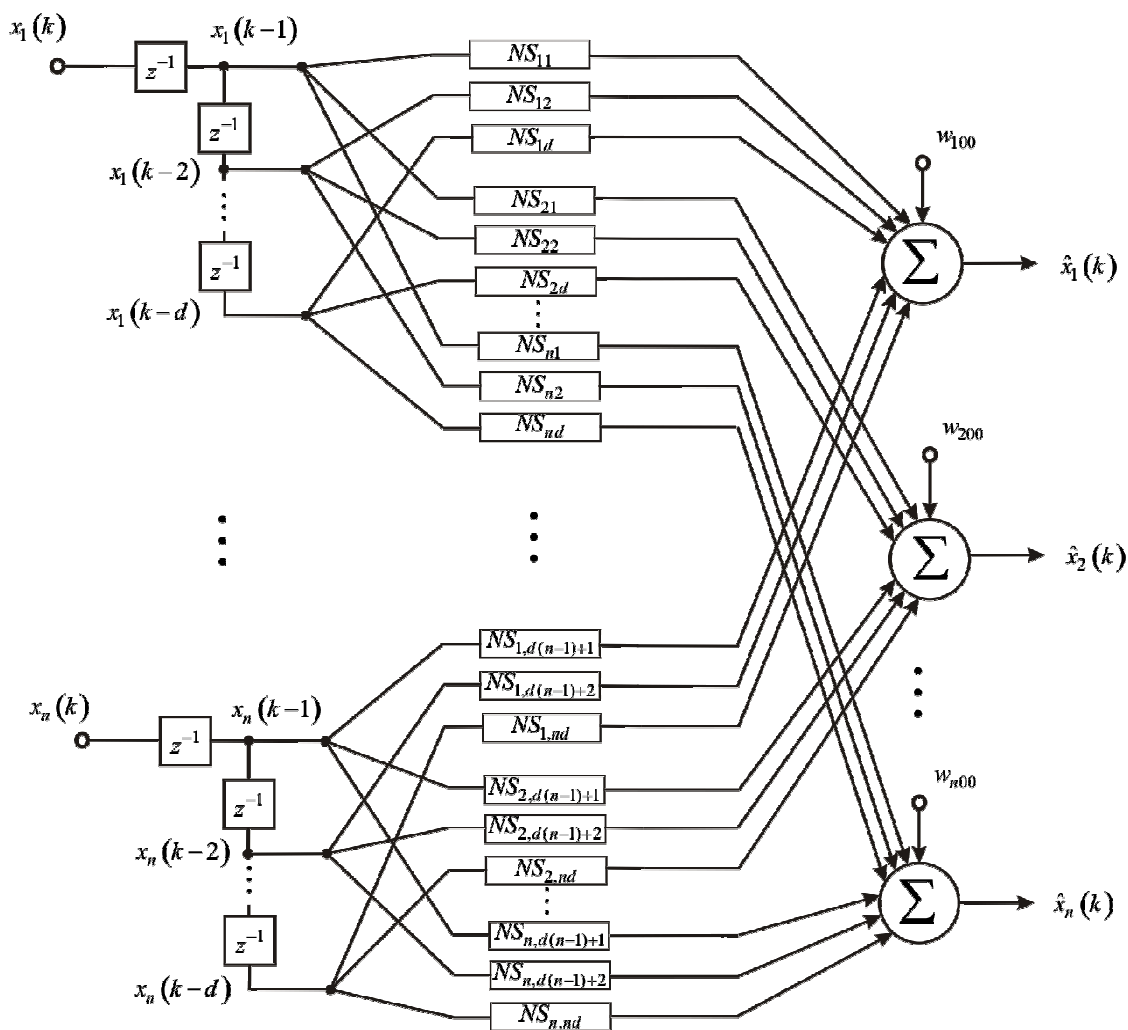


Рис. 1. Архітектура адаптивного нео-фаззі-предиктора

Ця архітектура в принципі збігається з конструкцією введеного раніше адаптивного фільтра-предиктора [4], однак на відміну від останнього в якості вузлів замість адаптивних лінійних асоціаторів містить n паралельно поєднаних нео-фаззі-нейронів [5] з $nd+1$ входами кожний (тут $d \geq 1$ – глибина використовуваної для прогнозування передісторії часового ряду), що є істотно нелінійними елементами. Кожний з нео-фаззі-нейронів замість звичайних синаптичних ваг містить nd нелінійних синапсів NS_{ji} з h синаптичними вагами w_{ijl} кожний, де $j=1,2,\mathbf{K},n;i=1,2,\mathbf{K},nd;l=1,2,\mathbf{K},h$, при цьому архітектура в цілому містить $n(ndh+1)$ налаштованих ваг. З кожним з нео-фаззі-нейронів пов'язаний ланцюжок з d елементів чистого запізнювання z^{-1} , що формує передісторію $x_1(k-1), x_1(k-2), \mathbf{K}, x_1(k-d), x_2(k-1), \mathbf{K}, x_i(k-p), \mathbf{K}, x_n(k-d)$, при цьому синаптичні ваги w_{j00} визначають рівень зміщення на виходах кожного нео-фаззі-нейрона.

