ISSN 2074-5893

ПИТАННЯ ПРИКЛАДНОЇ МАТЕМАТИКИ І МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

2010

ISSN 2074-3393

Міністерство освіти і науки України

Дніпропетровський національний університет ім. Олеся Гончара

ПИТАННЯ ПРИКЛАДНОЇ МАТЕМАТИКИ І МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Збірник наукових праць

Дніпропетровськ
Видавництво Дніпропетровського
національного університету

2010

УДК 001, 004, 007, 378, 514, 519, 532, 534, 539, 621, 631, 637, 669, 681
ББК 22.12я431+22.311я431+22.176я431+22.18я431
П 32

Друкується за ухвалою вченої ради
Дніпропетровського національного університету імені Олеся Гончара

Уміщені результати фундаментальних досліджень та практичних розробок з проблем математичного та програмного забезпечення інтелектуальних систем. Статті присвячені питанням математичного моделювання складних прикладних систем і розробки ефективних обчислювальних методів та алгоритмів їхнього розв'язання, а також оптимізації, чисельних методів, функціонального аналізу, математичної фізики. Для науковців, викладачів вузів, може бути корисним аспірантам і студентам

П 32 **Питання прикладної математики і математичного моделювання**: зб. наук. пр. / ред. кол. … О. М. Кісельова (голов. ред.) та ін. – Д.: Вид-во Дніпропетр. нац. ун-ту, 2010. – 344 с.

Содержатся результаты фундаментальных исследований и практических разработок по проблемам математического и программного обеспечения интеллектуальных систем. Статьи посвящены вопросам математического моделирования сложных прикладных систем и разработке эффективных вычислительных методов и алгоритмов их решения, а также оптимизации, численных методов, функционального анализа, математической физики. Для ученых, преподавателей вузов, может быть полезен аспирантам и студентам

УДК 001, 004, 007, 378, 514, 519, 532, 534, 539, 621, 631, 637, 669, 681
ББК 2.12я431+22.311я431+22.176я431+22.18я431

Рецензенти:
д-р фіз.-мат. наук, проф. **В. В. Лобода**
д-р фіз.-мат. наук, проф. **С. Б. Вакарчук**

УДК 519.8

©**Т.В. Бесчастний, О.Д. Фірсов**
Дніпропетровський національний університет імені Олеся Гончара

СИСТЕМА НЕПРЯМОЇ ОЦІНКИ ПАРАМЕТРІВ ВЗАЄМОДІЮЧИХ ОБ'ЄКТІВ ДЛЯ ПРОЕКТУ NETFLIX

Побудована ефективна система рекомендацій для проекту компанії NetFlix у рамках сучасних обчислювальних можливостей. Результатом є аналіз існуючих результатів по проблематиці, створення програмного фрэймворка для умовного алгоритму рекомендацій, розробка та реалізація удосконаленого алгоритму рекомендацій, обчислювальний експеримент на тестових даних пропонованих NetFlix. Також проведений кількісний і якісний аналіз роботи алгоритму.

Построена эффективная система рекомендаций для проекта компании NetFlix в рамках современных вычислительных возможностей. Результатом являются анализ существующих результатов по проблематике, создание программного фрэймворка для условного алгоритма рекомендаций, разработка и реализация усовершенствованного алгоритма рекомендаций, вычислительный эксперимент на тестовых данных предлагаемых NetFlix. Также проведен количественный и качественный анализ работы алгоритма.

The research suggests the optimal choice of recommendation system algorithm with taking into account scale of the NetFlix problem and restrictions of modern computers. The results of this work are: analysis of existing results, creation of the experimental program framework for the common recommendation system algorithm, implementation of the chosen recommendation system, calculation of the implemented algorithms on the test data supplied by NetFlix is performed. Qualitative and quantitative analysis of the algorithms work is done.

Ключові слова: система рекомендацій, колаборативна фільтрація, аналіз за змістом, метод k-найближчих сусідів.

Вступ. Останні часи спостерігається бурхливий ріст у теоретичних та практичних дослідженнях в галузі систем рекомендацій, оскільки актуальність цих систем для ринку IT важко переоцінити. Ця галузь зараз починає відігравати не менш важливу роль, ніж галузь пошуку інформації. Багато IT-гігантів, таких як Amazon, NetFlix, TiVo, iTunes, CDNOW, Musicmatch, Everyone's Critic, USENET, та інші використовують системи

рекомендацій, як потужний інструмент для більш повного та індивідуального задоволення потреб користувачів, завдяки більш спрямованої реалізації своїх послуг.

Системи рекомендацій мають різне застосування, наприклад: Amazon – рекомендує покупки багатьох видів товарів, NetFlix займається прокатом DVD.

Задача, що запропонувала компанія NetFlix, була взята з предметної галузі її бізнесу – прокату DVD фільмів та їх рекомендацій. Вона призвела до значного підйому інтересу до галузі систем рекомендацій – розроблено багато різних алгоритмів і побудовано багато моделей для дослідження і використання. Специфіка поставленої NetFlix задачі також пов'язана з розмірністю даних: у системі є 17770 фільмів, 480 189 користувачів і більш ніж 100 000 000 оцінок.

Аналіз існуючих результатів. Насам перед слід зазначити відсутність строгої теорії побудови систем рекомендацій. У літературі можно знайти навіть різні визначення поняття системи рекомендацій. Наприклад, наступні.

Система рекомендацій – система, що надає рекомендацію щодо відповідності об'єкту вимогам користувача, за попередньою історією оцінювання та іншою інформацією [2].

Системи рекомендацій аналізують інтерес користувача в об'єкті, задля надання індивідуальної рекомендації [3].

Система рекомендацій повинна [9]:

- Видавати користувачу список об'єктів, які йому, ймовірно, сподобаються. При цьому користувач надає список об'єктів, які йому подобаються і після цього система аналізує свою внутрішню базу.
- Здійснювати збір і збереження даних про смак користувачів.
- У процесі роботи виконувати самонавчання (автоматично регулювати свої внутрішні параметри, за якими вона оцінює ймовірність того, що даний об'єкт сподобається даному користувачу).

Теоретична база систем рекомендацій торкається таких дисциплін, як алгебра, нейронні мережі, теорія оптимізації, теорія прийняття рішень, психологія та інші.

З огляду на різноманітність галузей використання, різні моделі систем рекомендацій достатньо сильно відрізняються. Необхідність дослідження різних методів і моделей диктується ще й тим, що ефективність багатьох методів розв'язання значно залежить від індивідуальних особливостей задачі.

Загальна постановка задачі, приведена в [3] виглядає наступним чином:

Нехай $U=\{u_j, j=1,...,m\}$ – це множина користувачів, а $I=\{i_j, j=1,...,n\}$ – множина об'єктів, з якими працюють користувачі. Кожен користувач, здатен давати оцінку (рейтинг) r_{ui} об'єкта, що він переглянув.

Кожен користувач u_j надає список об'єктів з оцінками. Система рекомендації у відповідь повинна надавати список об'єктів, які мають потенційно сподобатися іншому користувачу (з відповідною потенціальною оцінкою).

Введемо матрицю

$$R=\{r_{ui}, u\in U, i\in I\}, \quad (1)$$

де елемент r_{ui} – оцінка користувача u об'єкта i. Взагалі кажучи, частина елементів матриці – ненульова (відомі оцінки), а частина – нульова (невідомі оцінки).

Введемо множину

$$V=\{(u,i), r_{ui}\neq 0, u\in U, i\in I\} \quad (2)$$

– індексів відомих оцінок, та множину відомих оцінок

$$R_v=\{r_{ui}, (u, i)\in V\}. \quad (3)$$

Усі елементи r_{ui} множини R_v відомі з матриці R.

Нехай $$N=\{(u,i), r_{ui}=0, u\in U, i\in I\} \quad (4)$$

– множина індексів невідомих оцінок. Множину невідомих оцінок будемо позначати

$$R_N=\{r_{ui}, (u, i)\in N\}. \quad (5)$$

Усі елементи R_N – невідомі. Елемент $r_{ui}\in R_N$ – точна оцінка користувача u, яку він ще не поставив об'єкту i.

Позначимо через

$$\bar{R}_N=\{\bar{r}_{ui}, (u, i)\in N\} \quad (6)$$

– множину потенційних (наближених) оцінок користувачів, що вони поставили би б ще не переглянутим об'єктам.

Отже задача системи рекомендацій – це за відомою множиною R_v оцінити значення елементів R_N – побудувати множину $\bar{R}_N$. У [2] надана загальна класифікація методів і підходів у галузі систем рекомендації. Автори виділяють три основних підходи (моделі), щодо побудови алгоритмів.

Аналіз за змістом (content-base analysis). У літературі цей підхід також називається частковою фільтрацією. До цього підходу відносять алгоритми, засновані на аналізі змісту (опису) об'єкта голосування зі змістом профілю користувача (частковій обробці). Вони дають рекомендації на

основі порівняння інформації про предмет рекомендації з уявленнями про те, що цікаво користувачам. Даний підхід в основному ґрунтується на кластеризації профілів користувачів та опису об'єктів з подальшим знаходженням відповідності та кількісної міри відповідності.

Оскільки має місце робота зі змістом (в основному в текстовому вигляді), то даний клас алгоритмів є спеціалізованим для конкретної предметної галузі. Зазвичай основна проблема побудови такої системи полягає в тому, що потрібне об'єктивне, точне і обов'язкове заповнення змісту профілів, що достатньо важко досягти у промисловій системі.

До того ж практика показала неефективність використання даного класу методів і віддає перевагу наступному класу.

Колаборативна фільтрація (collaborative filtering). Даний підхід ще називається загальною фільтрацією. Він є найпоширенішим у галузі систем рекомендацій і поєднує собою декілька моделей і багато алгоритмів.

Колаборативна фільтрація – підхід, де рекомендації надаються з урахуванням тільки попередньої історії оцінювання. Оскільки в даному класі моделей використовується лише інформація про попередню історію оцінювання і виключається будь-яка додаткова інформація про користувачів і об'єкти оцінювання, то одну модель (а також алгоритм) можна використовувати в різних предметних галузях без змін.

Найпоширеніша модель у даному підході і набір алгоритмів, що базуються на даній моделі є модель найближчих сусідів.

Метод k найближчих сусідів (k Nearest Neighbor Method). Нехай маємо множину користувачів U, множину об'єктів I і матрицю оцінок R. Позначимо терміном користувач вектор, розмірності n, усіх оцінок, що належать користувачу u_j, позначимо терміном об'єкт вектор, розмірністю m, усіх оцінок, що належать даному об'єкту i_j.

Уведемо поняття схожості між користувачами чи об'єктами. Будемо казати, що користувач u_j є схожим на користувача u_k, якщо вони здатні надавати одним і тим самим об'єктам схожі оцінки. Кількісну міру схожості визначає конкретна реалізація методу. Як правило, її вибирають із двох схожих між собою мір – косинуса між векторами чи коефіцієнта кореляції Пірсона.

Значення косинуса схожості між користувачами обчислюється

$$Sim_{u,v} = \frac{\sum_{i=1}^{n} r_{u,i} r_{v,i}}{\sqrt{\sum_{i=1}^{n} r_{u,i}^2 \sum_{i=1}^{n} r_{v,i}^2}}. \qquad (7)$$

Між об'єктами

$$Sim_{i,j} = \frac{\sum_{u=1}^{m} r_{u,i} r_{u,j}}{\sqrt{\sum_{u=1}^{m} r_{u,i}^2 \sum_{u=1}^{m} r_{u,j}^2}} \quad (8)$$

Коефіцієнт кореляції між користувачами обчислюється

$$Sim_{u,v} = \frac{\sum_{i=1}^{n} (r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i=1}^{n} (r_{u,i} - \bar{r_u})^2 \sum_{u=1}^{m} (r_{u,i} - \bar{r_v})^2}} \quad (9)$$

Між об'єктами

$$Sim_{i,j} = \frac{\sum_{u=1}^{m} (r_{u,i} - \bar{r_i})(r_{u,j} - \bar{r_j})}{\sqrt{\sum_{u=1}^{m} (r_{u,i} - \bar{r_i})^2 \sum_{u=1}^{m} (r_{u,j} - \bar{r_j})^2}} \quad (10)$$

Розглянемо два варіанти реалізації методу k найближчих сусідів.

Орієнтований на користувачів (user -oriented). У цьому підході алгоритм працює з користувачами системи (векторами оцінок користувачів). Ідея методу – знаючи вектори оцінок користувачів у системі встановити міру схожості між ними і намагатися представити невідомий рейтинг для даного користувача через лінійну комбінацію (зважену суму) k його сусідів.

Нехай маємо задачу системи рекомендацій у термінах (1)–(5). Наша задача – побудувати множину наближених оцінок $\bar{R}_N$. Для визначеності будемо розуміти обчислення на кожному кроці методу лише однієї наближеної оцінки $\bar{r}_{ui}$.

На першому кроці будуємо матрицю S, розміром (m×m), де m – кількість користувачів у системі. Кожен елемент матриці $s_{u,v}$ – кількісна міра схожості між користувачами u, v. Зазвичай коефіцієнт схожості визначається формулами (7) -(9). Оскільки користувачів у системі, як правило, велика кількість, то коефіцієнт схожості повинен обчислюватися швидко, бажано за лінійний час (лінійно від кількості об'єктів у системі).

На другому кроці будуємо список коефіцієнтів схожості для даного користувача системи u та сортуємо його у порядку спадання.

На третьому кроці обираємо k перших коефіцієнтів, що відповідають k найбільш схожим сусідам (користувачам системи). Коефіцієнт k визначається реалізацією алгоритму. Позначимо через множину $N(k,u)$ – множину, що складається із k найближчих сусідів користувача u.

На четвертому кроці обчислюємо невідому шукану оцінку $\bar{r}_{ui}$ як зважену суму оцінок найближчих сусідів за формулою

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{v \in N(k,u)} s_{uv}(r_{vi} - b_{vi})}{\sum_{v \in N(k,u)} s_{uv}}, \qquad (11)$$

де b_{ui} – деяким чином нормоване значення оцінки користувача u чи об'єкта i у системі. Найбільш часто b_{ui} визначається як середня оцінка даного користувача чи об'єкта.

Даний алгоритм обчислює всі значення елементів з множини $\bar{R}_N$.

Як показано в [1; 3; 7] даний підхід показує гарні практичні результати в наступних випадках:

- кількість об'єктів значно перевищує кількість користувачів у системі;
- кожен з користувачів системи має достатньо багато оцінених об'єктів;
- нові об'єкти та користувачі у системі додаються не досить часто;
- система орієнтована не тільки на надання наближеної невідомої оцінки, але й важливе надання списку користувачів, найбільш схожих за смаком з даним.

Орієнтований на об'єкти (item-oriented). Даний підхід є схожим з попереднім (орієнтованим на користувачів), проте замість користувачів алгоритм працює з об'єктами системи [4].

На першому кроці, як і у попередньому підході, будуємо матрицю S, але вже розміром $(n \times n)$, де n – кількість об'єктів у системі. Кожен елемент матриці $S_{i,j}$ – кількісна міра схожості між об'єктами i, j. Для знаходження коефіцієнта схожості використовуються ті ж самі підходи, що і у попередньому методі за формулами (7), (9).

На другому кроці будуємо список коефіцієнтів схожості для даного об'єкта системи та сортуємо його у порядку спадання.

На третьому кроці обираємо k перших коефіцієнтів, що відповідають k найбільш схожим сусідам (об'єктам системи).

Позначимо через множину $N(k,u)$ − множину, що складається із k найближчих сусідів об'єкта u.

На четвертому кроці обчислюємо невідому шукану оцінку $\bar{r}_{ui}$ як зважену суму оцінок найближчих сусідів за формулою

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_k(u,i)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in N_k(u,i)} s_{ij}}. \qquad (12)$$

Даний підхід має переваги у випадку, коли:
- кількість користувачів значно переважає кількість об'єктів системи;
- система орієнтована не тільки на надання наближеної невідомої оцінки, але й важливе надання списку об'єктів, що найбільш схожі між собою [4].

Оскільки обидві реалізації подібні між собою і результати модифікацій одного з підходів легко перенести на інший, то будемо надалі розглядати лише модифікації підходу, орієнтованого на об'єкти.

Розглянемо деякі з багатьох модифікації методів k найближчих сусідів, що мають переваги над класичною (вищеописаною) реалізацією.

Метод k найближчих сусідів з оцінками за замовчуванням. Оскільки, як правило, кількість користувачів і об'єктів у системі достатньо велика, то навіть при достатньо великій кількості оцінок для кожного користувача (об'єкта) її відношення до загальної кількості об'єктів (користувачів) зостається малим. Це призводить до того, що при обчисленні коефіцієнтів схожості та подальшому виборі найбільш схожих сусідів втрачається значна кількість корисної інформації.

Наприклад, об'єкт має лише кілька оцінок (кількох користувачів) і ці оцінки є схожими з оцінками іншого об'єкта, проте в іншого об'єкта оцінок значно більше й інші оцінки значно відрізняються. При цьому система буде вважати даний об'єкт дуже схожим з другим, хоча в даній ситуації він просто має завищені оцінки від кількох користувачів.

Другий приклад – при підрахунку зваженої суми, об'єкти, в яких немає оцінки від даного користувача відкидаються, хоча ці об'єкти є схожими з даним і попадають у список із k сусідів. У цьому випадку ми також втрачаємо значну кількість корисної інформації.

За даною модифікацією пропонується при обчисленні коефіцієнтів схожості і зваженої суми замість пустих рейтингів для деякого об'єкта брати деякі усереднені оцінки даного об'єкта й використовувати їх при обчисленнях. При цьому ми не втрачаємо корисну інформацію від схожих об'єктів і позбавляємося від вищезазначених проблем.

Метод k найближчих сусідів з узагальненими вагами. У [2; 3] наводиться значна модифікація методу k найближчих сусідів, орієнтованого на об'єкти, з метою усунення вищезазначених недоліків.

Для обчислення невідомої оцінки пропонується узагальнена формула зваженої суми

$$\bar{r}_{ui} = \sum_{j \in N(k,i)} w_{ij} r_{uj}. \qquad (13)$$

Для знаходження вагових коефіцієнтів w_{ij} виконуємо мінімізацію функції відхилу відомих оцінок від наближених за всіма оцінками даного користувача для всіх сусідів даного об'єкта

$$\min_{w} = \sum_{u \neq v} (r_{vi} - \sum_{j \in N(k,i)} w_{ij} r_{uj})^2. \qquad (14)$$

У загальному випадку автори пропонують вирішувати дану задачу через розв'язання СЛАР, що отримуємо, вирішуючи (14) методами класичної оптимізації. Після диференціювання, та переносу відомих значень у праву частину рівнянь отримуємо:

$$Aw = b;$$

$$A_{jk} = \sum_{v \neq u} r_{vj} r_{vk};$$

$$b_j = \sum_{v \neq u} r_{vj} r_{vi}.$$

Множина N(k,i) у (13), (14) – множина k найближчих сусідів об'єкта i, що визначається таким самим чином, як і в (12) (класичному підході, орієнтованому на об'єкти).

Авторами даної модифікації було помічено, що оскільки оцінок у системі порівняно мало з загальною кількістю всіх можливих, то добутки $r_{vj}r_{vk}$ і $r_{vj}r_{vi}$ з великою ймовірністю становляться нульовими, що призводить до втрати інформації про відношення між елементами пари об'єктів у подальших обчисленнях, навіть при ненульовому детермінанті матриці A.

У зв'язку з цим авторами запропоновано більш гнучкий узагальнений підхід для отримання коефіцієнтів w_{ij}:

$$\hat{A}\hat{w} = b,$$

$$\bar{A}_{jk} = \frac{\sum_{v \in U(j,k)} r_{vk}}{|U(j,k)|},$$

$$\bar{b}_j = \frac{\sum_{v \in U(j,k)} r_{vi}}{|U(j,k)|},$$

$$\hat{A}_{jk} = \frac{|U(j,k)| \bar{A}_{jk} + \beta \cdot avg}{|U(j,k)| + \beta},$$

$$\hat{b}_j = \frac{|U(j,k)| \bar{b}_j + \beta \cdot avg}{|U(j,k)| + \beta},$$

де коефіцієнт β – емпірично підібраний, що несе у собі сенс регулювання тяжіння до елементу матриці A та загальної середньої оцінки. На практиці, при розв'язанні задачі розмірності поставленої компанією NetFlix пропонується вибір коефіцієнту β=500; avg – загальна середня оцінка у системі (за всіма користувачами та об'єктами); U(j,k) – множина користувачів, що дали оцінку об'єктам i, k.

У даному підході розв'язуємо узагальнену задачу. Матриця $\hat{A}$, взагалі кажучи, не має нульових елементів і є не виродженою, тому, в даному випадку, отримуємо значно інформативніший набір вагових коефіцієнтів w_{ij}.

Можлива також модифікація даного підходу з нормуванням по середній оцінці кожного з користувачів:

$$\min_w = \sum_{u \neq v} \sum (r_{vi} - b_{vi} - \sum_{j \in N(u)} w_{ij}(r_{uj} - b_{uj}))^2,$$

$$\bar{r}_{ui} = b_{ui} + \sum_{j \in N(u)} w_{ij}(r_{uj} - b_{uj}).$$

Моделі виявлення скритих закономірностей (latent factor models) спрямовані на дослідженні скритих закономірностей і відношень між парами користувач, об'єкт. Нехай мємо X – множину тем, p(x|u) – неявний тематичний профіль користувача u, q(x|i) – неявний тематичний профіль об'єкта i. Ці моделі намагаються за оцінками користувача u знайти тематичний профіль цього користувача p(x|u) і запропонувати об'єкти тематичного профілю q(x|i).

Обмежена машина Больцмана (restricted Boltzmann machine) описана в [5] і застосовується як стохастичний алгоритм для обчислення наближеної оцінки.

З огляду на те, що машина Больцмана є нейронною сіткою, то їй притаманна не детермінованість процесу і довгий період навчання. До того ж, як показано в [5], її застосування можливе лише в обмеженому класі задач.

Беручи до уваги розмірність поставленої NetFlix задачі, застосування обмеженої машини Больцмана не є можливим.

Баєсовська модель відвідувань належить до моделей виявлення скритих закономірностей. Через термін відвідування позначається «присутність» оцінки у системі.

Дана модель за відомими ймовірностями, встановлених з відомих оцінок, максимізує ймовірність того, що дана оцінка тематичного профілю даного об’єкта (для якого виконується рекомендація) і даного користувача співпадуть

$$p(u,i)=\sum_x p(x)\,p(u\,|\,x)\,p(i\,|\,x)=\sum_x q(x)\,q(i\,|\,x)\,p(u\,|\,x);$$

$$q(x\,|\,i)=\frac{q(i\,|\,x)\,q(x)}{\sum_s q(i\,|\,s)\,q(s)};$$

$$q(x\,|\,u)=\frac{q(u\,|\,x)\,q(x)}{\sum_s q(u\,|\,v)\,q(v)}.$$

Надалі розв’язуємо задачу максимізації правдоподібності

$$\sum_{j=1}^{m}\ln p(u_j,i_j)\to\max_{p(x\,|\,u),\,q(i\,|\,x)}.$$

Дана модель є практично корисною для роботи на будь-яких об’ємах даних. Як показано в [1-4] вона є корисною для встановлення тих зв’язків у системі, що не можуть встановити методи найближчих сусідів, оскільки надає погляд на вихідні дані з більш високою абстракцією.

Наведемо список сильних сторін даної моделі:

- більш ефективно порівнюються користувачі і об’єкти системи;
- не потрібно великих структур даних тримати у пам’яті;
- можливо порівнювати користувача з об’єктом;
- тематичні профілі гарно інтерпретуються;
- вирішується проблема «холодного старту»;
- можливе «часткове навчання», коли тематичні профілі задаються лише в деяких об’єктів.

Гібридні методи. Обидва з вищезазначених класів методів мають свої недоліки, що можуть бути усунуті взаємодоповненням обох класів алгоритмів. Тому був запропонований ряд моделей і алгоритмів з гібридним (комбінованим) підходом, що усувають одну, чи кілька вищезазначених проблем.

У теоретичній і практичній частині галузі систем рекомендацій існує багато проблем в аспекті якості оцінювання, розгортання системи, моделювання психології користувача і таке інше.

Проблема холодного старту. Як було зазначено в описі підходів, щодо побудови систем рекомендацій на базі колаборативної фільтрації, дані системи погано працюють на початку експлуатації, коли кожен користувач ще не оцінив достатню кількість об'єктів і об'єкти не мають достатньо великої кількості оцінок.

До цієї ж проблеми відносять проблему додавання нового користувача чи об'єкта у систему. У цьому випадку новий користувач чи об'єкт взагалі не має оцінок, з чого не можливо знайти схожий до нього (коефіцієнт схожості буде рівним нулю).

Як розв'язок проблеми холодного старту в [10] пропонується перехід від чистої системи рекомендацій на базі колаборативної фільтрації до гібридної. У цьому випадку, доки користувач чи об'єкт не наберуть достатньої кількості оцінок для них рекомендації будуть визначатися на базі аналізу змісту. В іншому – на базі колаборативної фільтрації. Також у [10] пропонуються інші варіанти розв'язання даної проблеми.

Проблема неправдивості оцінок. Це проблема навмисного завищення чи заниження конкретним користувачем своїх оцінок. Це призводить до того, що в об'єктивність і правильність наближеної оцінки $\bar{r}_{ui}$ вноситься додатковий шум.

Тут можемо виділити дві проблеми:

– користувач ненавмисне природно здатен ставити всім об'єктам більші чи менші оцінки ніж інші;

– користувач навмисне ставить більші чи менші оцінки конкретним об'єктам без закономірності з метою зламування системи.

Як розв'язок першої проблеми пропонується нормалізація оцінок. У найпростішому випадку – це віднімання середнього значення оцінок користувача від значення оцінки, з якою проводяться обчислення, і проведення операцій вже не з оцінками, а відхилами від середньої для даного користувача чи об'єкта.

Достатньо гарного розв'язку другого варіанта проблеми не існує. Пропонуються варіанти знаходження деяким чином подібних користувачів чи об'єктів і їх нейтралізація.

Проблема дисперсії оцінок користувача. Як було помічено – користувачі дають невідповідні друг другу оцінки в різні моменти часу одним і тим ж об'єктам. Якщо розглядати оцінки користувача в різні моменти часу як деякі значення випадкової величини, то можна встановити дисперсію оцінок користувача. Дана проблема належить більш галузі психології і не має теоретичних і практичних пропозицій вирішення.

Також існують інші проблеми в галузі систем рекомендацій, що не розглядаються в даній роботі.

Оцінювання алгоритмів систем рекомендацій – це окрема, достатньо велика і різноманітна, тема. Це є наслідком декількох причин, що описані в [9]. По-перше, різні алгоритми можуть бути гарними чи поганими для різних, за об'ємом, даних. По-друге, багато алгоритмів колаборативної фільтрації були розроблені спеціально для масивів даних, де користувачів значно більше, ніж об'єктів, чи навпаки. Такі алгоритми будуть працювати погано на масивах даних з іншими показниками. По-третє, оцінювати алгоритми систем рекомендацій важко, бо можуть розрізнятися цілі оцінювання. Найперші роботи в області оцінювання концентрувалися на точності алгоритмів колаборативної фільтрації у передбаченні невідомих оцінок (обчисленні наближених оцінок). У більш пізніх роботах (Shardanand і Maes) було показано, що коли системи рекомендацій використовуються з метою допомоги користувачу в прийнятті рішень, то важніше виміряти наскільки часто система приводить користувачів до невірних рішень. Ряд досліджень проводився на тему, на скільки система рекомендацій охоплює увесь набір об'єктів (Mobasher). Оскільки, як вже зазначалося раніше, одна з позитивних сторін систем рекомендацій є неочевидність рекомендацій для користувача, то можна оцінювати ступінь неочевидності зроблених рекомендацій (McNee). Можна оцінювати систему рекомендацій за ступенем задоволення потреб користувача (в комерційних системах це може визначатися кількістю купленого товару). Останній з підходів полягає у поєднані вищезазначених оцінок з різними ваговими коефіцієнтами, при цьому складність полягає у правильному підборі вагових коефіцієнтів для даної предметної галузі.

Актуальним для даної роботи є оцінювання точності алгоритму. Найбільш поширенішим підходом до оцінювання загальної точності алгоритму є обчислення глобальної похибки алгоритму RMSE (root mean square error) чи MSA (mean square error) – середньоквадратичної похибки між відомою оцінкою і наближеною, обчисленою за допомогою алгоритмів системи рекомендацій, на деякому тестовому масиві даних

$$RMSE = \sqrt{\frac{1}{|V|}\sum_{(ui)\in V}(\bar{r}_{ui} - r_{ui})^2},$$

де V – множина індексів відомих оцінок ; $r_{ui} \in R_n$ – множина відомих оцінок (які поставив користувач); $\bar{r}_{ui} \in \bar{R}_n$ – множина наближених оцінок, обчислених за допомогою алгоритмів системи рекомендацій.

Вибір моделі для реалізації та дослідження. Для реалізації при розв'язанні поставленої задачі, було обрано наступні моделі з відповідними міркуваннями:

1) Класичний метод найближчих сусідів, орієнтований на об'єкти.

З умов поставленої задачі видно кілька фактів, при яких доцільно обирати даний метод. Об'єктів (фільмів) у системі NetFlix набагато менше за користувачів. Оскільки NetFlix надає достатньо великий масив робочих даних з реальної системи з достатньо гарною наповненістю, то немає проблеми холодного старту. Це найбільш швидка реалізація з методів колаборативної фільтрації.

2) Метод найближчих сусідів, орієнтований на об'єкти, з узагальненими вагами.

Даний метод поєднує у собі такі переваги як достатньо велика швидкість роботи порівняно з іншими алгоритмами колаборативної фільтрації, більш висока точність оцінювання, порівняно з попереднім обраним методом.

3) Авторська модифікація класичного методу найближчих сусідів, орієнтованого на об'єкти.

Метою створення даної модифікації та її вибору було збільшення точності, порівняно з класичним методом найближчих сусідів, орієнтованого на об'єкти.

Постановка задачі. Компанія NetFlix працює в галузі надання рекомендацій про перегляд фільмів на DVD щодо покращення результатів оцінювання.

Для розробників NetFlix надає тестовий масив даних зі своєї системи рекомендацій. У системі є 480189 користувачів та 17770 фільмів. У тестовому масиві даних більш ніж 100 000 000 оцінок, проставлених користувачами.

Разом з цим надано файл для дослідження роботи алгоритму системи рекомендацій. Він містить множину індексів підмножини відомих оцінок. Використання даного файлу дає можливість обчислити середньоквадратичну похибку алгоритму (різниця відомої у системі оцінки та оцінки

обчисленої алгоритмом системи рекомендації). Опис структур даних, наданих NetFlix можна подивитися на сайті компанії.

Потрібно побудувати алгоритм для надання рекомендацій про перегляд фільму та підрахувати його похибку на тестовій множині.

Формально задача виглядає як (1) -(5) з урахуванням реальних обмежень.

Алгоритм розв'язку. Як було зазначено вище, для побудови своєї реалізації системи обрано два методи: класичний метод найближчих сусідів та його узагальнений варіант метод найближчих сусідів з узагальненими вагами. Враховуючи переваги і недоліки обох методів запропоновано модифікацію класичного методу найближчих сусідів – метод най-ближчих сусідів з інформативністю об'єктів. Мотивацією у побудові даної модифікації було збільшення точності класичного методу найближчих сусідів з прийнятним збільшенням обчислювальної складності.

Метод найближчих сусідів з інформативністю об'єктів . Позначимо терміном інформативності об'єкта – коефіцієнт, що показує відношення кількості оцінок об'єкта до загальної кількості можливих оцінок (це кількість користувачів).

Позначимо через r_i – множину оцінок об'єкта і (чи просто об'єкт і). Позначимо кількість оцінок об'єкта $c_i = |r_i|$. Нехай c_{min} – мінімальна кількість оцінок у об'єкта у системі, c_{min} – максимальна кількість оцінок у об'єкта у системі.

Визначимо коефіцієнт інформативності

$$\gamma_i = \alpha + \frac{c_i - c_{min}}{c_{max} - c_{min}}(\beta - \alpha),$$

де α, β – емпірично підібрані коефіцієнти. Сенс даних коефіцієнтів у встановленні того факту, наскільки інформативність об'єкта є вагомою в обчисленнях потенційної оцінки.

При практичній реалізації даного алгоритму приймалися α=0,8, β=1,2 з міркувань балансу між покращенням точності роботи алгоритму та можливого збільшення похибки за рахунок шумів при обчисленнях при значному відхилі α, β від 1.

Очевидно, що чим більша інформативність об'єкта, тим більша об'єктивність його оцінок. Користувачі схильні надавати більш популярним об'єктам подібні оцінки і навпаки для більш специфічних – різні.

Модифікація розрахункової формули (12) з урахуванням інформативності користувачів виглядає наступним чином:

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{j \in N(i,u)} \gamma_i s_{ij} (r_{uj} - b_{uj})}{\sum_{j \in N(i,u)} \gamma_i s_{ij}}.$$

Розглянемо надалі етапи роботи побудованої системи оцінювання.

Маємо наступні вхідні дані:

Масив даних, наданих компанією NetFlix включає до себе:

- множину відомих оцінок R;
- метаінформацію про оцінки, користувачів, об'єкти;
- множину індексів невідомих оцінок (4).

Перший етап це препроцесінг, в якому база даних оцінок, матриця коефіцієнтів схожості, та масив середніх значень оцінок об'єктів обчислюються і зберігаються аналогічно попередньому методу. Крім того, обчислюємо множину значень c_i (для усіх об'єктів), c_{min}, c_{max}.

Далі для кожного індексу невідомої оцінки виконуємо:

1. Дістаємо із бази даних список із найближчих сусідів об'єкта і з відповідними коефіцієнтами схожості. Сортуємо список за не зростанням.
2. Будуємо новий список сусідів, де проставлена оцінка у користувача u. Для цього виконуємо наступні дії:

йдучи по списку, починаючи з початку, перевіряємо чи поставлена в даного сусіда оцінка користувача u:

- якщо ні, то йдемо далі по списку;
- якщо так, то додаємо в новий список;
- якщо ми дійшли до кінця списку, то переходимо на 3.

3. Обчислюємо коефіцієнт інформативності γ_i для даного об'єкта.
4. Обчислюємо наближений рейтинг $\bar{r}_{ui}$.

Обчислювальний експеримент проходив у кілька етапів. Як зазначалося раніше перед стартом роботи конкретного методу необхідний препроцесінг – заповнення бази даних та розрахунок і збереження матриці коефіцієнтів схожості між об'єктами, яку використовують всі методи у процесі своєї роботи.

Після препроцесінгу було почергово досліджено кожен із алгоритмів. Підрахунок похибки RMSE відбувався 4 рази для кожного із алгоритмів, для різних значень параметра $k = \{10, 20, 30, 40\}$. Потім досліджувалася похибка на кожному об'єкті із підрахунком дисперсії оцінок об'єкта задля встановлення зв'язку між ними.

Далі результати були порівняні між собою для різних алгоритмів і різних значеннях параметра k.

В якості тестової вибірки даних було використано наданий компанією NetFlix масив. Розрахунок RMSE відбувався на тестовій вибірці для елементів, які мають точні оцінки в базі.

Тестова множина даних, що надається компанією NetFlix у вигляді текстових файлів має об'єм трохи більше за 2 Гб. Для роботи з таким масивом даних перед початком препроцесінгу була створена спеціальна база даних, об'єм якої складав біля 4 Гб. Далі виконувалася операція включення головних ключів у базі даних, при цьому об'єм збільшився майже вдвічі і склав 7 Гб.

На наступному етапі відбувся підрахунок матриці коефіцієнтів схожості. При цьому база даних, в якій зберігалася дана матриця, мала розмір 1,5 Гб.

Першим проводилося дослідження залежності похибки від значення параметра k – кількості сусідів.

На тестовому масиві даних з приблизно 1 300 000 оцінок було проведено чотири обчислення похибки при різних значеннях k = {10, 20, 30, 40}. Результати обчислень приведені на діаграмі (1).

Діаграма 1. Залежність похибки класичного методу найближчих сусідів від кількості сусідів

Другий з методів для дослідження – метод k найближчих сусідів з узагальненими вагами підтвердив факт, наведений авторами алгоритму про збільшення обчислювальної вартості алгоритму лінійно, порівняно з класичною реалізацією. При цьому метод показав більш точні значення загальної похибки алгоритму. Результати наведені на діаграмі (2).

Діаграма 2. Залежність похибки методу найближчих сусідів з узагальненими вагами від кількості сусідів

Третім з досліджуваних алгоритмів був запропонований алгоритм з інформативністю об'єктів. Передбачалося поліпшення якісті роботи алгоритму, порівняно з класичним аналогом.

У ході експерименту було підтверджено дану гіпотезу. Алгоритм мав майже такий самий час роботи, як і в класичного аналога, проте покращені результати роботи. Результати наведені на діаграмі (3).

Діаграма 3. Залежність похибки методу найближчих сусідів з інформативністю об'єктів від кількості сусідів

Другим етапом експерименту було дослідження залежності похибки при оцінці одного об'єкта від дисперсії оцінок даного об'єкта. Висунуто гіпотезу про існування залежності, такої, що при збільшенні дисперсії буде збільшуватися похибка алгоритму. Це може бути наслідком того, що алгоритм не може достатньо точно знайти об'єктивне значення оцінки об'єкта, при великій дисперсії його оцінок.

При обчисленні використовувався наступний підхід: для кожного об'єкта обчислювалася дисперсія, а також обчислювалися похибки всіх оцінок, що знаходилися для даного об'єкта. Оскільки для більшості об'єктів їх дисперсія оцінок лежить у проміжку від одного до двох, то було обрано даний інтервал для дослідження. Він був розбитий на 10 підпроміжків, довжиною 0.1. Далі для всіх об'єктів, дисперсія яких попадає в конкретний проміжок, підраховувалася середня похибка.

Результатом даного дослідження була середня похибка для кожного проміжку на шкалі дисперсії оцінок об'єктів.

З результатів (діаграма 4) видно, що похибка зростає зі збільшенням дисперсії об'єкта.

Діаграма 4. Залежність похибки методу від дисперсії об'єкта

При порівнянні методів можемо побачити, що метод з узагальненими вагами значно випереджає два інші аналоги. Порівнюючи метод з узагальненими вагами та метод з інформативністю користувачів можна зробити висновок, що при обчисленні узагальнених вагових коефіцієнтів у них враховується інформація про інформативність. Однаковою рисою всіх алгоритмів є найкраща робота при виборі кількості сусідів у діапазоні 25 – 30.

Діаграма 5. Порівняння похибок реалізованих алгоритмів в залежності від кількості сусідів

Таким чином, у результаті обчислювального експерименту, показано залежність похибки від кількості сусідів в усіх методах. Оптимальним значенням кількості сусідів є приблизна кількість 25 -30. Показано, що метод з узагальненими вагами показує значно меншу загальну похибку, при лінійному збільшенні обчислювальної складності методу, порівняно з класичним. Чисельно підтверджено, що модифікований метод показує меншу загальну похибку, ніж класичний. Підтверджено гіпотезу про залежність між дисперсією оцінок об'єкта і якістю оцінювання на базі класичного методу.

Висновки. За виконаною роботою проаналізовано галузь систем рекомендацій і обрано два найбільш придатних до розв'язання поставленої задачі методи. Створено модифікацію класичного методу найближчих сусідів, орієнтованого на об'єкти. Досліджено і використано практично такі підходи програмування щодо роботи з великими об'ємами даних, як оптимізація БД і кешування. Створено програмний фреймворк для систем рекомендацій, що може використовуватися практично для створення системи рекомендації з необхідними характеристиками. Створено програмний продукт на базі розробленого фреймворку з реалізацією обраних алгоритмів. Проведено обчислювальний експеримент, у ході якого отримана оцінка роботи обраних алгоритмів. Модифікований метод показує більш гарну оцінку роботи, ніж класичний, проте гіршу за оцінку роботу алгоритму з узагальненими вагами. Показано залежність між дисперсією оцінок об'єкта і якістю оцінювання.