Отже, перетворення, що його реалізує ця архітектура, на відміну від лінійного відображення

$$\hat{x}(k) = WX(k), \quad (1)$$

що забезпечується адаптивним фільтром-предиктором [4], може бути записано у вигляді

$$\hat{x}(k) = F(X(k)), \quad (2)$$

де $\hat{x}(k) = (\hat{x}_1(k), \hat{x}_2(k), \mathbf{K}, \hat{x}_n(k))^T$ – $(n \times 1)$ -вектор прогнозів, отриманих за передісторією наявної до моменту часу k , $X(k) = (1, x_1(k-1), \mathbf{K}, x_1(k-d), \mathbf{K}, x_n(k-d))^T = (1, \%_1(k), \mathbf{K}, \%_n(k), \%_{n+1}(k), \%_{nd}(k))^T$ – $((nd+1) \times 1)$ -вектор цієї передісторії, W – $(n \times (nd+1))$ – матриця налаштованих синаптичних ваг лінійної конструкції (1), $F(\bullet)$ – нелінійний оператор спеціального вигляду, що забезпечує оптимальне в сенсі прийнятого критерію прогнозування багатовимірної послідовності $x(k)$.

Нео-фаззі-нейрон

Нео-фаззі-нейрон (NFN), що його ввели Т. Ямакава з колегами [5], є нелінійною системою, яка навчається, схема її стосовно задачі, що розглядається тут, наведена на рис. 2.

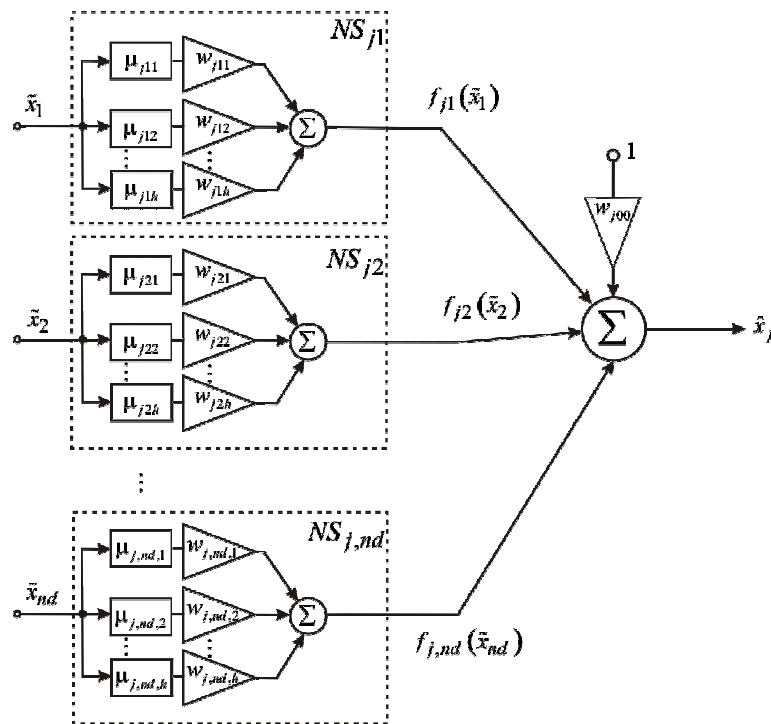


Рис. 2. Нео-фаззі-нейрон

Як можна побачити, NFN за архітектурою є достатньо близьким до стандартного nd -входового формального нейрона, однак на відміну від нього містить нелінійні синапси NS_{ji} , в яких реалізуються елементарні правила нечіткого виведення вигляду

$$IF \mathcal{X}_i IS X_{il} THEN f_{ji}(\mathcal{X}_i) = \sum_{l=1}^h m_{jil}(\mathcal{X}_i) w_{jil}, \quad (3)$$

де X_{il} – лінгвістичне значення (нечітка множина) на i -му вході в антецеденті l -го правила), $m_{jil}(\mathcal{X}_i)$ – функція належності, w_{jil} – синглетон у консеквенті l -го правила, h – кількість функцій належності у кожному нелінійному синапсі.

При подачі на вхід NFN векторного сигналу $X(k)$ на його виході з'являється скалярне значення

$$\hat{x}_j(k) = w_{j00} + \sum_{i=1}^{nd} f_{ji}(\mathcal{X}_i(k)) = w_{j00} + \sum_{i=1}^{nd} \sum_{l=1}^h m_{jil}(\mathcal{X}_i(k)) w_{jil}(k-1), \quad (4)$$

що визначається як функціями належності m_{jil} , так і поточними значеннями налаштованих синаптичних ваг $w_{jil}(k-1)$. Незважаючи на простоту архітектури, можна помітити, що, з одного боку, NFN є достатньо близьким до системи нечіткого виведення (FIS) М. Сугено [6], а, з іншого, – до радіально-базисних нейронних мереж (RBFN) [1] зі скалярними аргументами активаційних функцій. Враховуючи функціональну еквівалентність FIS та RBFN [7], а також їх універсальні апроксимувальні властивості [8, 9], логічно сподіватися на успішне використання NFN у системах прогнозування.

У якості функцій належності в нео-фаззі-нейроні зазвичай використовуються трикутні конструкції, вигляд яких наведено на рис. 3, а їх значення визначаються відстанню між величиною вхідного сигналу \mathcal{X}_i та параметрами центрів c_{jil} :

$$m_{jil}(\mathcal{X}_i) = \frac{c_{ji,l} - \mathcal{X}_i}{c_{ji2}}, \quad (5)$$

$$m_{jil}(\mathcal{X}_i) = \begin{cases} \frac{\mathcal{X}_i - c_{ji,l-1}}{c_{jil} - c_{ji,l-1}}, & \mathcal{X}_i \in [c_{ji,l-1}, c_{jil}], \\ \frac{c_{ji,l+1} - \mathcal{X}_i}{c_{ji,l+1} - c_{jil}}, & \mathcal{X}_i \in [c_{jil}, c_{ji,l+1}], \\ 0 & \text{в протилежному випадку,} \end{cases} \quad (6)$$

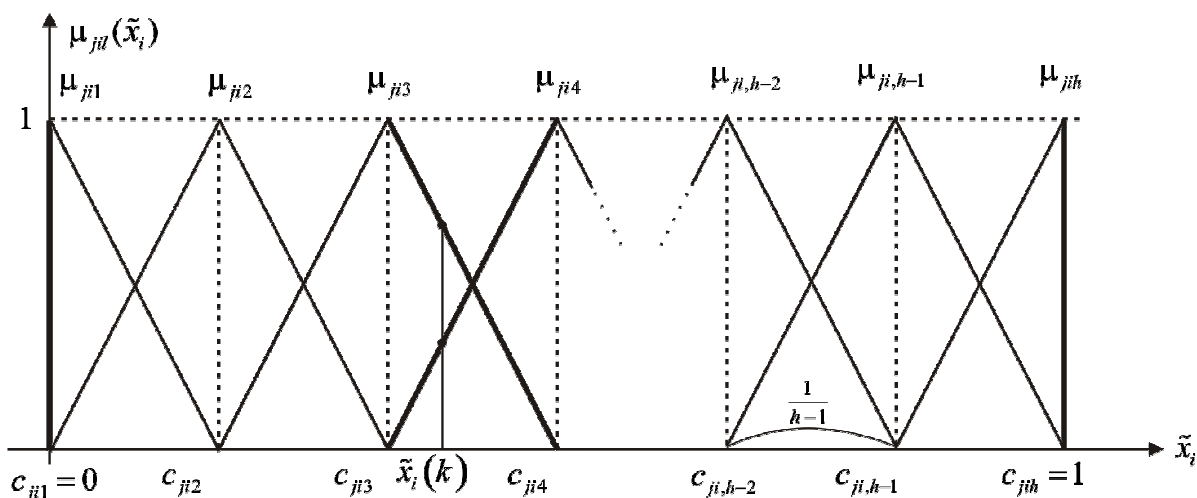


Рис. 3. Функції належності нео-фаззі-нейрона

$$m_{jih}(\mathcal{X}_p) = \frac{\mathcal{X}_p - c_{ji,h-1}}{1 - c_{ji,h-1}}, \quad (7)$$

$$c_{ji1} = 0, c_{ji2} = \frac{1}{h-1}, \mathbf{K}, c_{jil} = \frac{l-1}{h-1}, \mathbf{K}, c_{jih} = 1, \quad (8)$$

при цьому природно вважається, що всі вхідні дані завчасно закодовано в інтервал $\mathcal{X}_p \in [0,1]$.

Важливо відмітити, що така конструкція функцій належності автоматично забезпечує розбиття Руспіні (одиничне розбиття):

$$\sum_{l=1}^h m_{jil}(\mathcal{X}_p) = 1 \quad \forall i. \quad (9)$$

Якщо активним є нечіткий інтервал p , вихід нелінійного синапсу можна записати так:

$$\begin{aligned} f_{ji}(\mathcal{X}_p) &= \sum_{l=1}^h w_{jil} m_{jil}(\mathcal{X}_p) = w_{jip} m_{jip}(\mathcal{X}_p) + w_{ji,p+1} m_{ji,p+1}(\mathcal{X}_p) = \\ &= \frac{c_{ji,p+1} - \mathcal{X}_p}{c_{ji,p+1} - c_{jip}} w_{jip} + \frac{\mathcal{X}_p - c_{jip}}{c_{ji,p+1} - c_{jip}} w_{ji,p+1} = a_{ji} \mathcal{X}_p + b_{ji}, \end{aligned} \quad (10)$$

де

$$a_{ji} = \frac{w_{ji,p+1} - w_{jip}}{c_{ji,p+1} - c_{jip}}, \quad b_{ji} = \frac{c_{ji,p+1} w_{jip} - c_{jip} w_{ji,p+1}}{c_{ji,p+1} - c_{jip}}.$$

Перетворення, виконуване нелінійним синапсом NS_{ji} із фаззифікації поточного сигналу $\mathcal{X}_p(k)$, також проілюстровано на рис. 3. Товстими лініями вказано активні функції належності. Отже, кожен нелінійний синапс реалізує кусково-лінійну апроксимацію $f_{ji}(\mathcal{X}_p)$ нелінійного сигналу \mathcal{X}_p .