Бібліографічні посилання

1. **Adomavicius G., Tuzhilin A.** Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions // Transactions on Knowledge and Data Engineering. – 2005. – Vol. 17, N 6. – P. 734–749.
2. **Bell R., Koren Y.** Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights // AT&T Labs Research. IEEE Conference on Data Mining (ICDM'07), – 2007. P. 42–49.
3. **Bell R., Koren Y., Volinsky C.** Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems // AT&T Labs Research. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07), – 2007. P. 33–41.
4. **Linden G., Smith B. York J.** Amazon.com Recommendations: Item-to-Item Collaborative Filtering // IEEE Internet Computing. – January/February 2003. – Vol. 7, N 1. – P. 76–80.
5. **Salakhutdinov R., Mnih A., Hinton G.** Restricted Boltzmann Machines for Collaborative Filtering // Proc.24th Annual International Conference on Machine Learning. – 2007.
6. **Sarwar B., Karypis G., Konstan J., Riedl J.** Item-based Collaborative Filtering Recommendation Algorithms // Proc. 10th International Conferenceonthe World Wide Web, P.285–295, 2001.
7. **Wang J., Vriesand A., Reinders M.** Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion // Proc.29th ACM SIGIR Conference on Information Retrieval. – 2006. P.501–508.

8. **Melville P., Mooney R., Nagarajan R.** Content-boosted collaborative filtering for improved recommendations // 18th National Conference on Artificial Intelligence (AAAI). – 2002. – P. 187–192.
9. **Herlocker J., Konstan J., Terveen L., Riedl J.** Evaluating collaborative filtering recommender systems // ACM Translations on Information Systems. Vol. 22(1). – 2004. – P. 296–302.
10. **Schein A., Popescul A., Ungar L., Pennock D.** Methods and Metrics for Cold-Start Recommendations // 25th Annual. International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2002. – P. 253–260.

Надійшла до редколегії 12.12.09

УДК 519.863:534

©**В.Н. Богомаз**
Днепропетровский национальный университет имени Олеся Гончара

ОБ ОДНОЙ ЗАДАЧЕ ОПТИМИЗАЦИИ МЕХАНИЧЕСКОЙ ВИБРОСИСТЕМЫ

Запропонована нова конструкція вбудованої в каток вібросистеми. Побудована математична модель динамічних процесів та поставлена задача оптимізації її роботи. Запропонована схема релаксації поставленої задачі та отримані достатні умови її розв'язності.

Предложена новая конструкция встроенной в каток вибросистемы. Построена математическая модель динамических процессов и поставлена задача оптимизации ее работы. Предложена схема релаксации поставленной задачи и получены достаточные условия ее разрешимости.

The paper proposed a new design built-in roller vibrosistemy. A mathematical model of dynamic processes and the task of optimizing its work. A scheme for the relaxation of the problem and obtain sufficient conditions for its solvability.

Ключевые слова: вибросистема, дебаланс, уплотнение, водило, релаксация.

Введение. Уплотнение грунтов перед возведением различного рода инженерных сооружений является важнейшей операцией технологического процесса строительства. Для обеспечения уплотнения грунтов используются класс грунтоуплотняющих машин. Наиболее эффективными из них являются машины динамического действия, представляющие собой виброкатки и виброплиты. Преимуществом данного класса машин является относительно малая материалоемкость и возможность при том же уплотняющем эффекте снизить вес катка, а следовательно потребляемую мощность. Кроме этого, данный класс машин может выполнять уплотнение грунта на значительную глубину (несколько метров). При этом достигается высокая плотность и устойчивость грунтов. Вибрации в вибрационных машинах порождаются вращением тел со смещенным центром тяжести. Но существующие конструкции вибросистем, встроенных в рабочие органы уплотняющих машин, не позволяют увеличить (уменьшить)

вес машины на достаточно долгий заданный промежуток времени, обеспечить конкретное для каждого типа грунта оптимальное воздействие с точки зрения качества уплотнения.

Цель работы: построение математической модели динамических процессов предложенной конструкции вибросистемы, встроенной в каток, получение достаточных условий разрешимости задачи оптимизации ее работы.

Постановка задачи оптимизации

1.Физическая модель объекта управления. В данной работе предложено усовершенствовать рабочий орган каткового типа уплотняющей машины, поместив в него вибросистему, состоящую из комплекта вращающихся дебалансов. Выберем ее за объект управления. Физическая модель объекта управления представлена на рисунке 1.

Дебаланс – твердое тело со смещенным центром масс относительно оси его вращения.

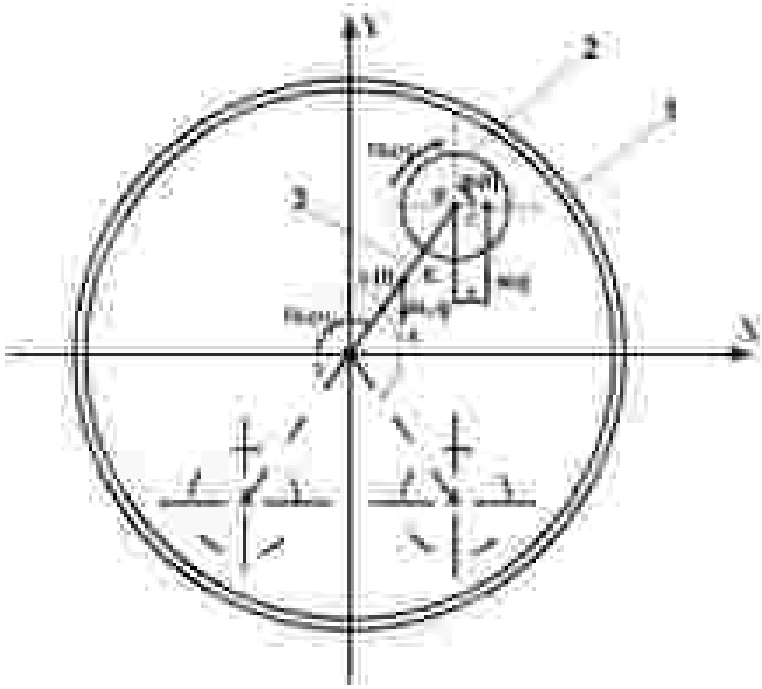


Рис. 1. Физическая модель объекта управления (1 – валец, 2 – дебаланс, 3 – водило)

Предлагаемая вибросистема состоит из заданного фиксированного числа n дебалансов 2 с центрами мас C_i, закрепленного на водиле 3, вращающегося относительно продольной оси вальца. При этом все звенья водила жестко между собой скреплены и центральный угол между двумя соседними звеньями водила является одинаковым для всех звеньев и определяется

$$\lambda = \frac{2\pi}{n}. \qquad (1)$$

При этом заметим, что валец 1, дебаланс 2, водило 3 имеют независимые индивидуальные приводы.

2. Математическая модель объекта управления . Обозначим угол поворота центра масс i -ого дебаланса относительно вертикальной оси через ψ_i, а угол поворота первого звена водила - γ_1. Поскольку будет рассматриваться движение системы во времени, то целесообразно рассматривать эти величины как функции времени, т. е., $\psi_i(t)$ и $\gamma_1(t)$.

При создании математической модели вибросистемы введем некоторые упрощения:

1) пренебрегаем моментами трения во вращательных парах $M_{тр} = 0$;

2) валец со встроенной вибросистемой находится в покое и не движется во время ее работы $v_A = 0$;

3) центры масс звеньев водила лежат на их геометрических серединах;

4) вибросистема начинает работать из состояния покоя и начального положения $\psi\gamma(0) = {}_1(0) \quad 0 \ \forall \in, i[\ ]$.

Водило и дебалансы могут совершать только вращательные движения. Массу i -ого дебаланса обозначим m_i, а i -го звена водила – $m_{iзв}$. Каждое звено водила имеет заданную фиксированную длину R_i и каждый дебаланс имеет заданный фиксированный эксцентриситет r_i. Первая и вторая производные величин $\psi_i(t)$ и $\gamma_1(t)$ являются угловыми скоростями и угловыми ускорениями соответственно водила и дебалансов. На каж - дый дебаланс действует сила тяжести и момент вращения привода, на водило – силы тяжести его звеньев и момент вращения привода. Развиваемые приводами моменты сил обозначим: приложенный к i -ому дебалансу – M_i, к водилу – M_{n+1}.

Для определения уравнений движения системы используем систему дифференциальных уравнений Лагранжа 2 -го рода, которая состоит из $n+1$ уравнений вида:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} = Q_i, \qquad (2)$$

где T – суммарная кинетическая энергия всей системы; Q_i – обобщенная сила, действующая на i-ый элемент системы; q_i и $\dot{q}_i$ – соответственно i-ая обобщенная координата и скорость.

Обобщенными координатами называются s любых независимых физических величин, которые однозначно определяют положение системы (s – число степеней свободы системы).

Обобщенная сила – силовой фактор, совершающий работу при вариации одной из обобщенных координат.

Примем за обобщенные координаты предложенной вибросистемы углы поворота дебалансов и угол поворота водила, т. е. вектор обобщенных координат размерности $n+1$ будет иметь вид $q = \{\psi_1, \psi_2, ..., \psi_n, \gamma_1\}$. Вектор обобщенных скоростей соответственно будет иметь вид $\dot{q} = \{\dot{\psi}_1, \dot{\psi}_2, ..., \dot{\psi}_n, \dot{\gamma}_1\}$.

Угол поворота i-го звена водила относительно вертикальной оси OY определим

$$\gamma_i(t) = \gamma_1(t) + (i-1).$$

Для описания движения дебаланса введем две системы координат: неподвижную XY (инерциальную систему отсчета), и движущуюся систему координат, которая предполагается жестко связанной с дебалансоми участвующей во всех его движениях. Таким образом, движение дебаланса разложим на поступательное движение центра инерции и вращение дебаланса относительно него.

Кинетическая энергия системы имеет вид

$$T = \sum_{i=1}^{n} \left[T_{iзв} + T_{in} + T_{iвр} \right],$$

где $T_{iзв}$ – кинетическая энергия вращения i-го звена водила; T_{in} – кинетическая энергия поступательного движения центра инерции i-го деба-

ланса; $T_{iвр}$ – кинетическая энергия вращения i -го дебаланса относительно центра инерции.

Выразив суммарную кинетическую энергию системы через обобщенные координаты и скорости, получим выражение

$$T = \sum_{i=1}^{n} \left[\frac{I_{iзв} \cdot \dot{\gamma}_1^2}{2} + \frac{m_i L_i^2(\gamma_i, \psi_i)\, \dot{\gamma}_1^2}{2} + \frac{I_{id}\, \dot{i}}{2} \right], \qquad (3)$$

где $I_{iзв} = \frac{m_i R_i^2}{3}$ - момент инерции i -го звена водила относительно оси вращения водила; $I_{id} = \frac{m_i \cdot (R_i^2 \; 2r_i)}{2}$ - момент инерции i -го дебаланса относительно оси, проходящей через центр инерции; $L_i^2(\gamma_i, \psi_i) = R_i^2 + r_i^2 + 2Rr \cdot_i \cdot \cos(\gamma_1 + \lambda(i-1) - \psi_i)$ – расстояние от оси вращения водила до центра инерции i -го дебаланса.

Обобщенную силу, действующую на i -й элемент системы, найдем проварьировав работу всех силовых воздействий по i -й обобщенной координате, т. е. $\delta A^{q_i} = Q_{q_i} \delta q_i$. Таким образом, обобщенные силы, действующие на дебалансы имеют вид

$$Q_i = M_i + mgr \cdot_i \cdot \sin\psi_i, \qquad (4)$$

а на водило

$$\begin{aligned} Q_{n+1} &= M_1 + \sum_{i=1}^{n} m_{зв} \cdot g \cdot \sin(\gamma_1 + \lambda(i-1)) + \\ &+ \sum_{i=1}^{n} mg \cdot \left[R_i \cdot \sin(\gamma_1 + \lambda(i-1)) + r_i \cdot \sin\psi_i \right]. \end{aligned} \qquad (5)$$

Подставляя (3), (4), (5) в (2) и принимая $C_i = m_i R r_i$, $k_i = mgr_i$, $N_{ii} = mg_{в} \frac{R_i}{2}$, $M = \sum_{i=1}^{n} \left(I_{iзв} + m_i L_i^2(\gamma_i, \psi_i) \right)$, $P_{ii} = mg$, получим систему дифференциальных уравнений второго порядка

$$
\begin{cases}
I_{1\partial}\cdot\ddot{\psi}_1 + C_1\sin\left(\gamma_1-\psi_1\right)\cdot\dot{\gamma}_1^2 = M_1 + k_1\cdot\sin\psi_1,\\
I_{2\partial}\cdot\ddot{\psi}_2 + C_2\sin\left(\gamma_1+\lambda-\psi_2\right)\cdot\dot{\gamma}_1^2 = M_2 + k_2\cdot\sin\psi_2,\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
I_{n\partial}\cdot\ddot{\psi}_n + C_n\sin\left(\gamma_1+\lambda(n-1)-\psi_n\right)\cdot\dot{\gamma}_1^2 = Mk_n + {}_n\cdot\sin{}_n,\\
M\cdot\ddot{\gamma}_{11}+\dot{\gamma}_1^2\sum_{i=1}^{n} C_{ii}\sin\left(\gamma+\lambda(i-1)-\psi\right)+\\
+2\ddot{\gamma}_{11}\sum_{i=1}^{n} C_i\sin\left(\gamma+\lambda(i-1)-\psi_i\right)\cdot\psi_i = M_{n+1}+\\
+\sum_{i=1}^{m}\sum_{i=1}^{n} N_i\cdot\sin\left(\gamma_{11}+\lambda\cdot(i-1)\right)+ P_i\cdot\left[R_i\sin\left(\gamma+\lambda\cdot(i-1)\right)+ r_i\cdot\sin\psi_i\right].
\end{cases}
\tag{6}
$$

Обозначим $\ddot{\psi}_i = \xi_i$, γ_{11} ξ_{n+}. Примем за параметры состояния системы углы поворота и угловые скорости дебалансов и водила. Вектор состояния системы обозначим через $x = \{\psi\psi_{1}{}_{2},\ldots,\psi\gamma_{n}{}_{1},\xi\xi_{1}{}_{2},\ldots,\xi_{+1}\}$. Фазовое пространство представляет собой R^{22+}. За управляющие параметры примем моменты сил, приложенные к валам дебалансов и водила. Вектор управления имеет вид $u = \{MM_{1}{}_{2},\ldots,MM_{n}{}_{+1}\}$. Область допустимых управлений является подмножеством пространства R^{n+1}. Всюду далее компоненты вектора u будем называть управлениями, а вектора x - фазовыми координатами.

Приводя систему (6) к нормальному виду $\dot{x} = f(x,u)$ при заданных фиксированных величинах $n,m,{}_{i},{}_{зв},R,{}_{i}$, динамика рассматриваемой системы описывается соотношениями:

$$
\begin{cases}
\dot{x}_{n+2} = \dfrac{1}{I_{1\partial}}\left[u_1 + k_1 \cdot \sin x_1 - C_1 \sin\left(x_{n+1} - x_1\right) \cdot x_2^2{}_2\right], \\
\dot{x}_{n+3} = \dfrac{1}{I_{2\partial}}\left[u_2 + k_2 \cdot \sin x_2 - C_2 \sin\left(x_{n+1} + \lambda - x_2\right) \cdot x_2^2{}_2\right], \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\dot{x}_{2n+} = \dfrac{1}{I_{n\partial}}\left[u_n + k \cdot \sin x_n - C_n \sin\left(x_{n+1} + \lambda(n-1) - x_n\right) \cdot x_{2n}^2{}_2\right], \\
\dot{x}_{2n+2} = \dfrac{1}{M}\left(\begin{array}{l}
+x_{2n+2}^2 \sum\limits_{i=1}^{n} C_i \sin\left(x_{n\ 1} + \lambda(i-1) - x_i\right) - \\
-2\ x_{2n+2} \sum\limits_{i=1}^{n} C_i \sin\left(x_{n+1} + \lambda(i-1) - x_i\right) \cdot x_{n+1\ i} + \\
+u_{n+1} + \sum\limits_{i=1}^{n} N_i \cdot \sin\left(x_n + \lambda \cdot (i-1)\right) + \\
+\sum\limits_{i=1}^{n} P_i \cdot \left[R_i \sin\left(x_{n+1} + \lambda \cdot (i-1)\right) + r_i \cdot \sin x_i\right.
\end{array}\right); \\
\dot{x}_{12} = x_{n+}\ ; \\
\ldots\ldots\ldots; \\
\dot{x}_{n+1} = {}_2\ {}_2.
\end{cases}
\tag{7}
$$

Всюду далее предположим отсутствие случайных внешних факторов на систему. Таким образом, уравнения движения вибросистемы представляют собой систему нелинейных дифференциальных уравнений.

Очевидно, что модель динамическая, сосредоточенная, детерминированная, нелинейная, непрерывная, нестационарная.

3.Ограничения на фазовые координаты и управления . Фазовые координаты системы будем рассматривать как элементы пространства непрерывно дифференцируемых функций $x(\cdot) \in C^1\left(0, T; R^{2n+2}\right)$ и $x(t) = \{x_1(t), x_2(t), \ldots, x_{2n+2}(t)\}$.

Поскольку существующие приводы могут разгонять дебалансы до некоторого конечного значения скорости, множество допустимых фазовых траекторий имеет вид:

$$X_{\partial} = \begin{cases} x \in C^1\left(0, T; R^{2n+2}\right): \alpha_{\min} \le x_i(t) \le \alpha_{\max}, \\ \alpha_{\min} < 0, \alpha_{\max} > 0, \forall i \in [n+2, 2n+2] \end{cases}. \quad (8)$$

Поскольку существующие приводы могут обладать ограниченным моментом вращения, то за множество допустимых управлений примем

$$U_{\partial} = \begin{cases} u_{ii} \in C(0, T; R^{n+1}): m_{\min} \le u_i \le m_{\max}, \\ m_{\min} < 0, m_{\max} > 0, \forall i \in [1, n+1] \end{cases}. \quad (9)$$

Непрерывность функции объясняется исключением из рассмотрения невозможных с точки зрения практики мгновенных переключений моментов вращения, мгновенных изменений углов поворота, угловых скоростей, угловых ускорений дебалансов и водила.

Все $x(t) \in X_{\partial}$ далее будем называть допустимыми состояниями системы, а все управления, удовлетворяющие условию $u \in U_{\partial}$, – допустимыми управлениями. Всюду далее пару (x, u) назовем процессом управления, а $\Xi = \{(x, u) \in X_{\partial} \times U : \text{удовлетворяет системе } (7)\}$ – множеством допустимых процессов управления.

Начальные условия в обозначениях вектора состояния системы имеют вид $x_i(0) = 0, \forall i \in [1, 2n+2]$.

4. Обоснование функционала качества. В качестве возмущающих сил в объекте управления будут выступать силы инерции, возникающие при вращении дебалансов и водила. Построим зависимости проекций суммарной возмущающей силы от функций углов поворота дебалансов и водила.

Проекции ускорений центра масс i-го дебаланса на координатные оси X и Y имеют вид:

$$a_{C_i}^X = a_{AB_i}^n \cdot \sin\gamma_i(t) + a_{AB_i}^{\tau} \cdot \cos\gamma_i(t) - a_{BC_i}^n \cdot \sin\psi_i(t) + a_{BC_i} \cdot \cos\psi_i(t), \quad (10)$$

$$a_{C_i}^Y = -a_{AB_i}^n \cdot \cos\gamma_i(t) - a_{AB_i}^{\tau} \cdot \sin\gamma_i(t) - a_{BC_i}^n \cdot \cos\psi_i(t) - a_{BC_i}^{\tau} \sin\psi_i(t), \quad (11)$$

где $a_{AB_i}^n = \left(\frac{d\gamma_1(t)}{dt}\right)^2 \cdot$ – нормальная составляющая ускорения точки B_i;

$a^{\tau}_{AB_i} = \left(\frac{d^2\gamma_1(\ )}{dt^2}\right)_i$ – касательная составляющая ускорения точки B_i;

$a^{n}_{BC_i} = \left(\frac{d\psi_i(\ )}{dt}\right)^2_i$ – нормальная составляющая ускорения точки C_i;

$a^{\tau}_{BC_i} = \frac{d^2\psi_i(\ )}{dt^2}_i$ – касательная составляющая ускорения точки C_i.