Багатовимірний нео-фаззі-нейрон

Архітектура, наведена на рис. 1, є надлишковою, якщо містить однотипові нелінійні синапси NS_{ji} . Це пояснюється тим, що $m_{1il}(\mathcal{X}_p) = m_{2il}(\mathcal{X}_p) = \mathbf{K} = m_{jil}(\mathcal{X}_p) = \mathbf{K} = m_{nil}(\mathcal{X}_p)$. Уникнути цього можна, вводячи до розгляду багатовимірний нео-фаззі-нейрон, архітектура якого наведена на рис.4 і є модифікацією архітектур, розглянутих у [10, 11]. Вузлами цієї конструкції є складені нелінійні синапси MNS_j , кожний з котрих містить h функцій належності m_{jil} (індекс j тут в принципі може бути прийнято довільним $1 \leq j \leq n$) та nh налаштованих синаптичних ваг w_{jih} . Отже, багатовимірний нео-фаззі-нейрон, загалом, містить $ndgh + n = n(ndh + 1)$ синаптичних ваг, але ж тільки ndh функцій належності, що в n разів менше ніж в архітектурі, наведеній на рис. 1.

Отже, нео-фаззі-предиктор можна достатньо просто реалізувати на базі багатовимірного нео-фаззі-нейрона.

Метод навчання нео-фаззі-предиктора

Вводячи до розгляду $(ndh+1) \times 1$ -вектор поточних значень функцій належності $m(k) = (1, m_{j11}(\mathcal{X}_p(k)), \mathbf{K}, m_{j,nd,h}(\mathcal{X}_{hd}(k)))^T$ та $(n \times (ndh+1))$ -матрицю синаптичних ваг

$$W = \begin{pmatrix} w_{100} & w_{111} & \mathbf{L} & w_{11h} & \mathbf{L} & w_{1,hd,h} \\ w_{200} & w_{211} & \mathbf{L} & w_{21h} & \mathbf{L} & w_{2,hd,h} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ w_{300} & w_{n11} & \mathbf{L} & w_{n1h} & \mathbf{L} & w_{n,hd,h} \end{pmatrix}, \quad (11)$$

можна записати сигнал на виході нео-фаззі-предиктора у k -й момент часу у вигляді

$$\hat{x}(k) = F(X(k)) = W(k-1)m(k), \quad (12)$$

де $W(k-1)$ – матриця налаштованих синаптичних ваг, що її отримано на основі інформації, наявної до k -го такту.

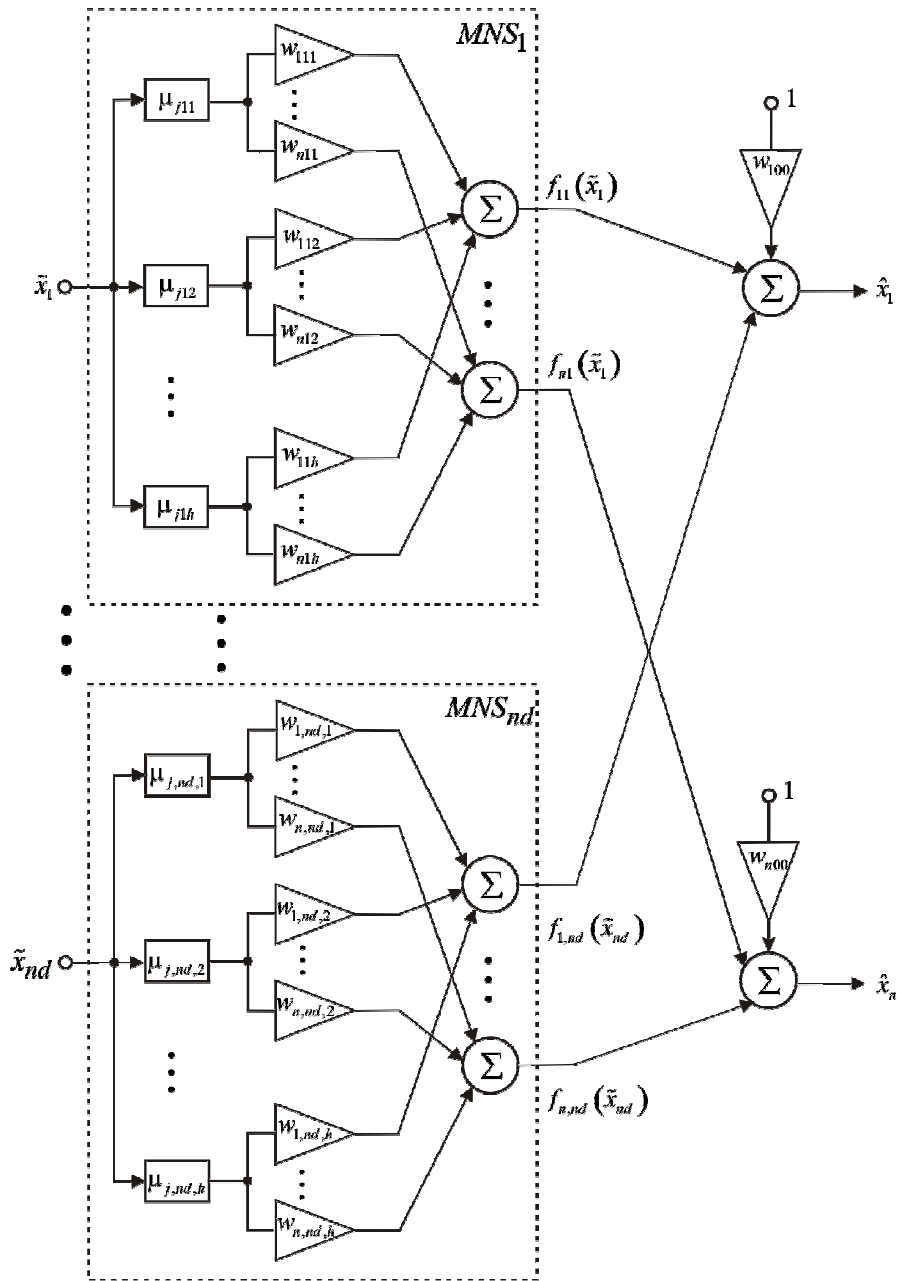


Рис. 4. Багатовимірний нео-фаззі-нейрон

Навчання багатовимірного нео-фаззі-предиктора може бути реалізовано за допомогою матричної модифікації експоненційно-зваженого рекурентного методу найменших квадратів у формі

$$\begin{cases} W(k) = W(k-1) + \frac{(X(k) - W(k-1)m(k))m^T(k)P(k-1)}{a + m^T(k)P(k-1)m(k)}, \\ P(k) = \frac{1}{a} \left(P(k-1) - \frac{P(k-1)m(k)m^T(k)P(k-1)}{a + m^T(k)P(k-1)m(k)} \right), \end{cases} \quad (13)$$

де $0 < a \leq 1$ – параметр згладжування, що забезпечує компроміс між слідкуючими та фільтруючими властивостями алгоритму.

Відзначимо також, що в кожний момент часу k з надходженням нового вектора спостережень часового ряду $x(k)$ спочатку уточнюється матриця синаптичних ваг $W(k)$, а на її основі будується однокроковий прогноз

$$\hat{x}(k+1) = W(k)m(k+1), \quad (14)$$

де вектор $m(k+1)$ містить фаззифіковані дані з часовими індексами, що не перевищують k .

Експериментальне моделювання

Аби продемонструвати ефективність запропонованої адаптивної прогнозуючої нео-фаззі-моделі та її алгоритму навчання, у якості тестової вибірки було обрано багатовимірний ряд на основі диференційних рівнянь моделі Лоренца:

$$\begin{cases} \dot{x} = s(y - x), \\ \dot{y} = x(r - z) - y, \\ \dot{z} = xy - bz. \end{cases} \quad (15)$$

Диференційне рівняння (15) можна записати у дискретному часі в рекурентній формі:

$$\begin{cases} x(i+1) = x(i) + s(y(i) - x(i))dt, \\ y(i+1) = y(i) + (rx(i) - x(i)z(i) - y(i))dt, \\ z(i+1) = z(i) + (x(i)y(i) - bz(i))dt, \end{cases} \quad (16)$$

де $s = 10, r = 28, b = \frac{8}{3}, dt = 0.001$.

Вибірка даних, що отримана за допомогою формули (15), містила 10000 спостережень, серед яких 7000 спостережень – навчальна вибірка, 3000 – перевірна. Процес навчання тривав 1–2 хв залежно від характеристик комп'ютера, на якому було запущено експеримент.

На рис. 5–6 наведено вихід часового ряду та прогноз (темним кольором позначено часовий ряд, світлим – прогноз).