Проекции возмущающих сил, возникающих при вращении i-го дебаланса, на координатные оси X и Y определим как проекции сил инерции:

$$F^{XX}_{ин_{ii}} = -ma_C,$$
$$F^{YY}_{ин_{ii}} = -ma_C. \qquad (12)$$

Проекции сил инерции, которые возникают при вращении звеньев водила, на координатные оси:

$$F^{X}_{зв_i} = m_{зв_i} \cdot \left(\frac{d\gamma_1(t)}{dt}\right)^2 \cdot \frac{R_i}{2} \cdot \sin\gamma_1(t) - m_{зв_i} \cdot \frac{d^2(t)}{dt^2} \cdot \frac{R_i}{2} \cdot \cos_i(t);$$
$$F^{Y}_{зв_i} = m_{зв_i} \cdot \left(\frac{d\gamma_1(t)}{dt}\right)^2 \cdot \frac{R_i}{2} \cdot \cos\gamma_1(t) + m_{зв_i} \cdot \frac{d^2(t)}{dt^2} \cdot \frac{R_i}{2} \cdot \sin_i(t). \qquad (13)$$

Сумма проекций сил инерции, возникающих при вращении водила и всех дебалансов, на координатные оси:

$$\sum X = \sum_{i=1}^{n} F^{X}_{ин_i} + \sum_{i=1}^{n} F^{X}_{зв_i};$$
$$\sum Y = \sum_{i=1}^{n} F^{Y}_{ин_i} + \sum_{i=1}^{n} F^{Y}_{зв_i}. \qquad (14)$$

Отметим, что при четном количестве дебалансов сумма проекций сил инерций от вращения звеньев водила при $\lambda = const$ и $R_i = const$ равна нулю.

Таким образом, при заданных $m_i, m_{зв}, R_i$

$$\Sigma X = X\left(\psi(t),\ \gamma(t),\ \frac{d\psi(t)}{dt},\ \frac{d\gamma(t)}{dt},\ \frac{d^2\psi(t)}{dt^2},\ \frac{d\gamma(t)}{dt}\right);$$

$$\Sigma Y = Y\left(\psi(t),\ \gamma(t),\ \frac{d\psi(t)}{dt},\ \frac{d\gamma(t)}{dt},\ \frac{d^2\psi(t)}{dt^2},\ \frac{d\gamma(t)}{dt}\right),$$

где $\psi(t) = (\psi_1(t), \psi_2(t), .., \psi_n(t))$ – вектор_функция углов поворота центров масс дебалансов; $\frac{d\psi(t)}{dt} = \left(\frac{d\psi_1(t)}{dt}, \frac{d\psi_2(t)}{dt}, ..., \frac{d\psi_n(t)}{dt}\right)$ – вектор_функция угловых скоростей дебалансов; $\frac{d^2\psi(t)}{dt^2} = \left(\frac{d^2\psi_1(t)}{dt^2}, \frac{d^2\psi_2(t)}{dt^2}, ..., \frac{d^2\psi_n(t)}{dt^2}\right)$ – вектор_функция угловых ускорений дебалансов; $\gamma(t) = (\gamma_1(t), \gamma_2(t), .., \gamma_n(t))$ – вектор_функция углов поворота звеньев водила; $\frac{d\gamma_1(t)}{dt}$ – угловая скорость водила; $\frac{d^2\gamma_1(t)}{dt^2}$ – угловое ускорение водила.

Таким образом, используя обозначения вектора состояния системы , получим:

$$\Sigma X = X\left(x(t), \frac{dx(t)}{dt}\right),$$

$$\Sigma Y = Y\left(x(t), \frac{dx(t)}{dt}\right).$$

Для оценки оптимальности процесса управления введем функционал качества. Для увеличения веса катка на достаточно долгий заранее заданный промежуток времени, за счет возникающей при работе вибросистемы возмущающей силы целесообразно рассмотреть как функционал качества, среднее значение проекции суммарной силы инерции на вертикальную ось. Таким образом, функционал качества будет иметь вид

$$B = \frac{1}{T}\int_0^T \Sigma Y(x(t), \dot{x}(t))\, dt. \qquad (15)$$

Подставляя (11) в (12), а (12) и (13) в (14), получим функционал качества вида

$$B = \frac{1}{T}\int_0^T \sum_{i=1}^{n} \left(m_i \left[\frac{2}{2n+2} \cdot R_i \cdot \cos x_{n+1} + \ddot{x}_{2n+2} \cdot R_i \cdot \sin x_{n+1} + x_{n+1} \cdot r_i \cdot \cos x_i + x_{n+1} \cdot r_i \cdot \sin x_i \right. \right. +$$
$$\left. \left. + m_{зві} \frac{R_i}{2} \left[x_{2n+2} \cdot \sin\left(x_{n+1} + \lambda_i(i-1)\right) + x_{2n+2}^2 \cdot \cos\left(x_{n+1} + (i-1)\right) \right. \right. \right. \qquad dt$$

Формулировка задачи оптимизации. Среди всех допустимых управлений $u \in U_\partial$ найти такое, которое, выводя систему (7) из начального состояния $x_i(0) = 0 \quad \forall i \in [1, 2n+2]$, доставляет минимум функционалу (15) на $[0, T]$.

Задача оптимизации имеет вид

$$\inf_{(x,u) \in \Xi} B(x(\cdot), \dot{x}(\cdot)), \quad \Xi \subset C^1(0, T, R^{2n+2}) \times C(0, T, R^{n+1}). \qquad (16)$$

Исследование разрешимости задачи оптимизации. Рассмотрим функционал качества. Подынтегральная функция при некоторых фиксированных вектор-функциях $x(\cdot)$ и $\dot{x}(\cdot)$ представляет собой функцию времени, т. е. $\sum Y(t)$. Для того чтобы функционал качества был ограничен достаточно, чтобы $\sum Y(t)$ была непрерывна и ограничена на $[0, T]$. Очевидно, что $\sum Y$ непрерывна по всем своим аргументам на множестве X_∂.

По теореме о суперпозиции непрерывных функций функция $\sum Y(t)$ является непрерывной. Согласно теореме Вейерштрасса функция $\sum Y(t)$ ограничена на $[0, T]$. Тогда по свойству определенного интеграла $\int_0^T \sum Y(x(t), \dot{x}(t))\, dt \geq AT$ (где $A = \inf_{t \in [0, T]} \sum Y(t)$). Отсюда, $B = \frac{1}{T} \cdot \int_0^T \sum Y(x(t), \dot{x}(t))\, dt \geq A > -\infty$.

Таким образом, функционал качества $B\left(xu,\dot{x}(\cdot),x(\cdot)\right)$ ограничен снизу. Значит задача (16) имеет смысл.

Очевидно, что правые части уравнений в системе (7) удовлетворяют условию Липшица и непрерывны по всем аргументам, следовательно, система (7) имеет решение $\forall u\in U_{\partial}$.

Рассмотрим множество допустимых пар $\Xi=\{(x,u)\in X_{\partial}\times U : \textit{удовлетворяет системе } (7)\}$.

1) Ограниченность Ξ.

– исходя из системы (7) $\forall(x,u)\in\Xi\ \exists C_1>$ такое, что выполняется условие $\|x\|_{C^1(0,T;R^{2n+2})}\le 1$;

– по условию, $\forall u\in U_{\partial}\ \exists C_2>0$ такое, что выполняется $\|u\|_{C(0,T;R^{n+1})}\le 2$;

– полагая $\|(x,u)\|_{X_{\partial}\times U}=\|x\|_{X_{\partial}}+\|u\|$, получаем, что множество допустимых пар (x,u) лежит в некотором шаре пространства $C^1(0,T;R^{2n+2})\times C(0,T;R^{n+1})$, $\Xi\subset B((0,0),C_1+C_2)$.

Таким образом, Ξ – ограничено.

2) Замкнутость Ξ.

Для доказательства замкнутости множества Ξ погружаем множество допустимых состояний в пространство Соболева $X_{\partial}\subset H^1(0,T;R^{2n+2})$, которое, как известно, является гильбертовым пространством $H^1(0,T;R^{2n+2})=\{x(\cdot)\in R^{2n+2} : x\in L^2(0,T;R^{2n+2}),\ x'\in L^2(0,T;R^{2n+2})\}$.

По теореме вложения Соболева имеем: $H^1(0,T;R^{2n+2})\subset C(0,T;R^{2n+2})$.

Таким образом, множество допустимых состояний является подмножеством пространства абсолютно непрерывных

функций $X_\partial \subset H^1(0,T;R^{2n+2}) \cap AC^2(0,T;R^{2n+2})$, где

$$AC^2(0,T;R^{2n+2}) = \{x(\cdot) \in R^{2n+2} : x \in C(0,T;R^{2n+2}), x' \in L^2(0,T;R^{2n+2})\}.$$

Аналогично, множество допустимых управлений погружаем в пространство суммируемых в квадрате функций $U_\partial \subset C(0,T;R^{n+1}) \cap L^2(0,T;R^1)$.

После релаксации введем множество допустимых управлений $\Xi \subset H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$, ограничения в множествах X_∂ и U_∂ выполняются почти всюду на $[0,T]$. Так (8) и (9) имеют вид

$$X_\partial = \left\{ x \in H^1(0,T;R^{2n+2}) : \alpha_{\min} \le x_i(t) \le \alpha_{\max} \text{ почти всюду на } [0,T], \right.$$
$$\left. \alpha_{\min} < 0, \alpha_{\max} > 0, \forall i \in [n+2, 2n+2] \right\},$$

$$U_\partial = \left\{ u_i \in L^2(0,T;R^{n+1}) : m_{\min} \le u_i \le m_{\max} \text{ почти всюду на } [0,T], \right.$$
$$\left. m_{\min} < 0, m_{\max} > 0, \forall i \in [1, n+1] \right\}.$$

Очевидно, что $\Xi \subset \Xi$ и

$$\inf_{(x,u) \in \Xi} B\left(x(\cdot), \dot{x}(\cdot)\right) \le \inf_{(x,u) \in \Xi} B\left(x(\cdot), \dot{x}(\cdot)\right).$$

Таким образом, задача оптимизации после релаксации имеет вид

$$\inf_{(x,u) \in \Xi} B\left(x(\cdot), \dot{x}(\cdot)\right), \quad \Xi \subset H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1). \qquad (17)$$

Замечание. Вследствие релаксации расширено множество допустимых процессов управления, поэтому оптимальная пара (x^0, u^0) задачи (17) может и не принадлежать исходному множеству Ξ.

Лемма 1. Множество $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1) \supset \Xi \ne \varnothing$.

Доказательство. Рассмотрим систему (7). Очевидно, что правые части $f_i(x,u)$ дифференциальных уравнений удовлетворяют условию Липшица $\|f(x,u)\|_{R^{2n+2}} \le N\|(x,u)\|_{R^2 \times R^2}$ и дифференцируемы по каждому из своих аргументов. По теореме о существовании и единственности реше-

ния дифференциальных уравнений получим, что $\forall u \in U_{\partial}$, множество $\Xi = \{(x,u) \in X_{\partial} \times U_{\partial} : \text{удовлетворяет системе } (7)\}$ не пусто. Лемма доказана.

Лемма 2. Множество Ξ - ограничено в $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$.

Доказательство. Исходя из системы (7), где равенство в уравнениях выполняются почти всюду на $[0,T]$, то

$$\|x(\cdot)\|_{H^1(0,T;R^{2n+2})} \le \sqrt{\int_0^T \int_0^T \|x(\cdot)\|^2_{R^{2n+2}} dt + \left\|\dot{x}(\cdot)\right\|^2_{R^{2n+2}} dt} \le \sqrt{C_1 + C_2} \le C_1'$$

По аналогии, $\forall u \in U_{\partial}$ $\exists C_2' > 0$ такое, что выполняется

$$\|u\|_{L_2(0,T;R^{n+1})} = \left(\int_0^T \int_0^T \|u\|^2_R dt\right)^{1/2} \le \left(m^2_{\max} dt\right)^{1/2} \le CT = C_2'.$$

Полагая $\|(x,u)\|_{X_{\partial} \times U_{\partial}} = \|x\|_{X_{\partial}} + \|u\|_{U_{\partial}}$, получаем, что множество допустимых пар (x,u) лежит в некотором шаре пространства $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$, $\Xi \subset B((0,0), C_1' + C_2')$.

Таким образом, Ξ - ограничено в $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$.

Лемма доказана.

Введем в пространстве $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$ топологию $\mu = \tau_{XU}$ τ, где τ_X – слабая топология в $H^1(0,T;R^{2n+2})$ и τ_U – слабая топология в $L^2(0,T;R^{n+1})$.

Поскольку множество Ξ является ограниченным множеством рефлексивного пространства $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$, то $\forall \varphi$ $H^1(0,T;R^{2n+2})$ выполняется

$$\lim_{n\to\infty}\int_0^T \left[\left(\varphi(t), x_{kk}(t)\right)_{R^{2n+2}} + \left(\varphi'(t), x'_{kk}(t)\right)_{R^{2n+2}}\right] dt$$

$$= \int_0^T \left[\left(\varphi(t), x_{00}(t)\right)_{R^{2n+2}} + \left(\varphi'(t), x'_{00}(t)\right)_{R^{2n+2}}\right] dt$$

Таким образом, из $x_k \xrightarrow{\tau_x} x_0$ и $H^1(0,T;R^{2n+2}) \subset C(0,T;R^{2n+2})$ следует, что $x_k \longrightarrow x_0$ в топологии равномерной сходимости.

Поскольку функции $x_k(t)$ и $\dot{x}_k(t)$ $\forall k \in [1,2,\ldots]$ ограничены почти всюду на $[0,T]$, то интегрант $Y\left(x_k(t), \dot{x}_k(t)\right)$ ограничен почти всюду на $[0,T]$. Поскольку функционал качества $B\left(x_k(\cdot), \dot{x}_k(\cdot)\right)$ ограничен снизу, то задача минимизации имеет смысл.

Лемма 3. Множество Ξ – μ-замкнуто в $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$.

Доказательство. Рассмотрим некоторую последовательность $(x_k, u_k) \subset H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$. Пусть (x_0, u_0) – ее μ-предел.

Поскольку равенства в уравнениях системы (7) выполняются почти всюду на $[0,T]$ и ее решение существует, а $x \in H^1(0,T;R^{2n+2}) \subset C(0,T;R^{2n+2})$, то правые части уравнений системы (7) $f_i(x,u)$ являются непрерывными по x, а поскольку $u \in L_2(0,T;R^{n+1})$, то $f_i(x_0, u_0) \in L_2(0,T;R^{2n+2})$. Значит $(x_0, u_0) \in \Xi$. Тогда, Ξ - μ-замкнуто в $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$.

Лемма доказана.

Лемма 4. Множество Ξ – секвенциально μ-компактно в $H^1(0,T;R^{2n+2}) \times L^2(0,T;R^1)$.

Доказательство. Пространство $H^1\left(0,T;R^{2n+2}\right)\times L^2\left(0,T;R^{1}\right)$ является рефлексивным банаховым пространством. В рефлексивном банаховом пространстве, как известно, ограниченное множество слабо компактно. Таким образом, из всякой последовательности $\left(x_k, u_k\right)\subset H^1\left(0,T;R^{2n+2}\right)\times L^2\left(0,T;R^{1}\right)$ можно выделить сходящуюся подпоследовательность $\left(x_{k_j}, u_{k_j}\right)\subset H^1\left(0,T;R^{2n+2}\right)\times L^2\left(0,T;R^{2}\right)$, такую, что $\forall(\varphi,\phi)\in H^1\left(0,T;R^{2n+2}\right)\times L^2\left(0,T;R^{1}\right)$

$$\lim_{j\to\infty}\int_0^T\left[\left(\varphi(t),x_{k_j}(t)\right)_{R^{2n+2}}+\left(\varphi'(t),x'_{k_j}(t)\right)_{R^{2n+2}}+\left(\phi(t),u_{k_j}(t)\right)_{R^{n+1}}\right]dt=$$

$$=\int_0^T\left[\left(\varphi(t),x_0(t)\right)_{R^{2n+2}}+\left(\varphi'(t),x'_0(t)\right)_{R^{2n+2}}+\left(\phi(t),u_0(t)\right)_{R^{n+1}}\right]dt.$$

Таким образом, с учетом лемм 2, 3 множество допустимых процессов управления является секвенциально μ-компактным.

Лемма доказана.

Для получения свойства μ-полунепрерывности снизу функционала качества $B\left(x(\cdot),\dot{x}(\cdot)\right)$ разделим его на сумму функционалов вида:

$$B_1=\sum_{i=1}^{n}\frac{m_i R_i}{T}\int_0^T x_{2n+2}^2\cos x_{1}\,dt,\ B_{21}=\sum_{i=1}^{n}\frac{m_i R_i}{T}\int_0^T \dot{x}_{2n+2}\sin x_{n+}\,dt,$$

$$B_{31}=\sum_{i=1}^{n}\frac{m_i}{T}\int_0^T x_{n+}^2\cos x_{n+1}\,dt,\ B_{41}=\sum_{i=1}^{n}\frac{m_i}{T}\int_0^T \dot{x}_{n+1}\sin x_{n+}\,dt,$$

$$B_{51}=\sum_{i=1}^{n}\frac{m_i R_i}{2T}\int_0^T \dot{x}_{2n+}\cdot\sin\left(x_{n+}+\lambda(i-1)\right)dt,$$

$$B_6=\sum_{i=1}^{n}\frac{m_i R_i}{2T}\int_0^T x_{2n+2}^2\cdot\cos\left(x_{1}+\lambda(i-1)\right)dt.$$

Рассмотрим функционал $B_1 = \sum_{i=1}^{n} \frac{m_i R}{T} \int_0^T x_{2n+2}^2 \cos x_{1} dt$. $\forall \{x_k\}_{k \in \mathbb{N}}$ $H^1(0,T; R^{2n+2})$, $\forall \{u_k\}_{k \in \mathbb{N}}$ $L^{21}(0,T; R^{n+})$ выполняется $x_k \xrightarrow{\tau_X} x^{**}$ в $H^1(0,T; R^{2n+2})$ и $u_k \xrightarrow{\tau_U} u^{**}$ в $L^{21}(0,T; R^{n+})$.

Условие полунепрерывности снизу имеет вид

$$\liminf_{k \to \infty} B_1(x_k, u_k) \geq B_1(x^{**}, u^{**}), \quad x_k \xrightarrow{\tau_X} x^{**}, \; u_k \xrightarrow{\tau_U} u^{**} \tag{18}$$

Поскольку $x_k \to x^{**}$ в $0,(CT)$ и $x_k(t)$ $\forall k$ ограничены, то $\int_0^T x_{kk}^2 \cdot \cos x dt \leq CT$.

По теореме Лебега о предельном переходе под знаком интеграла $\lim_{k \to \infty} \int_0^T x_k^{22} \cos x_k dt = \int_0^T \lim_{k \to \infty} x_k \cdot \cos\left(\lim_{k \to \infty} x_k\right) dt = \int_0^T x^{**2} \cdot \cos x dt$. Отсюда следует, что $\liminf_{k \to \infty} B_{11}(x_{kk}, u_k) = B_1(x^{**}, u^{**})$, т. е., условие (18) выполняется.

Аналогично доказывается, что функционалы B_3 и B_6 μ -полунепрерывны снизу.

Рассмотрим функционал $B_{21} = \sum_{i=1}^{n} \frac{m_i R}{T} \int_0^T \dot{x}_{2n+2} \sin x_{n+1} dt$. Представим его в виде

$$\sum_{i=1}^{n} \frac{m_i R}{T} \int_0^T \dot{x}_{2n+2} \cdot \sin x_{n+1} dt = C\dot{x}_{k\,2} \cdot \sin x^{*} dt +$$

$$+ C \int_0^T \dot{x}_{kk\,2n+2} \cdot \left[\sin x_{n+1} - \sin x^{*}\right] dt.$$

Обозначим второе слагаемое A.

Рассмотрим величину

$$|A| \leq 2C \int_0^T \left| \dot{x}_{k\,2n+2} \right| \cdot \left| \cos \frac{x_{n+1}^{kk} + x_{n+1}^{**}}{22} \right| \cdot \left| \sin \frac{x_{n+1} - x_{n+1}}{} \right| dt.$$

Поскольку величина $\left|\cos\frac{x^k_{n+1}+x^*_{n+1}}{2}\right|\le 1,\quad \left|\sin\frac{x^k_{n+1}-x^*_{n+1}}{2}\right|\to 0$ ($x^k_{n+1}\to x^*_{n+1}$ в $(0,T)$).

Следовательно,

$$|A|\le 2\int_0^T\int_0^T \left|\ddot{x}^k_{2n+2}\right|\cdot\left|\cos\frac{x^k_{n+1}+x^*_{n+1}}{2}\right|\cdot\left|\sin\frac{x^k_{n+1}-x^*_{n+1}}{2}\right|dt\le 2\cdot\left|x_{2n+2}\right|dt,$$

а $\int_0^T\left|\dot{x}^k_{2n+2}\right|dt<\infty$, т.е., $|A|\to 0$.

По определению слабой сходимости в $L^2(0,T;R^{2n+2})$ для $\cos x^*_{n+1}\in L^2(0,T;R^{2n+2})$ $\lim\limits_{k\to\infty}\int_0^T\int_0^T \ddot{x}^k_{n+1}\cos x^*_{n+1}dt=\ddot{x}^*_{n+1}\cdot\cos x^*_{n+1}dt$.

Значит $\liminf\limits_{k\to\infty} B_{21}(x_k,u)=B_{21}(x^*,u)$. Аналогично доказывается, что функционалы B_4 и B_5 μ-полунепрерывны снизу.

Следовательно, функционал $B(x,u)(\cdot),\dot{x}(\cdot))$ μ-полунепрерывен снизу как сумма μ-полунепрерывных снизу функционалов вида B_{16}.

Применяя теорему Вейерштрасса о том, что полунепрерывный снизу функционал на компактном ограниченном множестве достигает своей нижней границы, можно сделать следующий вывод: задача условной оптимизации $\inf\limits_{(x,u)\in\Xi} B(x,u)(\cdot),\dot{x}(\cdot))$ имеет непустое множество решений.

Библиографические ссылки

1. **Ландау Л.Д.** Теоретическая физика, т.1 Механика / Л.Д. Ландау, Е.М. Лифшиц – М, 1988.
2. **Иоффе А.Д.**Теория экстремальных задач / А.Д. Иоффе, В.М. Тихомиров. – М., 1974.
3. **Треногин В.А.** Функциональный анализ / В.А. Треногин – М., 1980.
4. **Васильев Ф.П.** Методы решения экстремальных задач/ Ф.П. Васильев–М., 1981.
5. **Колмогоров А.Н., Фомин С.В.** Элементы теории функций и функционального анализа / А.Н. Колмогоров, С.В. Фомин – М., 1976.

Надійшла до редкаолегії: 26.01.10

УДК 519.6+669.17.046.517В.001.24

©**Л.Т. Бойко*, В.П. Піптюк**, С.В. Греков**, М.П. Пасинков***
**Дніпропетровський національний університет імені Олеся Гончара*
***Інститут чорної металургії НАН України ім. З.І.Некрасова*

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ МАТЕРІАЛЬНОГО БАЛАНСУ ПРОЦЕСУ ПОЗАПІЧНОЇ ОБРОБКИ СТАЛІ ТА ЙОГО ПРОГРАМНА РЕАЛІЗАЦІЯ

Побудовано математичну модель матеріального балансу процесу позапічної обробки сталі за наявності установки ківш ‑піч. Запропоновано програмно реалізований алгоритм мінімізації відхилів рівнянь матеріального балансу та розглянуто конкретний приклад.

Построена математическая модель материального баланса процесса внепечной обработки стали при наличии установки ковш ‑печь. Предлагается программно реализованный алгоритм минимизации невязок уравнений материального баланса и рассмотрен конкретный пример.

A mathematical model of the material balance of the ladle treatment of steel in the presence of ladle furnace is made. A program implemented algorithm for minimizing the residuals of equations of material balance is offered and it is applied in the calculations on a specific example.

Ключові слова: позапічна обробка сталі, математичне моделювання, система нелінійних рівнянь у некоректній постановці, метод регуляризації Тихонова, комп'ютерна система.

Актуальність проблеми. Позапічна обробки сталі є багатофакторним виробничим процесом, одним із важливих компонентів якого є баланс матеріалів на початку та по завершенні процесу. Матеріальний баланс є невід'ємною характеристикою процесу і будується виходячи із законів збереження маси.

Визначення параметрів процесу для приведення відхилів рівнянь матеріального балансу позапічної обробки сталі до відповідної точності є актуальною задачею, яка є важливою для вдосконалення автоматичного та оперативного керування важливим етапом виробництва сталі. Розбіжності балансу матеріалів можуть бути обумовлені такими факторами, як

похибка вимірювань, невідповідність звітних даних реаліям виробництва, надмірне використання добавок або нераціональні умови тощо. Технологам виробництва важливо мати більш точні дані щодо процесу для прийняття рішень по оптимізації параметрів доводки розплаву в ковші до марочного хімічного складу сталі, регламентованого нормативно-технічною документацією.

Для технологів є, зокрема, дуже важливою узгодженість даних за схемою «вхід-вихід» для побудови адекватних моделей перебігу процесу. Під час здійснення позапічної обробки сталі хімічний склад розплаву поступово змінюється, що зумовлено введенням до нього добавок. Дуже важливим є визначення необхідної кількості добавок, що вводяться в розплав. Однак, через те, що відсутня інформація про їх розчинення на окремих стадіях обробки, зашумленість даних вимірювання та низку інших обставин, визначення цієї кількості є дуже складною задачею і базується здебільшого на досвіді людей, що безпосередньо керують процесом. Для того, щоб стало можливим збільшити долю автоматичного керування, потрібно розробити комп'ютерну програму, яка дозволить, мінімізуючи відхили матеріального балансу, зменшити зашумленість значень параметрів процесу.

Виробнича постановка задачі. Процес позапічної обробки сталі є етапом більш загального процесу – виробництва сталі. Цей етап у технологічній схемі виробництва знаходиться між етапами виплавки напівпродукту (первинної речовини металу, що має рідкий стан, високу температуру та половинний хімічний склад, порівняно з вимогами нормативно-технічної документації) та розливу металу. Безпосередньо процес позапічної обробки необхідний, зокрема, для доведення хімічного складу розплаву до регламентованих норм марки сталі. Позапічна обробка, у свою чергу, поділяється на кілька стадій: випуск металу в ківш, обробка на установці ківш-піч (УКП), вакуумування металу (за наявності обладнання). Етап розливу металу є завершальним («виходом») у процесі виробництва сталі. Під час випуску в ківш та обробки на УКП до розплаву вводяться добавки з метою розкислення та рафінування металу. Розкислення – операція по зменшенню вмісту кисню в металі, здійснюється за допомогою кремній- та марганецьмістких феросплавів, а також речовин, що містять алюміній та кальцій. Рафінування – операція по зменшенню вмісту елементів, що погіршують якість сталі (наприклад, сірка, фосфор, азот [1]), яка здійснюється шляхом обробки металу шлаками, сформованими із різних шлакоутворюючих добавок.

Дані про хімічний склад деяких добавок відомі наближено, через похибку вимірювального обладнання. Крім того, маси добавок, введених до розплаву під час процесу, також, у багатьох випадках, відомі наближено через, знову таки, похибку вимірювань і через людський фактор (певна кількість добавок вводиться вручну).

У ковші знаходяться великі маси розплавленого металу та шлаку. Технологічно важливими є знання маси та хімічного складу цих речовин в моменти переходу від одної стадії до іншої. Такі дані не завжди наявні. За наявності ці дані містять похибку. В той же час вони є важливими для визначення кількості добавок, що необхідно ввести додатково та/або забезпечення вилучення з ковша маси шлаку (для запобігання зворотних реакцій при задовільному хімічному складі металу).

На стадії випуску до ковша добавки розчиняються в розплаві за рахунок його перемішування під дією струменя металу, що витікає з плавильного агрегата. Під час обробки на УКП такий рух організується штучно, завдяки продувці розплаву аргоном.

Рівняння матеріального балансу складаються в моменти переходу від випуску в ківш до початку обробки на УКП, а потім від першої стадії обробки на УКП до другої. Схема побудови рівняння є такою: маса компонента в шлаку, що відповідає хімічному елементу, за яким будується рівняння, в кінці наступної стадії має збігатись із сумою маси компонента в шлаку на початку поточної стадії та сумою мас компонента, що утворилися в результаті введення добавок, металева складова балансу враховується при визначенні маси хімічного елемента, що утворює компонент шлаку. Якщо відбувається зворотна реакція та деяка маса компонента відновлюється до елемента і маса останнього переходить із шлаку в метал, частина балансового рівняння, що відповідає металевій складовій збільшується на вивільнену масу. Деякі елементи можуть згоряти. Під масою елемента в речовині розуміємо масу, яка не згоріла під час її розчинення в розплаві. Маса елемента в добавці є частиною маси добавки. При підрахунку маси елемента в добавці враховується лише та частина маси добавки, яка розчинилася в розплаві.

Параметрами процесу є маси та хімічний склад добавок, маса і хімічний склад металу та шлаку на початку процесу і в моменти переходу від однієї стадії до іншої, коефіцієнти розчинення добавок на окремих стадіях, коефіцієнти згоряння хімічних елементів. Параметри процесу можуть бути відомими точно, відомими з похибкою або відомими, виходячи із даних по подібних плавках (також із похибкою).

Відомими з похибкою є маси всіх речовин, наявних у процесі, хімічні склади металу, шлаку та окремих добавок. Для всіх параметрів, відомих із

похибкою, відомі інтервали, в межах яких можуть знаходитись значення цих параметрів. Початкове наближення для цих параметрів задається по даних, отриманих під час вимірювань у ході виробничого процесу.

Коефіцієнти згоряння елементів та розчинення добавок є, взагалі кажучи, невідомими параметрами, проте із виробничих реалій відомі інтервали зміни значень кожного такого коефіцієнта. За початкове наближення параметрів приймаються середини відповідних інтервалів.

Рівняння матеріального балансу будуються для таких елементів та компонентів: вуглець (C), марганець (Mn), кремній (Si), сірка (S), фосфор (P), залізо (Fe), алюміній (Al), магній (Mg), кальцій (Ca), FeO, CaF_2, а також за домішками. Балансове рівняння за хімічним елементом враховує баланс елемента в металі та відповідного йому компонента у шлаку. Як було зазначено, будуються дві системи матеріального балансу під час переходу між стадіями позапічної обробки сталі. Рівняння для CaF_2 будується лише в рамках другої системи. Системи є пов'язаними між собою, а тому можуть бути об'єднані в єдину систему двадцяти трьох балансових рівнянь .

Окрім рівнянь матеріального балансу до системи математичної моделі також включаються додаткові обмеження. Вони необхідні для того, щоб дольовий вміст елементів та компонентів у хімічному складі речовини за сумою давав одиницю. Разом із додатковими обмеженнями система рівнянь математичної моделі має 35 рівнянь і 143 невідомих.

Система рівнянь математичної моделі є нелінійною, оскільки до неї входять нелінійні алгебраїчні рівняння. Максимальний порядок нелінійностей у рівняннях моделі дорівнює чотирьом.

Очевидно, що математична модель описує фізичний процес наближено, а отже, і розв'язок математичної задачі будемо шукати наближено.

Задача полягає в тому, щоб знайти такий вектор параметрів виробничого процесу, при якому відхили рівнянь матеріального балансу будуть найменшими.

Метод розв'язування математичної моделі. Систему нелінійних алгебраїчних рівнянь (СНР), що моделює матеріальний баланс, будемо розв'язувати ітераційним методом [2]. Для цього запишемо СНР у вигляді

$$f(x) = \theta, \quad x \in G, \tag{1}$$

де позначено

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_n \end{bmatrix}, \quad f(x) \quad \begin{bmatrix} f_1(x_1, x_2, \quad , x_n) \\ f_2(x_1, x_2, \quad , x_n) \\ \\ f_m(x_1, x_2, \quad , x_n) \end{bmatrix}.$$

Тут $f_i(x_1, x_2, \ldots, x_n)$, $i = \overline{1,m}$ – відомі неперервні функції своїх аргументів, що мають неперервні похідні в області G; m, взагалі кажучи, не дорівнює n. Кожна з цих функцій відповідає окремому рівнянню матеріального балансу. Компоненти шуканого вектора x відповідають тим параметрам процесу позапічної обробки сталі, які необхідно уточнити. Область G визначається обмеженнями, накладеними на аргументи вектора x. Ці обмеження відомі з фізичної постановки задачі.

Незалежно від того має СНР (1) в області G єдиний розв'язок, має їх безліч або не має жодного, будемо шукати такий вектор $x^* \in G$, що

$$\inf_{x \in G} \| f(x) \| = \| f(x^*) \|. \qquad (2)$$

Якщо СНР (1) має не один вектор x^*, що задовольняє умову (2), то позначимо множину цих векторів $G_1 \subset G$. У цьому випадку вибираємо такий вектор $x^* \in G_1$, що є найближчим до заданого вектора $x^{(0)} \in G$, тобто

$$\inf_{x \in G_1} \| x - x^{(0)} \| = \| x^* - x^{(0)} \|.$$

Вектор $x^* \in G_1$ шукаємо ітераційним методом, взявши за нульове наближення заданий вектор $x^{(0)}$. Припустивши, що нам відоме p-е наближення, шуканий вектор представляємо у вигляді

$$x^* = x^{(p)} + \varepsilon^{(p)}, \qquad (3)$$

де $\varepsilon^{(p)}$ – вектор поправок до наближення $x^{(p)}$.

Підставимо (3) в (1) та ліву частину отриманої рівності розкладемо в ряд Тейлора в околі точки $x^{(p)}$, обмежившись тільки лінійними членами розкладання

$$f(x^{(p)}+\varepsilon^{(p)}) = f(x^{(p)}) + A(x^{(p)})\varepsilon^{(p)} + \ldots = 0,$$

де $A(x^{(p)}) = \left[\frac{\partial f_i}{\partial x_j}\right]_{x=x^{(p)}}$ $i=\overline{1,m}, j=\overline{1,n}$ – прямокутна матриця.

Приходимо до системи лінійних алгебраїчних рівнянь (СЛАР)

$$A(x^{(p)})\varepsilon^{(p)} = -f(x^{(p)}). \qquad (4)$$

СЛАР (4) розв'язуємо методом регуляризації Тихонова [3] та знаходимо наближено вектор $\varepsilon^{(p)}$. Наступну ітерацію обчислюємо за формулою

$$x^{(p+1)} = x^{(p)} + \varepsilon^{(p)}. \qquad (5)$$

Ітераційний процес зупиняємо, якщо виконана задана кількість ітерацій, або якщо виконується умова

$$\left\|\varepsilon^{(p)}\right\| \le \varepsilon,$$

де ε – задане додатне число.

Виходячи із особливостей поставленої задачі, до критеріїв виходу також додаються малість норми вектора відхилів рівнянь системи матеріального балансу та малість змін норми вектора відхилів протягом декількох послідовних ітерацій.

Для зручності позначимо вектор $-f(x^{(p)})$ як u, а вектор $\varepsilon^{(p)}$ як z, тоді систему (4) можна переписати у вигляді

$$Az = u. \qquad (6)$$

Враховуючи що матриця A та вектор u відомі наближено, будемо шукати наближення до нормального розв'язку СЛАР (6), тобто будемо шукати такий вектор z, що мінімізує функціонал [3]

$$M^{\alpha}[z,u,A] = \|Az-u\|^2 + \alpha\|z\|^2,$$

а параметр α визначати із умови

$$\|Az-u\| \le 2(h\|z\| + \delta) + \mu \quad . \qquad (7)$$

Тут $\mu = \inf_{z \in R^n} \|Az - u\|$; додатні сталі h, δ характеризують похибку, з якою відома матриця A та вектор u, відповідно.

Зауважимо, що якщо система (6) має розв'язки, то $\mu = 0$, і тоді умова (7) замінюється умовою

$$\|Az-u\| \le h\|z\| + \delta \; .$$

Отже, приходимо до такого алгоритму пошуку наближеного розв'язку СНР (1):

1. Вхідними даними є наближені рівняння матеріального балансу (1), область G, $x^{(0)}$ – нульове наближення шуканого вектора (відомий з виробничої постановки задачі), додатне число ε.
2. Вважаємо відомим наближення $x^{(p)}$ (при $p = 0$ воно відомо).
3. Відповідно до СЛАР (6) та (4) обраховуємо матрицю A та вектор u.
4. Розв'язуємо СЛАР (6), знаходимо вектор z, який є вектором шуканих поправок $\varepsilon^{(p)}$. За формулою (5) уточнюємо наближене значення шуканого вектора x.
5. Якщо виконується хоча б одна з таких трьох умов: $\|z\| < \varepsilon$ або $D_k < \varepsilon$, або кількість ітерацій більша за допустиму, то переходимо на п. 6, інакше на п. 3.
6. Виводимо уточнений розв'язок x СНР (1).

У цьому алгоритмі D_k – середній відхил норми вектора відхилів від середнього значення за задану кількість k кроків. Малість величини вказує на недоцільність подальших уточнень, оскільки суттєвих змін норми вектора відхилів не відбувається. Рекомендації до застосування подібного критерію виходу із циклу знайдено в [4]. Рекомендації застосування в якості критерію виходу із ітераційного процесу заздалегідь зазначеної кількості ітерацій містяться в [5].

Комп'ютерна система та її тестування. Підходи до розробки математичної моделі матеріального балансу позапічної обробки сталі обговорювалися в [6]. Однак виявилося, що задача складання балансових рівнянь та їх лінеаризація є дуже трудомісткою при ручному виконанні.

Враховуючи, що математична модель складається з великої кількості рівнянь та великої кількості невідомих, крім того, в реальному виробничому процесі позапічної обробки сталі можливі ситуації, при яких може виникнути необхідність модифікувати побудовану математичну модель, актуальною стала задача автоматизації процесу побудови балансових рівнянь з подальшою їх лінеаризацією. Для цього була розроблена комп'ютерна програма, яка не тільки будувала систему нелінійних балансових рівнянь, враховувала всі обмеження, розв'язувала СНР, але і мала зручний, зрозумілий для виробничників інтерфейс.

Матриця лінеаризованої системи рівнянь у реальних прикладах має декілька тисяч елементів. Вона може мати лінійно залежні рядки або стовпчики. СЛАР з такою матрицею може бути несумісною, або мати безліч розв'язків. Крім того, СНР може мати безліч розв'язків або не мати жодного. Тестування програми на всіх можливих ситуаціях підтвердило збіжність ітераційного алгоритму розв'язування СНР до очікуваних результатів при будь-яких варіантах вибору нульового наближення.

Час розрахунків, як по тестових прикладах, так і по реальних даних, вимірюється хвилинами. Проте в окремих випадках реальних задач може досягати годин (залежно від кількості параметрів конкретного виробничого процесу, потужності апаратної бази та кількості ітерацій, що буде виконано протягом ітераційного процесу). Тестування системи проводилось на апаратній базі з процесором Intel Core 2 Duo CPU E8400 та 4 Гб оперативної пам'яті.

Реальний приклад та аналіз результатів. Розроблений програмний продукт було застосовано для розрахунків за реальними даними плавок сталі марки 3сп, які відбувалися на ВАТ «Єнакієвський металургійний завод». Для прикладу наведено параметри реальної плавки.