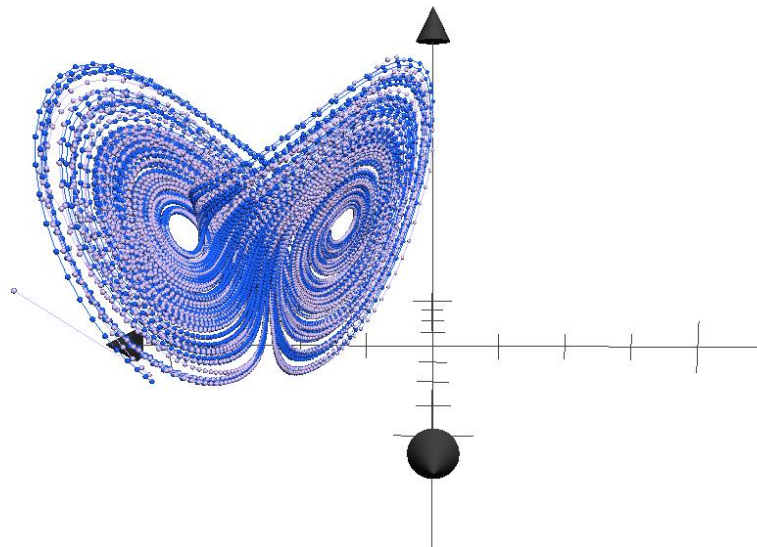


Рис. 5. Візуальна репрезентація прогнозу

Для проведення чисельного експерименту було обрано такі критерії оцінки:

– MSE (mean squared error, середньоквадратична похибка):

$$MSE = \frac{1}{N} \sum (y(k) - \hat{y}(k));$$

– SMAPE (Symmetric mean absolute percentage error, симетрична абсолютна відсоткова похибка) – характеризує похибку прогнозу у відсотках:

$$SMAPE = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{y(k) + \hat{y}(k)}.$$

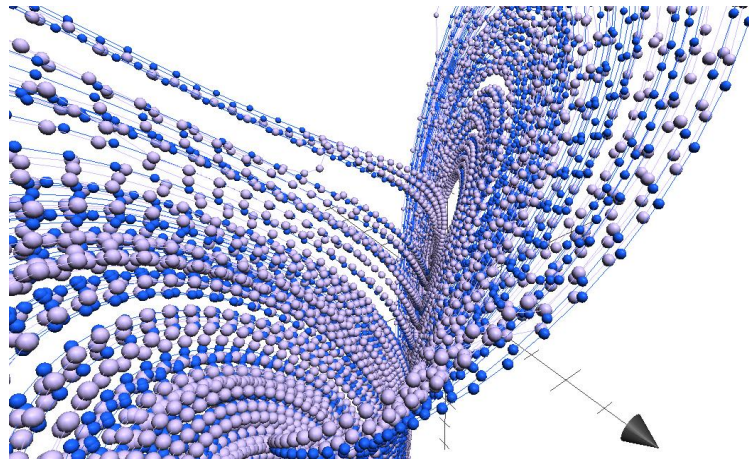


Рис. 6. Деталізована візуалізація прогнозу

На рис. 7–8 наведено змінення у часі характеристик прогнозування.

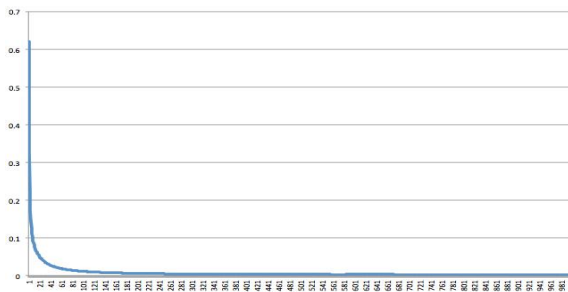


Рис. 7. Середньоквадратична похибка (MSE)

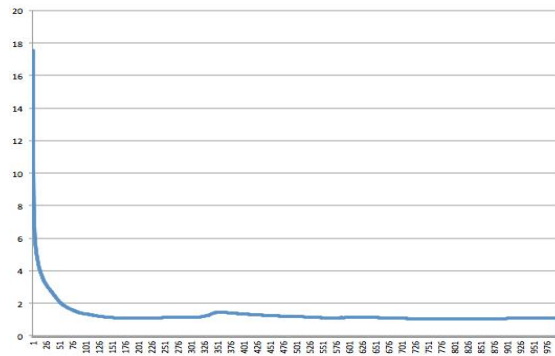


Рис. 8. Абсолютна похибка прогнозування (SMAPE)

На основі експерименту можна стверджувати про високу якість розв'язання задачі прогнозування.

Висновки

Запропоновано адаптивну систему для прогнозування багатовимірних нестационарних часових рядів у режимі реального часу. Введено неймережеву архітектуру, вузлами якої є нео-фаззі-нейрони, оскільки її узагальнення – багатовимірний нео-фаззі-нейрон, що характеризується простотою чисельної реалізації та можливістю швидкого та ефективного навчання.