На етапі випуску металу в ківш додаються такі добавки: феросиліцій ФС65 (400 кг), силікомарганець МнС17 (850 кг), антрацит (100 кг), алюмінієві чушки АВ87 (46 кг), вапно (500 кг).

Під час етапу обробки на УКП вводяться: силікомарганець МнС17 (100 кг), феросиліцій ФС65 (150 кг), силікокальцієвий дріт СК30 (80 кг), кокс (126 кг), вапно (1100 кг), плавиковий шпат (370 кг), алюмофлюс АК-45 (200 кг).

Масу конверторного шлаку прийнято за 400 кг. Масу шлаку на межі останньої стадії першого етапу та першої стадії другого – за 1000 кг, кінцеву масу шлаку – за 1600 кг. Допуск по масі шлаку – 1000 кг.

Масу металу в усі три ключові моменти процесу (початки кожної із трьох стадій) визначено із середніх значень питомої маси у відповідні моменти. Допуск по масі металу – п'ять тонн.

Маси та хімічні склади добавок взяті із наданої інформації за конкретною плавкою. Допуски за процентним вмістом окремих елементів (компонентів) сформовано виходячи із наданих даних по можливих відхилах. Допуск по масі – 20 кг.

Слід окремо відмітити футеровку ковша, як складову процеса. Окремо вносяться дані по футеровці шлакового пояса та футеровці стін та дна ковша, що знаходиться під металом. Хімічний склад є сталим і варіюватись може лише маса. Допуск за масою футеровки шлакового пояса – 3 кг, футеровки стін та дна – 17 кг. Загальна маса футеровки ковша, що руйнується під час процесу, – 200 кг.

Хімічні склади речовин, які можуть уточнюватися в ході розрахунків, наведено в табл. 1, 2, 3.

Табл. 1, 2 складені в результаті виконаного аналізу стандартними методами (хімічним чи спектральним) взятих проб речовин (металу чи шлаку). У перших стовпчиках цих таблиць (а також табл. 4, 5) номери I відповідають тим пробам, що відбиралися на останній стадії випуску в ківш; номери II відповідають пробам, що відбиралися на першій стадії обробки на УКП; номери III відповідають пробам, відібраним на другій стадії обробки на УКП.

Стадії вказані в табл. 3 (аналогічно і в табл. 6, 7, 8) відповідають внесенню добавок на випуску металу в ківш (I) та на УКП (II).

Таблиця 1

Хімічний склад металу на різних стадіях виробництва сталі, %

Номер проби	C	Mn	S	P	Si	Al	Cu	Cr	Ni	Fe
I	0,060	0,070	0,062	0,025	**0,050**	Н.а.	**Н.а.**	**Н.а.**	**Н.а.**	**99,733**
II	0,090	0,440	0,050	0,025	0,150	0,001	**0,010**	**0,010**	**0,010**	**99,214**
III	0,190	0,540	0,015	0,024	0,170	0,050	0,030	0,070	**0,010**	**98,265**

Н. а. – не аналізувався

Таблиця 2

Хімічний склад шлаку на різних стадіях виробництва сталі, %

Номер проби	CaO	SiO_2	MgO	MnO	Al_2O_3	P_2O_5	S^*	CaF_2	FeO	Домішки
I	50,01	22,45	4,75	3,93	1,69	1,33	0,12	0,23	15,38	**0,11**
II	45,56	23,98	5,80	2,42	2,61	0,70	0,60	3,81	14,00	**0,52**
III	41,11	25,51	6,85	0,90	3,52	0,07	1,00	7,40	12,63	**1,01**

∗ – вміст S в компоненті CaS

Таблиця 3

Хімічний склад феросплавів, %

Речовина	Стадія	C	Mn	S	P	Si	Fe	Al
FeSi	**I**	0,070	0,050	0,010	0,035	**67,000**	**32,814**	**0,021**
SiMn	**I**	1,500	71,000	0,010	0,050	**17,000**	**11,040**	**Д.в.**
FeSi	**II**	0,070	0,050	0,010	0,035	**67,000**	**32,814**	**0,021**
SiMn	**II**	1,500	71,000	0,010	0,050	**17,000**	**11,040**	**Д.в.**

Д. в. – дані відсутні

Уточнений хімічний склад речовин (після розв'язання СНР) наведено в табл. 4, 5, 6.

Таблиця 4

Хімічний склад металу на різних стадіях виробництва сталі (уточнений), %

Номер проби	C	Mn	S	P	Si	Al	Cu	Cr	Ni	Fe
I	0,080	0,085	0,040	**0,029**	0,000	Н.а.	**Н.а.**	**Н.а.**	**Н.а.**	**98,670**
II	0,147	0,379	0,035	0,028	0,130	0,007	**0,048**	**0,048**	**0,048**	**99,020**
III	0,230	0,498	0,023	0,019	0,195	0,005	0,040	0,070	**0,060**	**98,550**

Н. а. – не аналізувався

Таблиця 5

Хімічний склад шлаку на різних стадіях виробництва сталі (уточнений), %

Номер проби	CaO	SiO_2	MgO	MnO	Al_2O_3	P_2O_5	S*	CaF_2	FeO	Домішки
I	45,00	24,95	4,00	4,16	1,92	1,63	0,20	0,77	15,78	**0,00**
II	41,11	25,09	4,75	3,53	3,52	1,32	1,29	4,93	15,12	**2,36**
III	42,00	25,00	8,00	0,82	3,32	0,05	0,50	7,00	11,60	**0,00**

∗ – аналогічно табл. 2

Таблиця 6

Хімічний склад феросплавів (уточнений), %

Речовина	Стадія	C	Mn	S	P	Si	Fe	Al
FeSi	**I**	0,055	0,044	0,005	0,019	**66,748**	**33,290**	**0,010**
SiMn	**I**	1,450	70,690	0,438	0,106	**17,600**	**10,330**	**Д.в.**
FeSi	**II**	0,090	0,093	0,050	0,046	**67,240**	**30,630**	**1,747**
SiMn	**II**	1,500	71,000	0,010	0,051	**16,990**	**10,440**	**Д.в.**

Д. в. – дані відсутні.

Враховуються коефіцієнти згоряння для всіх компонентів матеріального балансу. Коефіцієнти розчинення на етапі випуску в ківш уточнюються для таких речовин: феросиліцій, силікомарганець, антрацит, алюмінієві чушки.

Далі наведені маси речовин після коригування . На випуску в ківш: метал – 153070 кг, шлак – 333,301 кг, феросиліцій ФС65 – 400 кг (розчиняється на 95,383 %), силікомарганець МнС17 – 850,385 кг (розчиняється на 80,48 %), антрацит – 99,2879 кг (розчиняється на 90 %), алюмінієві чушки АВ87 – 45,6506 кг (розчиняються на 95,015 %), вапно – 479,999 кг. При обробці на УКП: метал – 152669 кг, шлак – 970,836 кг, феросиліцій ФС65– 149,992 кг, силікомарганець МнС17– 99,997 кг, силікокальцієвий дріт СК30– 73,5806 кг, кокс – 125,012 кг, вапно – 1080, плавиковий шпат – 390 , алюмофлюс АК-45 – 198,552 кг, футеровка під

металом – 180,642 кг, футеровка шлакового пояса – 33,2433 кг. Після обробки на УКП: метал – 153400 кг, шлак – 1644 кг.

Уточнені коефіцієнти згоряння елементів наведено в табл. 7. До розв'язання СНР (1) були відомі лише проміжки, в яких ці коефіцієнти можуть знаходитися.

Таблиця 7

Згоряння елементів (після уточнення)

Елемент	Етап	Коефіцієнт згоряння
C	I	0.05000
Mg	I	0.85081
Si	I	0.10014
Ca	I	0.85000
Mn	I	0.04002
S	I	0.30000
P	I	0.70000
Fe	I	0.00002
Al	I	0.50001
C	II	0.05000
Mn	II	0.02001
Si	II	0.10010
S	II	0.30000
P	II	0.70000
Fe	II	0.00001
Al	II	0.60000
Mg	II	0.85009
Ca	II	0.95000

Абсолютні значення майже всіх відхилів зменшуються (значення відхилів рівнянь матеріального балансу наведено в табл. 8).

Ті відхили, які не зменшуються, зберігають порядок чисел. Збільшення значень відхилів по Mg, Al і P, вірогідно можна пояснити непоказністю та обмеженістю проб шлаку, відібраних на різних стадіях позапічної обробки сталі, що, у свою чергу, пов'язано з ліквацiйною рухомістю відповідних компонентів шлаку. Тому відмічене потребує додаткового уточнення та перевірки.

Як видно з табл. 8, відхили по залізу зменшуються на декілька порядків. Це елемент, початкові відхили рівнянь якого є найбільшими і відрізняються на порядок від усіх інших, а отже норма вектора відхилів у результаті також зменшується на порядок.

Таблиця 8

Порівняльна таблиця відхилів рівнянь до та після розрахунків

Компонент матеріального балансу, за яким побудовано рівняння	**Етап**	**Відхил рівняння до коригування**	**Відхил рівняння після коригування**
C	**I**	0,26936	−0,10693
Mg	**I**	0,01791	0,04722
Si	**I**	1,41247	0,92939
Ca	**I**	1,18560	1,11012
Домішки	**I**	0,17919	0,05530
Mn	**I**	−0,90410	−0,00055
S	**I**	0,08966	−0,00907
P	**I**	−0,00341	−0,00028
Fe	**I**	11,06580	−0,00051
Al	**I**	0,30833	−0,00053
C	**II**	−1,03696	0,00009
Mn	**II**	−0,31530	0,12148
Si	**II**	2,83954	1,62206
S	**II**	0,25061	0,15522
P	**II**	−0,01047	0,47297
Fe	**II**	6,39364	0,30390
Al	**II**	1,41412	1,51246
Mg	**II**	0,55713	0,28384
Ca	**II**	0,26936	−0,10693
Домішки	**II**	0,01791	0,04722

Отриманий результат є допустимим, оскільки всі вхідні дані є обмеженими допустимими інтервалами значень і значення вектора

параметрів на кожній ітерації проектується на допустиму область. У ході ітераційного процесу уточнено ряд параметрів виробничого процесу, в результаті чого суттєво зменшено відхили матеріального балансу. Це означає, що уточнені дані можуть бути використані для подальшого моделювання виробничого процесу.

Висновок. Розроблена математична модель та її комп'ютерна реалізація дають можливість уточнити значення параметрів виробничого процесу позапічної обробки сталі, при цьому відхили рівнянь матеріального балансу зменшуються у середньому на порядок. Це дозволить у подальшому використати розроблену комп'ютерну систему для вирішення актуальної задачі автоматизації керування виробничим процесом.

Бібліографічні посилання

1. **Дюдкин Д.А.**, Производство стали на агрегате ковш-печь. / Д.А.Дюдкин, С.Ю. Бать, С.Е. Гринберг, С.Н. Маринцев; под науч. ред. докт. техн. наук, проф. Дюдкина Д.А. — Донецк, 2003. — 300 с.
2. **Бойко Л.Т.** Застосування методу Тихонова до розв'язування систем нелінійних рівнянь у некоректній постановці «Проблеми прикладної математики та комп'ютерних наук» / Л.Т. Бойко // Тези доповідей тематичної наукової конференції за підсумками науково-дослідної роботи Дніпропетровського національного університету за 2007–2008 роки. — Д., 2008.— С. 81.
3. **Тихонов А.Н.** Методы решения некорректных задач / А.Н. Тихонов, В.Я. Арсенин. – М., 1979. – 288 с.
4. **Гетьман В.Б.** Моделювання та регуляризація задачі визначення розподілу частинок за розмірами в дисперсних середовищах оптичним методом: автореф. дис. канд. техн. наук: 01.05.02 [Електронний ресурс] / В.Б. Гетьман; Нац. ун-т «Львів. Політехніка». — Л., 2003. — 16 с.: рис. — укр.
5. http://www.nbuv.gov.ua/portal/natural/Mm/2009_1/section%202/s-2-14.pdf
6. **Бойко Л.Т**. Математичне моделювання процесу позапічної обробки сталі в частині його матеріального балансу / Л.Т. Бойко, В.П Піптюк, С.В. Греков, М.П. Пасинков // Шоста міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем». Тези доповідей. 12–14 листопада 2008р. Дніпропетровськ. С. 51.

Надійшла до редколегії 12.05.10

УДК 631.3:637.1

©**Ю.В. Версаль**
Кіровоградський національний технічний університет

ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ ДІАГНОСТИКИ ФІЗІОЛОГІЧНОГО СТАНУ ЛАКТУЮЧИХ КОРІВ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Побудовано математичну модель діагностики фізіологічного стану лактуючих корів у вигляді системи нечітких логічних рівнянь, дерево логічного висновку, що визначає ієрархічний зв'язок параметрів тварини з діагнозом та функції належності нечітких термів вхідних змінних . Створено шість нечітких баз знань. Наведено приклад роботи моделі.

Построено математическую модель диагностики физиологического состояния лактирующих коров в виде системы нечетких логических уравнений, дерево логического вывода, определяющее иерархическую связь параметров животного с диагнозом и функции принадлежности нечетких термов входных переменных. Создано шесть нечетких баз знаний. Приведен пример работы модели.

In article the mathematical model of lactating cows physiological state diagnostics as fuzzy logic equations system, the logic conclusion tree determining hierarchical communication of animal parameters with the diagnosis and entrance variables fuzzy terms membership functions are constructed. Six fuzzy knowledge bases are created. The example of model work is resulted.

Ключові слова: фізіологічний стан, математична модель, нечітка логіка, функція належності, нечітка база знань

Вступ. Інтенсифікація молочного виробництва, ріст її ефективності та поліпшення якості продукції неможливі без неперервного автоматичного контролю за якістю молока у процесі його отримання та станом здоров'я тварин. Тому діагностика фізіологічного стану лактуючих корів, зокрема маститу та статевої охоти, є однією з найбільш важливих проблем у молочному господарстві.

За даними Міжнародної молочної федерації, клінічно виражений мастит зареєстровано майже у 2 % корів, субклінічний – 50 % [1]. Продуктивність при ураженій частці вимені становить 69,6 % очікуваного надою, від кожної хворої на мастит корови за добу втрачається у середнь-

ому 0,7–0,9 л молока. Окрім того, знижується кількість молочного жиру на 0,37–0,4 %, сухої речовини на 0,31–0,58 % [1].

Вибір оптимального строку осіменіння корів і телиць є одним із основних факторів у роботі з відтворення стада. Плодотворне осіменіння досягається тільки у періоди статевої охоти, яка наступає і повторюється через визначений період після отелення корів і після наступання статевої зрілості телиць. У 60–70 % випадків охота проявляється вночі і зранку і триває в середньому 12–18 годин [2].

Діагностиці фізіологічного стану лактуючих корів присвячено велику кількість досліджень [3]. Найбільш розповсюдженими є методи, побудовані на вимірюванні електропровідності та температури молока [1; 3 – 6]. В останні роки зарубіжними вченими проведено дослідження можливості діагностики фізіологічного стану тварин на основі змін надою молока [7], результати яких є досить перспективними. Недоліком існуючих методів є те, що вони не пристосовані до роботи з якісними (нечисловими) і нечіткими знаннями, а саме такі евристичні або інтуїтивні знання частіш за все використовуються ветеринарами для постановки діагнозу і мають форму, доступну для спеціалістів даної області.

Постановка задачі. Основою математичної обробки нечислової (лінгвістичної) інформації є теорія нечітких множин [8]. Тому задача даного дослідження – побудова математичної моделі діагностики фізіологічного стану лактуючих корів породи «чорна ряба» на основі нечіткої логіки.

Метод розв'язування. Розв’язання поставленої задачі передбачає ідентифікацію залежності входи (параметри тварини) – вихід (діагноз) [8]. З математичної точки зору, – це відображення типу (1):

$$X = \{x_1, x_2, ..., x_n\} \rightarrow d \in D = \{d_1, d_2, ..., d_m\}, n=10, m=3, \qquad (1)$$

де X – множина параметрів тварини; D – множина діагнозів.

Ієрархічний зв’язок параметрів тварини $x_1 ... x_{10}$ з діагнозом D представлено деревом логічного висновку (рис. 1), яке визначає структуру діагностичної моделі. Особливість моделі полягає в тому, що вона враховує як традиційні для діагностики фізіологічного стану лактуючих корів значення температури молока з різних долей вимені, так і додатково введений показник разового надою молока.

Вершини дерева інтерпретуються таким чином: корінь дерева – діагноз; термінальні вершини – частинні параметри стану; нетермінальні вершини (подвійні кола) – згортка частинних параметрів стану в укрупнені.

Дуги, що виходять з нетермінальних вершин дерева, відповідають укрупненим параметрам стану.

Множину діагнозів позначимо наступним чином:

- d_1 – тварина здорова;
- d_2 – у тварини стан статевої охоти;
- d_3 – тварина хвора маститом.

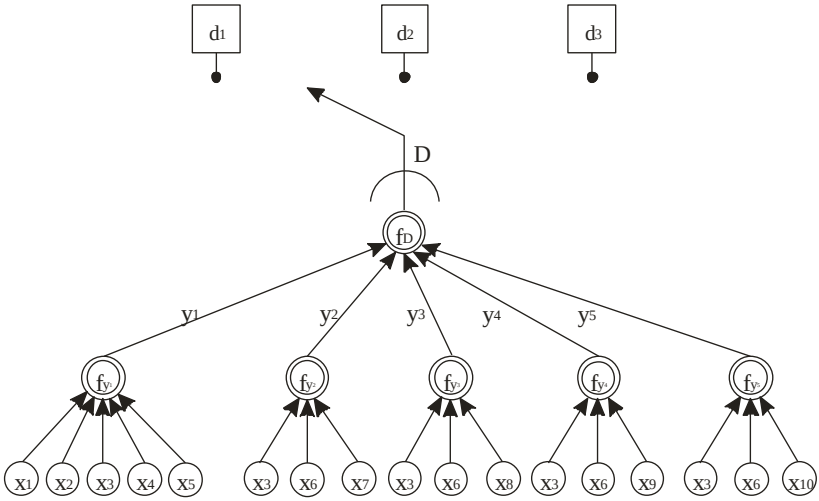


Рис. 1. Дерево логічного висновку

Наведеному на рисунку 1 дереву логічного висновку відповідає система співвідношень (2) – (7):

$$D = f_D(y_1, y_2, y_3, y_4, y_5), \tag{2}$$

$$y_1 = f_{y_1}(x_1, x_2, x_3, x_4, x_5), \tag{3}$$

$$y_2 = f_{y_2}(x_3, x_6, x_7), \tag{4}$$

$$y_3 = f_{y_3}(x_3, x_6, x_8), \tag{5}$$

$$y_4 = f_{y_4}(x_3, x_6, x_9), \tag{6}$$

$$y_5 = f_{y_5}(x_3, x_6, x_{10}), \tag{7}$$

де $f(\bullet)$ – функціональний зв'язок між вхідними та вихідними змінними.

Практичне застосування теорії нечітких множин передбачає представлення параметрів тварини у вигляді лінгвістичних змінних. Формалізація лінгвістичних значень у рамках теорії нечітких множин здійснюється через функції належності. Формалізацію частинних та укрупнених параметрів наведено в таблицях 1 та 2 відповідно.

Кожен з термів, якими оцінюються вхідні лінгвістичні змінні формалізовано нечіткими множинами на відповідному універсумі за допомогою трикутної функції належності що описується формулою (8). Її графік зображено на рисунку 2 [9].

$$\mu^{jp}(x_i) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}, \tag{8}$$

де: a, b і c – параметри настройки, які мають наступну інтерпретацію:

(a, c) – носій нечіткої множини;

b – координата максимуму (рисунок 2).

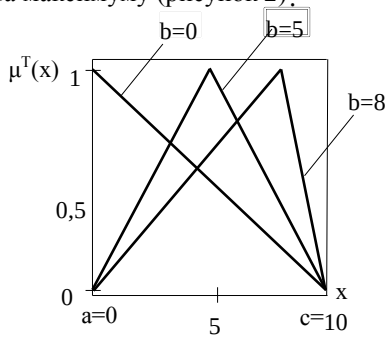


Рис. 2. Модель функції належності

Функції належності побудовано за методом, викладеним у [12]. Вони отримані на основі наступної вихідної інформації:

- діапазон $[\underline{x_i}, \overline{x_i}]$ зміни значень параметра x_i, $i = \overline{1, n}$;
- кількість термів, що використовуються для лінгвістичної оцінки параметра x_i.

Таблиця 1

Формалізація частинних параметрів стану тварини лінгвістичними змінними

Позначення і назва частинного параметра	Універсум	Терми для лінгвістичної оцінки
x_1 – номер лактації	$1 \div 3$ у.о.	перша (І); друга (ІІ); третя і старше (ІІІ).
x_2 – тривалість лактації	$1 \div 9$ у.о.	перший етап (І); другий (ІІ); третій (ІІІ); четвертий (ІV); п'ятий (V); шостий (VI); сьомий (VII); восьмий (VIII); дев'ятий (IX).
x_3 – час доби	$1 \div 2$ у.о.	ранок (Р); вечір (В).
x_4 – температура зовнішнього повітря	$0 \div +35°$ С	нормальна (Н); жарка (Ж); дуже жарка (дЖ).
x_5 – разовий надій молока	$1{,}5 \div 10{,}9$ л	дуже низький (дН); низький (Н); набагато нижче середнього (ннС); нижче середнього (нС); середній (С); вище середнього (вС); набагато вище середнього (нвС); високий (В); дуже високий (дВ).
x_6 – температура навколишнього повітря	$0 \div +35°$ С	нормальна (Н); вище нормальної (вН); жарка (Ж); дуже жарка (дЖ).
x_7, x_8, x_9, x_{10} – температура молока з різних долей вимені	$+38{,}5 \div +41{,}6°$ С	низька (Н); нижче середньої (нС); середня (С); вище середньої (вС); висока (В); дуже висока (дВ).

Таблиця 2

Формалізація укрупнених параметрів стану тварини лінгвістичними змінними

Позначення і назва укрупненого параметра	Терми для лінгвістичної оцінки
y_1 – продуктивність	нормальна (Н); знижена (зН).
y_2, y_3, y_4, y_5 – температурні показники по різним долям вимені	нормальний (Н); високий (В).

За визначенням [10] функція належності $\mu^{jp}(x_i)$ характеризує суб'єктивну міру (в діапазоні $[0, 1]$) впевненості експерта в тому, що чітке значення x_i відповідає нечіткому терму з номером jp. Кількість термів для лінгвістичної оцінки вхідних параметрів x_n, обираємо з діапазону 2÷9 [11]. Параметри функцій належності нечітких термів, що відповідають частинним параметрам тварини (таблиця 1), наведені в таблиці 3, графіки зображені на рисунку 3.

Таблиця 3

Параметри функцій належності нечітких термів

Параметр стану тварини		x_1			x_2								
Терми		I	II	III	I	II	III	IV	V	VI	VII	VIII	IX
Параметри функцій належності	a	−1	1	1	−1	1	1	1	1	1	1	1	1
	c	3	3	4	9	9	9	9	9	9	9	9	12
	b	1	2	3	1	2	3	4	5	6	7	8	9

Продовження таблиці 3

Параметри функцій належності нечітких термів

Параметр тварини		x_3		x_4			x_5							
Терми		Р	В	Н	Ж	дЖ	дН	Н	ннС	нС	С	вС	нвС	В
Параметри функцій належності	a	−1	1	−5	0	0	0	1.5	1.5	1.5	1.5	1.5	1.5	1.5
	c	2	3	35	35	45	10.9	10.9	10.9	10.9	10.9	10.9	10.9	10.9
	b	1	2	13	29	35	1.5	2.5	3	3.8	4.8	6.4	7.85	9.4

Продовження таблиці 3

Параметри функцій належності нечітких термів

Параметр стану тварини		x_5	x_6				x_7, x_8, x_9, x_{10}					
Терми		дВ	Н	вН	Ж	дЖ	Н	нС	С	вС	В	дВ
Параметри функцій належності	a	1,5	−5	0	0	0	38	38,5	38,5	38,5	38,5	38,5
	c	13	35	35	35	45	41,6	41,6	41,6	41,6	41,6	42
	b	10,9	4,5	23,5	28	35	38,5	38,7	38,9	39,1	39,4	39,7

У нечітких діагностичних системах установлення діагнозу здійснюється шляхом логічного висновку за нечіткою базою знань (або матриці знань), що являє собою сукупність лінгвістичних знань –правил типу [8]:

$$\text{Якщо } \bigcup_{p=1}^{k_j}\Big[\bigcap_{i=1}^{n}(x_i = X_i^{jp})\Big], \text{ то } y = d_j,\ j = \overline{1,m}, \qquad (9)$$

де d_j – нечіткий терм для оцінки j-го рівня вихідної змінної D; X_i^{jp} – нечіткий терм для оцінки вхідної змінної x_i в p-му рядку матриці знань, що відповідає терму d_j, $p = \overline{1,k_j}$; k_j – кількість рядків, що відповідають терму d_j; $\bigcup(\bigcap)$ – символ операції АБО (І).

У правилах нечіткої бази знань сконцентровані досвід експерта та його розуміння зв'язків «входи–вихід». Упевненість експерта в достовірності правила відображає ваговий коефіцієнт a_{jp}, що приймає значення з інтервалу [0…1]. Нечіткі бази знань за багатовимірними залежностями «параметри тварини – діагноз», що відповідають співвідношенням (2)–(7), наведені в таблицях 4–6, загальна кількість правил дорівнює 401. Кожний рядок цих таблиць відповідає одному правилу типу «Якщо–тоді». Зв'язок лінгвістичних змінних у межах правила здійснюється логічною операцією ТА, правил у межах бази знань – логічною операцією АБО.

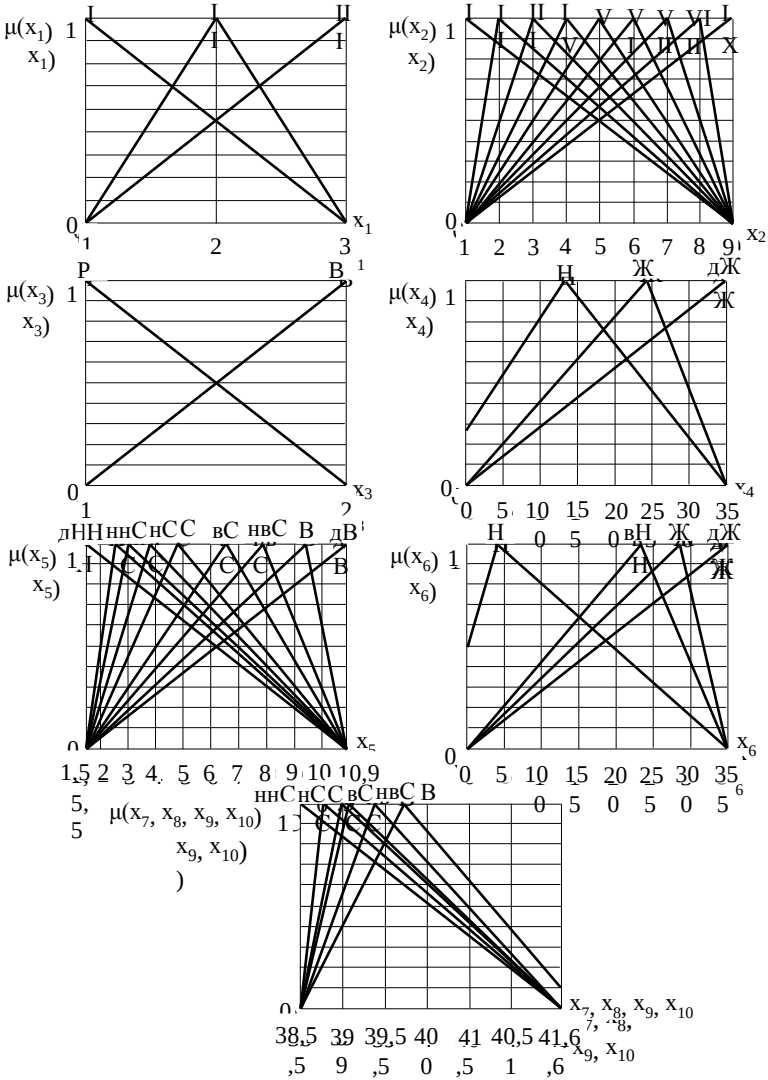


Рис. 3. Графіки функцій належності нечітких термів

Таблиця 4

База знань про співвідношення (2)

y_1	y_2	y_3	y_4	y_5	a_{jp}	D
Н	Н	Н	Н	Н	1	d_1
Н	В	В	В	В	1	d_2
зН	В	В	В	В	1	
	В	Н	Н	Н	1	d_3
	Н	В	Н	Н	1	
	Н	Н	В	Н	1	
	Н	Н	Н	В	1	
	В	В	Н	Н	1	
	В	Н	В	Н	1	
	В	Н	Н	В	1	
	Н	В	В	Н	1	
	Н	В	Н	В	1	
	Н	Н	В	В	1	

Таблиця 5

Витяг з бази знань про співвідношення (3)

x_1	x_2	x_3	x_4	x_5	a_{jp}	y_1
I	II	В	Ж	С	0.5	Н
	III	Р	дЖ	С	0.5	
II	IV	В	Н	вС	0.5	
	V	Р	Ж	вС	0.5	
	VI	В	дЖ	ннС	0.5	
III	VII	Р	Н	нвС	0.5	
	VIII	В	Ж	С	0.5	
I	IX	В	Н	Н	0.5	зН
	VIII	Р	Ж	нС	0.5	
	VII	В	дЖ	дН	0.5	
II	VI	Р	Н	вС	0.5	
	IV	Р	дЖ	С	0.5	
III	III	В	Н	С	0.5	
	II	Р	Ж	нвС	0.5	
	I	В	дЖ	нС	0.5	

Таблиця 6

База знань про співвідношення (4) - (7)

x_3	x_6	$x_7\ x_8\ x_9\ x_{10}$	a_{jp}	$y_2\ y_3\ y_4\ y_5$
Р	Н	Н	1	Н
	вН	нС	1	
	Ж	С	1	
	дЖ	вС	1	
В	Н	Н	1	
	вН	нС	1	
	Ж	вС	1	
	дЖ	В	1	
Р	Н	нС	1	В
	вН	С	1	
	Ж	вС	1	
	дЖ	В	1	
В	Н	нС	1	
	вН	С	1	
	Ж	В	1	
	дЖ	дВ	1	

Математичною моделлю діагностики фізіологічного стану лактуючих корів є система нечітких логічних рівнянь типу (10), що відповідають системі висловлювань (9) [9]:

$$\mu^{d_j}(D) = \max_{p=\overline{1,k_j}} [a_{jp} \cdot \min_{i=\overline{1,n}} \mu^{jp}(x_i)],\ j = \overline{1,m}, \qquad (10)$$

де a_{jp} – вага експертного правила з номером jp ; $\mu^{d_j}(D)$ і $\mu^{jp}(x_i)$ – функції належності змінних y та x_i до термів d_j і X_i^{jp} відповідно.

Замінивши операції min і max операціями $\wedge$ і $\vee$ відповідно, за базою знань з таблиці 4 записуємо нечіткі логічні рівняння (11)–(13) для визначення діагнозу, що відповідають співвідношенню (2):

$$\mu^{d_1}(D) = \mu^{H}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{H}(y_5), \qquad (11)$$

$$\mu^{d_2}(D) = \mu^{H}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{B}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{B}(y_5), \quad (12)$$

$$\mu^{d_3}(D) = \mu^{зН}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{B}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{B}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{B}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{H}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{B}(y_3) \wedge \mu^{H}(y_4) \wedge \mu^{B}(y_5) \vee \\ \vee \mu^{зН}(y_1) \wedge \mu^{H}(y_2) \wedge \mu^{H}(y_3) \wedge \mu^{B}(y_4) \wedge \mu^{B}(y_5) \quad (13)$$

За базою знань з таблиці 5 записуємо нечіткі логічні рівняння (14)– (15) для визначення продуктивності, що відповідають співвідношенню (3):

$$\mu^{H}(y_1) = \mu^{I}(x_1) \wedge \mu^{II}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{Ж}(x_4) \wedge \mu^{C}(x_5) \vee \\ \vee \mu^{I}(x_1) \wedge \mu^{III}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{дЖ}(x_4) \wedge \mu^{C}(x_5) \vee \\ \vee \mu^{II}(x_1) \wedge \mu^{IV}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{H}(x_4) \wedge \mu^{вС}(x_5) \vee \\ \vee \mu^{II}(x_1) \wedge \mu^{V}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{Ж}(x_4) \wedge \mu^{вС}(x_5) \vee \\ \vee \mu^{II}(x_1) \wedge \mu^{VI}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{дЖ}(x_4) \wedge \mu^{ннС}(x_5) \vee \\ \vee \mu^{III}(x_1) \wedge \mu^{VII}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{H}(x_4) \wedge \mu^{нвС}(x_5) \vee \\ \vee \mu^{III}(x_1) \wedge \mu^{VIII}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{Ж}(x_4) \wedge \mu^{C}(x_5) \vee \quad (14)$$

$$\mu^{зН}(y_1) = \mu^{I}(x_1) \wedge \mu^{IX}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{H}(x_4) \wedge \mu^{H}(x_5) \vee \\ \vee \mu^{I}(x_1) \wedge \mu^{VIII}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{Ж}(x_4) \wedge \mu^{нС}(x_5) \vee \\ \vee \mu^{I}(x_1) \wedge \mu^{VII}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{дЖ}(x_4) \wedge \mu^{дН}(x_5) \vee \\ \vee \mu^{II}(x_1) \wedge \mu^{VI}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{H}(x_4) \wedge \mu^{вС}(x_5) \vee \\ \vee \mu^{II}(x_1) \wedge \mu^{IV}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{дЖ}(x_4) \wedge \mu^{C}(x_5) \vee \\ \vee \mu^{III}(x_1) \wedge \mu^{III}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{H}(x_4) \wedge \mu^{C}(x_5) \vee \\ \vee \mu^{III}(x_1) \wedge \mu^{II}(x_2) \wedge \mu^{P}(x_3) \wedge \mu^{Ж}(x_4) \wedge \mu^{нвС}(x_5) \vee \\ \vee \mu^{III}(x_1) \wedge \mu^{I}(x_2) \wedge \mu^{B}(x_3) \wedge \mu^{дЖ}(x_4) \wedge \mu^{нС}(x_5) \quad (15)$$

За базою знань з таблиці 6 записуємо нечіткі логічні рівняння (1 6)–(17) для визначення температурних показників по різним долям вимені, що відповідають співвідношенням (4)–(7):

$$\begin{aligned}&\mu^{H}(y_2)=\mu^{P}(x_3)\wedge\mu^{H}(x_6)\wedge\mu^{H}(x_7)\vee\\&\vee\mu^{P}(x_3)\wedge\mu^{вH}(x_6)\wedge\mu^{нC}(x_7)\vee\mu^{P}(x_3)\wedge\mu^{Ж}(x_6)\wedge\mu^{C}(x_7)\vee\\&\vee\mu^{P}(x_3)\wedge\mu^{дЖ}(x_6)\wedge\mu^{вC}(x_7)\vee\mu^{B}(x_3)\wedge\mu^{H}(x_6)\wedge\mu^{H}(x_7)\vee\\&\vee\mu^{B}(x_3)\wedge\mu^{вH}(x_6)\wedge\mu^{нC}(x_7)\vee\mu^{B}(x_3)\wedge\mu^{Ж}(x_6)\wedge\mu^{вC}(x_7)\vee\\&\vee\mu^{B}(x_3)\wedge\mu^{дЖ}(x_6)\wedge\mu^{B}(x_7)\end{aligned}\qquad(16)$$

$$\begin{aligned}&\mu^{B}(y_2)=\mu^{P}(x_3)\wedge\mu^{H}(x_6)\wedge\mu^{нC}(x_7)\vee\\&\vee\mu^{P}(x_3)\wedge\mu^{вH}(x_6)\wedge\mu^{C}(x_7)\vee\mu^{P}(x_3)\wedge\mu^{Ж}(x_6)\wedge\mu^{вC}(x_7)\vee\\&\vee\mu^{P}(x_3)\wedge\mu^{дЖ}(x_6)\wedge\mu^{B}(x_7)\vee\mu^{B}(x_3)\wedge\mu^{H}(x_6)\wedge\mu^{нC}(x_7)\vee\\&\vee\mu^{B}(x_3)\wedge\mu^{вH}(x_6)\wedge\mu^{C}(x_7)\vee\mu^{B}(x_3)\wedge\mu^{Ж}(x_6)\wedge\mu^{B}(x_7)\vee\\&\vee\mu^{B}(x_3)\wedge\mu^{дЖ}(x_6)\wedge\mu^{дB}(x_7)\end{aligned}\qquad(17)$$

Прийняття рішення з визначення фізіологічного стану тварини проводиться за алгоритмом в наступній послідовності:

Крок 1. Визначаємо значення частинних парметрів тварини $x_1 ... x_{10}$.

Крок 2. Знаходимо ступені належності частинних параметрів тварини лінгвістичним термам з таблиці 1.

Крок 3. Підставляємо знайдені на кроці 2 ступені належності в нечіткі логічні рівняння (1 4)–(15) та обчислюємо ступені належності змінної y_1 до термів «Н», «зН» відповідно.

Крок 4. Підставляємо знайдені на кроці 2 ступені належності в нечіткі логічні рівняння (1 6)–(17) та обчислюємо ступені належності змінних y_2, y_3, y_4, y_5 до термів «Н», «В» відповідно.

Крок 5. Підставляємо знайдені на кроках 3 та 4 ступені належності в нечіткі логічні рівняння (11)–(13) та обчислюємо ступені належності діагнозу до термів d_1, d_2, d_3 відповідно.

Крок 6. В якості фізіологічного стану тварини обираємо терм із множини $\{d_1, d_2, d_3\}$ з максимальним ступенем належності.

Аналіз одержаних результатів. Проілюструємо використання запропонованих моделей та алгоритму на прикладі визначення фізіологічного стану корови з ідентифікаційним номером UA2100079714 породи «чорна

ряба» четвертої лактації (за даними Державного підприємства «Дослідне господарство імені 9 Січня Полтавського інституту агропромислового виробництва імені М.І. Вавілова Української академії аграрних наук»). Значення частинних параметрів тварини наведені у першій графі таблиці 7.

За формулою трикутної функції належності (8) та даними таблиці 3 знаходимо ступені належності значень частинних параметрів тварини нечітким термам та зводимо їх до таблиці 7.