1. Haykin S. *Neural Networks. A Comprehensive Foundation*. – Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. – 842 p. 2. Mandic D.P., Chambers J.A. *Recurrent Neural Networks for Prediction*. – Chichester: John Wiley & Sons, 2001. – 285p. 3. Jang J.-S. R., Sun C.-T., Mizutani E. *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. – Upper Saddle River, N.J.: Prentice Hall, Inc., 1997. – 614p. 4. Бодянский Е.В., Романюк О.А., Удовенко О.С. Адаптивный фильтр-предиктор многомерных существенно нестационарных временных рядов // Системи обробки інформації. – 2009. – Вип. 4 (78). – С. 23–28. 5. Yatake T., Uchino E., Miki T., Kusanagi H. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior // Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks “IIZUKA-92”. – Iizuka, Japan, 1992. – P. 477–483.

6. Sugeno M. *An introductory survey of fuzzy control* // *Information Sciences*. – 1985. – 36. – P. 59–83.
7. Jang J.-S. R., Sun C.-T. *Functional equivalence between radial basis function networks and fuzzy inference systems* // *IEEE Trans. on Neural Networks*. – 1993. – 4. – P. 156-159.
8. Park J., Sandberg I.W. *Universal approximation using radial-basis-function network* // *Neural Computation*. – 1991. – 3. – P. 246–257.
9. Wang L.X. *Fuzzy systems are universal approximators* // *Proc. 1st IEEE Conf. on Fuzzy Systems*. – San Diego. – 1992. – P. 1163-1169.
10. Landim R.P., Rodrigues B., Silva S.R., Matos W. *A neo-fuzzy-neuron with real-time training applied to flux observer for an induction motor* // *Proc. Vth Brazilian Symp. on Neural Networks*. – Los Alamitos, CA: IEEE Computer Society, 1998. – P. 67–72.
11. Бодянский Е.В., Тищенко А.К. *Адаптивный нео-фаззи-предиктор многомерных нестационарных временных рядов* // *Системы обработки информации*. – 2012. – Вып. 3(101). – Т.2. – С. 8–12.

ЗМІСТ

АРХІТЕКТУРА ТА КОМПОНЕНТИ КОМП'ЮТЕРНИХ СИСТЕМ

<i>Пастернак І.</i> Параметризовані мережні інтерфейси	3
<i>Тимощук П.</i> Аналогова структурно-функціональна нейронна схема визначення максимальних сигналів	10
<i>Овсяк В., Козелко М.</i> Модель конвертації абстрактних до типових графічних унітермів автоматів	18
<i>Пиц М., Деміда Б.</i> Використання MVC для створення соціальної мережі ветеринарних клінік	23
<i>Шаховська Н., Виклюк Я.</i> Застосування сховищ та простори даних у системах прийняття рішень	28
<i>Висоцька В., Окрушко О.</i> Інтелектуальна система розподілу дайджестів між працівниками електронних засобів масової інформації	41
<i>Мухін В., Корнага Я., Снегірев Л.</i> Підвищення ефективності механізмів пошуку в базах даних на основі К-дерев	53
<i>Цмоць І., Скорохода О., Андрієцький Б., Ткаченко Р.</i> Апаратна реалізація інтелектуальних компонентів робототехнічних систем	58
<i>Леонтьєва О., Обельовська К.</i> Порівняння пропускних здатностей схем доступу DCF та EDCA безпроводних локальних мереж	64
<i>Гече Ф., Коцовський В., Батюк А.</i> Спектральні властивості булевих функцій, реалізованих одним нейронним елементом і суматорами за модулем 2	69
<i>Дедів І.</i> Комп'ютерне опрацювання дихального шуму синфазним методом для підвищення інформативності аускультативних систем	77
<i>Матвійків О., Дмитришин Б.</i> Пасивний Т-мікроміксер з бар'єрами для ламінарного потоку	82
<i>Мельник В.</i> Дослідження характеристик засобів генерування програмних моделей спеціалізованих процесорів у самоконфігурованій комп'ютерній системі	86

МОДЕЛЮВАННЯ ПРОЦЕСІВ І СИСТЕМ

<i>Промович Ю.</i> Застосування штучної нейронної мережі прямого поширення сигналу для реконструкції зображень біполярної імпедансної томографії	94
<i>Бодяньський С., Винокурова О.</i> Адаптивний w-нейрон та його навчання в задачах прогнозування і виявлення розладнань	100
<i>Яковлев С.</i> Доказова та практична стійкість R-схеми блочного шифрування до диференціального криптоаналізу	107
<i>Вартузов В., Бабічев С., Литвиненко В., Фефелов А.</i> Модель аналітичної системи профілювання наркотичних речовин на основі мережі Байеса	114
<i>Мельник Г.</i> Метод знаходження відповідних точок на контурах мікрооб'єктів біомедичної природи	120

<i>Овсяк О., Овсяк В., Петрушка Ю.</i> Несуперечність алгебри секвенційних алгоритмів.....	128
<i>Ковальчук А., Пелешко Д., Борзов Ю.</i> Використання побітових операцій і додаткового зашумлення в алгоритмі RSA при шифруванні-дешифруванні зображень.....	132
<i>Литвин В., Бойчук М.</i> Постановка задачі оцінювання новизни онтологічних знань інтелектуальних систем.....	137
<i>Висоцька В., Шестакевич Т., Щербина Ю.</i> Утворення речень англійською та німецькою мовами за допомогою породжувальних граматик.....	142
<i>Яворський Н., Фармага І., Марікуца У.</i> Розроблення дискретної моделі знаходження ефективних теплофізичних характеристик композитних матеріалів зі складною структурою.....	152
<i>Кравець П.</i> Мультиагентна ігрова модель прийняття рішень з корельованими стратегіями....	159
<i>Притула Н.</i> Математичне моделювання перехідних процесів в системах транспортування газу.....	169
<i>Стрямець С., Стрямець О.</i> Геоінформаційний підхід до оцінювання емісії та поглинань парникових газів в лісовому господарстві.....	173
<i>Різник О., Дубик Б., Куліш Н., Чабан Т.</i> Інформаційні технології оптимізації структур систем на основі комбінаторних методів.....	177
<i>Тимчак М., Хвостівський М., Дедів Л.</i> Математична модель електрогастроентеросигналу для підвищення достовірності електрогастроентеросистем.....	181
<i>Гамар Вікторія, Гамар Віталія, Деміда Богдан.</i> Дослідження методики пошуку оптимального маршруту пасажирів у транспортній мережі міста.....	187
<i>Гоц Н.</i> Моделювання методик реалізації температурної шкали багатоточковим методом для термометрів випромінення.....	192
<i>Кузьмін О., Ковальчук А.,¹Ступень Р.</i> Розділення території на економічні складові в умовах невизначеності.....	198

МЕТОДИ І АЛГОРИТМИ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

<i>Шевчук Д.</i> Концепція реконфигурації управління для відновлення управляємості і устійливості самолета в отказних ситуаціях.....	204
<i>Kokosiński Z.</i> Parallel metaheuristics in graph coloring.....	209
<i>Дратан Я., Овсяк В., Овсяк О.</i> Методологія синтезу моделей алгоритмічної складової автоматів.....	215
<i>Передерій В., Бабичев С., Литвиненко В.</i> Застосування мережі Байєса для оцінки ступеня значимості впливаючих факторів на ЛПП в автоматизованих системах при прийнятті релевантних рішень.....	221
<i>Пукач А., Теслюк В., Іванців Р.-А., Лобур М.</i> Розроблення методу визначення діапазону значень величини вимірюваного електричного опору.....	227
<i>Павельчак А., Самотий В., Василів Р.</i> Алгоритм керування двохосьовою платформою.....	232
<i>Іванов Ю., Пелешко Д., Ковальчук А., Пелешко М.</i> Пошук траєкторій рухомих об'єктів для автоматизованих систем відеоспостереження.....	239
<i>Гринів О., Притула Н., Притула М.</i> Математичне моделювання та оптимізація сумісної роботи газосховищ.....	243

<i>Верес О., Верес Ю.</i> Розподіл обмежених ресурсів засобами СППР.....	248
<i>Кутельмах Р., Павич Н., Андрухів Х.</i> Алгоритмічні та програмні засоби опрацювання зображень в операційній системі MAC OS X.....	264
<i>Бойко Н.</i> Умови, стратегія та перспективи розвитку інформаційних технологій.....	271
<i>Заяць В.</i> Підходи до побудови екоінформаційних систем на основі інформаційно-комп'ютерних технологій	277
<i>Різник В.</i> Застосування теорії елегантних обертових симетрій-асиметрій для створення новітніх інформаційних технологій.....	286
<i>Павлишенко Б.</i> Класифікація повідомлень груп новин у векторному просторі семантичних полів.....	294
<i>Процько І.</i> Синтез та обчислення основних типів ДПХ на основі циклічних згорток	302
<i>Бодяньський Є., Тищенко О., Копаліані Д.</i> Прогнозування багатовимірних нестационарних часових рядів на основі адаптивної нео-фаззі-моделі.....	312

Збірник наукових праць

ВІСНИК
Національного університету
“Львівська політехніка”

Видається з 1964 р.

№ 744

**КОМП'ЮТЕРНІ
НАУКИ
ТА ІНФОРМАЦІЙНІ
ТЕХНОЛОГІЇ**

Редактор *Ольга Дорошенко, Галина Клим*

Комп'ютерне верстання *Ірини Жировецької, Галини Сукмановської*

Художник-дизайнер *Уляна Келеман*

Здано у видавництво 18.12.2012. Підписано до друку 27.12.2012.

Формат 60×84¹/₁₆. Папір офсетний. Друк офсетний.

Умовн. друк. арк. 31,8. Обл.-вид. арк. 30,4.

Наклад 120 прим. Зам. 121185.

Видавець і виготівник: Видавництво Львівської політехніки
Свідоцтво суб'єкта видавничої справи ДК № 4459 від 27.12.2012 р.

вул. Ф. Колесси, 2, Львів, 79000
тел. +380 32 2582146, факс +380 32 2582136
vlp.com.ua, ел. пошта: vnr@vlp.com.ua