Таблиця 7

Ступені належності частинних параметрів тварини

Значення частинних параметрів тварини	Ступені належності частинних параметрів тварини нечітким термам
x_1=3 у.о.	$\mu^{I}(x_1)=0$; $\mu^{II}(x_1)=0$; $\mu^{III}(x_1)=1$
x_2=6 у.о.	$\mu^{I}(x_2)=0$; $\mu^{II}(x_2)=0$; $\mu^{III}(x_2)=0$; $\mu^{IV}(x_2)=0$; $\mu^{V}(x_2)=0$; $\mu^{VI}(x_2)=1$; $\mu^{VII}(x_2)=0$; $\mu^{VIII}(x_2)=0$; $\mu^{IX}(x_2)=0$
x_3=2 у.о.	$\mu^{Р}(x_3)=0$; $\mu^{В}(x_3)=1$
x_4=5 °С	$\mu^{Н}(x_4)=0{,}43$; $\mu^{Ж}(x_4)=0{,}2$; $\mu^{дЖ}(x_4)=0{,}14$;
x_5=3 л	$\mu^{дН}(x_5)=0{,}84$; $\mu^{Н}(x_5)=0{,}95$; $\mu^{ннС}(x_5)=1$; $\mu^{нС}(x_5)=0{,}63$; $\mu^{С}(x_5)=0{,}55$; $\mu^{вС}(x_5)=0{,}29$; $\mu^{нвС}(x_5)=0{,}24$; $\mu^{В}(x_5)=0{,}19$; $\mu^{дВ}(x_5)=0{,}16$
x_6=5 °С	$\mu^{Н}(x_6)=0{,}97$; $\mu^{вН}(x_6)=0{,}21$; $\mu^{Ж}(x_6)=0{,}17$; $\mu^{дЖ}(x_6)=0{,}14$
x_7=40,3 °С	$\mu^{Н}(x_7)=0{,}42$; $\mu^{нС}(x_7)=0{,}46$; $\mu^{С}(x_7)=0{,}49$; $\mu^{вС}(x_7)=0{,}51$; $\mu^{В}(x_7)=0{,}57$; $\mu^{дВ}(x_7)=0{,}71$
x_8=38,5 °С	$\mu^{Н}(x_8)=1$; $\mu^{нС}(x_8)=0$; $\mu^{С}(x_8)=0$; $\mu^{вС}(x_8)=0$; $\mu^{В}(x_8)=0$; $\mu^{дВ}(x_8)=0$
x_9=38,5 °С	$\mu^{Н}(x_9)=1$; $\mu^{нС}(x_9)=0$; $\mu^{С}(x_9)=0$; $\mu^{вС}(x_9)=0$; $\mu^{В}(x_9)=0$; $\mu^{дВ}(x_9)=0$
x_{10}=40,3 °С	$\mu^{Н}(x_7)=0{,}42$; $\mu^{нС}(x_7)=0{,}46$; $\mu^{С}(x_7)=0{,}49$; $\mu^{вС}(x_7)=0{,}51$; $\mu^{В}(x_7)=0{,}57$; $\mu^{дВ}(x_7)=0{,}71$

Підставляючи знайдені ступені належності в нечіткі логічні рівняння (14)-(15), що визначають продуктивність тварини, знаходимо:

$$\mu^{H}(y_1) = 0 \wedge 0 \wedge 1 \wedge 0{,}2 \wedge 0{,}55 \vee$$
$$\vee 0 \wedge 0 \wedge 0 \wedge 0{,}14 \wedge 0{,}55 \vee 0 \wedge 0 \wedge 1 \wedge 0{,}43 \wedge 0{,}29 \vee 0 \wedge 0 \wedge 0 \wedge 0{,}2 \wedge 0{,}29 \vee$$
$$\vee 0 \wedge 1 \wedge 1 \wedge 0{,}14 \wedge 1 \vee 1 \wedge 0 \wedge 0 \wedge 0{,}43 \wedge 0{,}24 \vee 1 \wedge 0 \wedge 1 \wedge 0{,}2 \wedge 0{,}55 \vee$$
$$\vee \ldots = 0{,}29$$

$$\mu^{зH}(y_1) = 0 \wedge 0 \wedge 1 \wedge 0{,}43 \wedge 0{,}95 \vee 0 \wedge 0 \wedge 0 \wedge 0{,}2 \wedge 0{,}63 \vee$$
$$\vee 0 \wedge 0 \wedge 1 \wedge 0{,}14 \wedge 0{,}84 \vee 0 \wedge 1 \wedge 0 \wedge 0{,}43 \wedge 0{,}29 \vee 0 \wedge 0 \wedge 0 \wedge 0{,}14 \wedge 0{,}55 \vee$$
$$1 \wedge 0 \wedge 1 \wedge 0{,}43 \wedge 0{,}55 \vee 1 \wedge 0 \wedge 0 \wedge 0{,}2 \wedge 0{,}24 \vee 1 \wedge 0 \wedge 1 \wedge 0{,}14 \wedge 0{,}63 \vee$$
$$\vee \ldots = 0{,}43$$

Підставляючи знайдені ступені належності в нечіткі логічні рівняння (16)-(17), що визначають температурний показник тварини, знаходимо:

$$\mu^{H}(y_3) = 0 \wedge 0{,}97 \wedge 0{,}42 \vee 0 \wedge 0{,}21 \wedge 0{,}46 \vee 0 \wedge 0{,}17 \wedge 0{,}49 \vee 0 \wedge 0{,}14 \wedge 0{,}51 \vee$$
$$1 \wedge 0{,}97 \wedge 0{,}42 \vee 1 \wedge 0{,}21 \wedge 0{,}46 \vee 1 \wedge 0{,}17 \wedge 0{,}51 \vee 1 \wedge 0{,}14 \wedge 0{,}57 = 0{,}42$$

$$\mu^{B}(y_2) = 0 \wedge 0{,}97 \wedge 0{,}46 \vee 0 \wedge 0{,}21 \wedge 0{,}49 \vee 0 \wedge 0{,}17 \wedge 0{,}51 \vee$$
$$\vee 0 \wedge 0{,}14 \wedge 0{,}57 \vee 1 \wedge 0{,}97 \wedge 0{,}46 \vee 1 \wedge 0{,}21 \wedge 0{,}49 \vee 1 \wedge 0{,}17 \wedge 0{,}57 \vee$$
$$\vee 1 \wedge 0{,}14 \wedge 0{,}71 = 0{,}46$$

$$\mu^{H}(y_3) = 0 \wedge 0{,}97 \wedge 1 \vee 0 \wedge 0{,}21 \wedge 0 \vee 0 \wedge 0{,}17 \wedge 0 \vee 0 \wedge 0{,}14 \wedge 0 \vee$$
$$1 \wedge 0{,}97 \wedge 1 \vee 1 \wedge 0{,}21 \wedge 0 \vee 1 \wedge 0{,}17 \wedge 0 \vee 1 \wedge 0{,}14 \wedge 0 = 0{,}97$$

$$\mu^{B}(y_3) = 0 \wedge 0{,}97 \wedge 0 \vee 0 \wedge 0{,}21 \wedge 0 \vee 0 \wedge 0{,}17 \wedge 0 \vee 0 \wedge 0{,}14 \wedge 0 \vee$$
$$1 \wedge 0{,}97 \wedge 0 \vee 1 \wedge 0{,}21 \wedge 0 \vee 1 \wedge 0{,}17 \wedge 0 \vee 1 \wedge 0{,}14 \wedge 0 = 0$$

$$\mu^{H}(y_4) = 0 \wedge 0{,}97 \wedge 1 \vee 0 \wedge 0{,}21 \wedge 0 \vee 0 \wedge 0{,}17 \wedge 0 \vee 0 \wedge 0{,}14 \wedge 0 \vee$$
$$1 \wedge 0{,}97 \wedge 1 \vee 1 \wedge 0{,}21 \wedge 0 \vee 1 \wedge 0{,}17 \wedge 0 \vee 1 \wedge 0{,}14 \wedge 0 = 0{,}97$$

$$\mu^{B}(y_4) = 0 \wedge 0{,}97 \wedge 0 \vee 0 \wedge 0{,}21 \wedge 0 \vee 0 \wedge 0{,}17 \wedge 0 \vee 0 \wedge 0{,}14 \wedge 0 \vee$$
$$1 \wedge 0{,}97 \wedge 0 \vee 1 \wedge 0{,}21 \wedge 0 \vee 1 \wedge 0{,}17 \wedge 0 \vee 1 \wedge 0{,}14 \wedge 0 = 0$$

$$\mu^{H}(y_5) = 0 \wedge 0{,}97 \wedge 0{,}42 \vee 0 \wedge 0{,}21 \wedge 0{,}46 \vee 0 \wedge 0{,}17 \wedge 0{,}49 \vee$$
$$\vee 0 \wedge 0{,}14 \wedge 0{,}51 \vee 1 \wedge 0{,}97 \wedge 0{,}42 \vee 1 \wedge 0{,}21 \wedge 0{,}46 \vee 1 \wedge 0{,}17 \wedge 0{,}51 \vee$$
$$\vee 1 \wedge 0{,}14 \wedge 0{,}57 = 0{,}42$$

$$\mu^{B}(y_5) = 0 \wedge 0{,}97 \wedge 0{,}46 \vee 0 \wedge 0{,}21 \wedge 0{,}49 \vee 0 \wedge 0{,}17 \wedge 0{,}51 \vee$$
$$0 \wedge 0{,}14 \wedge 0{,}57 \vee 1 \wedge 0{,}97 \wedge 0{,}46 \vee 1 \wedge 0{,}21 \wedge 0{,}49 \vee 1 \wedge 0{,}17 \wedge 0{,}57 \vee$$
$$\vee 1 \wedge 0{,}14 \wedge 0{,}71 = 0{,}46$$

Підставляючи знайдені ступені належності до нечітких логічних рівнянь (11)-(13), що визначають діагноз, знаходимо:

$$\mu^{d_1}(D) = 0{,}29 \wedge 0{,}42 \wedge 0{,}97 \wedge 0{,}97 \wedge 0{,}42 = 0{,}29$$

$$\mu^{d_2}(D) = 0{,}29 \wedge 0{,}46 \wedge 0 \wedge 0 \wedge 0{,}46 \vee 0{,}43 \wedge 0{,}46 \wedge 0 \wedge 0 \wedge 0{,}46 = 0\,,$$

$$\mu^{d_3}(D) = 0{,}43 \wedge 0{,}46 \wedge 0{,}97 \wedge 0{,}97 \wedge 0{,}42 \vee 0{,}43 \wedge 0{,}42 \wedge 0 \wedge 0{,}97 \wedge 0{,}42 \vee$$
$$0{,}43 \wedge 0{,}42 \wedge 0{,}97 \wedge 0 \wedge 0{,}42 \vee 0{,}43 \wedge 0{,}42 \wedge 0{,}97 \wedge 0{,}97 \wedge 0{,}46 \vee$$
$$\vee 0{,}43 \wedge 0{,}46 \wedge 0 \wedge 0{,}97 \wedge 0{,}42 \vee 0{,}43 \wedge 0{,}46 \wedge 0{,}97 \wedge 0 \wedge 0{,}42 \vee$$
$$0{,}43 \wedge 0{,}46 \wedge 0{,}97 \wedge 0{,}97 \wedge 0{,}46 \vee 0{,}43 \wedge 0{,}42 \wedge 0 \wedge 0 \wedge 0{,}42 \vee$$
$$\vee 0{,}43 \wedge 0{,}42 \wedge 0 \wedge 0{,}97 \wedge 0{,}46 \vee 0{,}43 \wedge 0{,}42 \wedge 0{,}97 \wedge 0 \wedge 0{,}46 = 0{,}43$$

Таким чином, у результаті логічного висновку отримана така нечітка множина

$$\tilde{D} = \left(\frac{0{,}29}{d_1}, \frac{0}{d_2}, \frac{0{,}43}{d_3}\right).$$

Найбільший ступінь $0{,}43$ має рішення d_3, тому в якості фізіологічного стану тварини обираємо «хвора маститом». Зазначимо, що поставлений діагноз збігається з дійсним фізіологічним станом тварини.

Висновки. В даній статті запропоновано математичну модель діагностики фізіологічного стану лактуючих корів породи чорна ряба, зокрема маститу і стану статевої охоти, на основі нечіткої логіки. Вона являє собою систему нечітких логічних рівнянь, записаних по шести нечітким базам знань, правила яких шляхом логічного висновку за ієрархічною структурою визначають продуктивність, температурні показники з різних долей вимені та діагноз тварини. Наведено алгоритм діагностики та продемонстровано його виконання на конкретному прикладі.

Переваги даної моделі в тому, що вона працює з якісними (нечисловими) і нечіткими знаннями, які частіш за все використовуються ветеринарами для постановки діагнозу і мають форму, доступну для спеціалістів даної області. Подальший розвиток моделі може здійснюватися за напрямком розповсюдження на інші породи великої рогатої худоби.

Запропонована модель може бути використана у системах діагностики фізіологічного стану тварин при автоматизованих доїльних установках всіх типів у сільськогосподарських підприємствах усіх форм власності.

Найбільший ступінь $0{,}43$ має рішення d_3, тому в якості фізіологічного стану тварини обираємо «хвора маститом». Зазначимо, що поставлений діагноз збігається з дійсним фізіологічним станом тварини.

Бібліографічні посилання

1. **Пешук Л.** Електропровідність молока як метод виявлення прихованих маститів у корів / Л. Пешук // Пропозиція. – 2001. – № 7.
2. http://agroua.net/advisory/usefuladvices/index.php?qid=9
3. **Луценко М.** Прилади для контролю якості виконання технологічних процесів у тваринництві / М. Луценко, В. Смоляр // Техніка АПК. – 2005. – № 5–6. – 32–34.
4. **Gil Z.** Milk temperature fluctuations during milking in cows with subclinical mastitis // Livestock Production Science. – 1988. – № 20. – P. 223–231.
5. **Fordham D.P.** Oestrus detection in dairy cows by milk temperature measurement / Fordham D.P., Rowlinson P., McCarthy T.T. // Res Vet Science. – 1988. – № 44(3). – P. 366–374.
6. **Firk R.** Systematic effects on activity,milk yield, molk flow rate and electrical conductivity / Firk R., Stamer E., Junge W., Krieter J. // Arch. Tierz. – 2002. – № 3. – P. 213–222.
7. **Gröhn Y. T.** Effect of Pathogen-Specific Clinical Mastitis on Milk Yield in Dairy Cows / Gröhn Y. T., Wilson D. J., González R. N., Hertl J. A., Schulte H., Bennett G., Schukken Y. H. // Journal of Dairy Science. – 2004. – № 87. – P. 3358–3374.
8. **Ротштейн А.П.** Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. / А.П. Ротштейн – Винница, 1999. – 320 с.
9. **Панкевич О.Д.** Діагностування тріщин будівельних конструкцій за допомогою нечітких баз знань / Панкевич О.Д., Штовба С.Д. // Вінниця, 2005. – 108 с.
10. **Заде Л.** Понятие лингвистической переменной и ее применение к принятию приближенных решений. /Л.Заде – М., 1976. – 167 с.
11. **Miller C. A.** The Magic Number Seven Plus or Minus Two: Some limits on our Capacity for Proceeding Information // Psychological Review. – 1956. – № 64. – P.81–97.
12. **Ротштейн А.П.** Медицинская диагностика на нечеткой логике. / А.П. Ротштейн – Винница, 1996. – 132 с.

Надійшла до редколегії 25.11.09

УДК 519.8

©**Л.Л. Гарт, Н.В. Поляков**
Днепропетровский национальный университет имени Олеся Гончара

ПРИМЕНЕНИЕ ПРОЕКЦИОННО_ИТЕРАЦИОННОГО МЕТОДА К РЕШЕНИЮ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ КОЛЕБАТЕЛЬНЫМИ ПРОЦЕССАМИ

Досліджується питання про застосування проекційно _ітераційного методу, заснованого на методі умовного градієнту, до розв'язування задач оптимального керування гіперболічними системами з квадратичним функціоналом. Проводиться порівняльний аналіз цього методу і звичайного проекційного методу на прикладі розв'язання конкретної задачі.

Исследуется вопрос о применении проекционно_итерационного метода, основанного на методе условного градиента, к решению задач оптимального управления гиперболическими системами с квадратичным функционалом. Проводится сравнительный анализ этого метода и обычного проекционного метода на примере решения конкретной задачи.

The problem of applying the projection-iteration method based on the conditional gradient method to solving problems of optimal control of hyperbolic systems with a quadratic functional is investigated. The comparative analysis of this method and the ordinary projection method for solving the concrete problem is carried out.

Ключевые слова: оптимальное управление, проекционно_итерационный метод, разностная схема, метод условного градиента, последовательность, приближенное решение.

Введение. Задачи оптимального управления колебательными процессами имеют многочисленные приложения – к ним, например, приводят задачи об успокоении качки судна или стрелы подъемного крана, о работе вибротранспортеров, об организации виброзащиты, амортизации и т. п. [1]. Все вышеупомянутые задачи можно трактовать как экстремальные, в

подходящим образом выбранных функциональных пространствах и для исследования этих задач использовать аппарат и методы функционального анализа. Такая трактовка позволяет выявлять общие закономерности, присущие широким классам экстремальных задач, создавать и исследовать общие методы решения таких задач [1]. Существует большое количество приближенных методов решения задач оптимального управления, однако их наличие не исключает возможности создания новых, более эффективных методов и усовершенствования существующих.

В данной работе, которая является продолжением и обобщением [2], исследуется вопрос о применении проекционно-итерационного подхода к решению задач оптимального управления процессами, описываемыми уравнениями колебания. При этом в роли проекционного метода предлагается использовать метод конечных разностей, а в роли итерационного метода минимизации функционала качества – метод условного градиента. Предложенный подход, укладывается в общую схему проекционно-итерационных методов решения задач минимизации с ограничениями в гильбертовых пространствах, теоретически обоснованных в [3], и сопровождается детально описанной вычислительной технологией, позволяющей довести идею проекционно-итерационного метода до успешного расчета.

Постановка задачи. Пусть имеется однородная упругая гибкая струна $0 \le s \le l$, один конец которой свободен, на другой ее конец действует внешняя сила $p=p(t)$ и, кроме того, к каждой точке струны также приложена внешняя сила $f = f(s, t)$. Требуется, управляя указанными внешними силами, к заданному моменту времени T привести струну в состояние, как можно меньше отличающееся от некоторого заданного состояния (например, состояния покоя) [1].

Математическая формулировка этой задачи: минимизировать функционал

$$J(u)= \beta_0 \int_0^l |x(s,T,u) - y(s)|^2 ds + \beta_1 \int_0^l \left| \frac{\partial x}{\partial t}(s,T,u) - z(s) \right|^2 ds \qquad (1)$$

при условиях

$$\frac{\partial^2 x}{\partial t^2} = a^2 \frac{\partial^2 x}{\partial s^2} + f(s,t), \qquad (s, t) \in Q=\{0<s<l,\ 0<t<T\}, \qquad (2)$$

$$\frac{\partial x}{\partial s}(0, t)= p(t), \qquad \frac{\partial x}{\partial s}(l, t)= 0, \qquad 0< t< T, \qquad (3)$$

$$x(s, 0)= \varphi(s), \qquad \frac{\partial x}{\partial t}(s, 0)= \mu(s), \qquad 0\le s\le l, \tag{4}$$

$$u= (p(t), f(s, t))\in U, \tag{5}$$

где $a> 0$ – скорость распространения колебаний, $l> 0$, $T> 0$, $\beta_0 \ge 0$, $\beta_1 \ge 0$ – заданные постоянные, $\beta_0^2 + \beta_1^2 > 0$; $y(s)$, $z(s)$, $\varphi(s)$, $\mu(s)$ – заданные функции на отрезке $0\le s\le l$, причем $y(s), z(s), \mu(s)\in L_2[0, l]$, $\varphi(s)\in W_2^{(1)}[0, l]$, $W_2^{(1)}[0, l]$ – пространство Соболева [4]; U – заданное выпуклое замкнутое ограниченное множество из $H= L_2[0,T]\times L_2(Q)$. В частном случае, когда $y(s)= z(s)= 0$, $\beta_0 = \beta_1 = 1$, будем иметь дело с задачей о наилучшем успокоении струны к моменту времени T; если же $\inf\limits_U J(u)= 0$, то можно говорить о возможности полного успокоения струны к моменту T.

Под решением краевой задачи (2) – (4), соответствующим управлению $u= (p(t), f(s, t))\in H$, будем понимать функцию $x= x(s, t)\equiv x(s, t, u)\in W_2^{(1)}(Q)$, след [1] которой при $t= 0$ совпадает с $\varphi(s)$ и которая удовлетворяет интегральному тождеству

$$\iint\limits_Q \left(a^2 \frac{\partial x}{\partial s}\frac{\partial \psi}{\partial s} - \frac{\partial x}{\partial t}\frac{\partial \psi}{\partial t} - \psi f \right) ds\, dt + a^2 \int_0^T \psi(0,t)p(t)\,dt - \int_0^l \psi(s,0)\mu(s)\,ds = 0$$

для всех функций $\psi=\psi(s, t)\in W_2^{(1)}(Q)$, след которых при $t= T$ равен нулю.

Можно показать [1], что краевая задача (2) – (4) при каждом $u\in H$ имеет, и притом единственное, решение, которое может быть представлено с помощью формулы Даламбера, а функционал (1) при условиях (2) – (5) является выпуклым и непрерывнодифференцируемым для всех $u\in H$, причем его градиент на H удовлетворяет условию Липшица и имеет вид

$$J'(u)= (a^2\psi(0, t, u); \psi(s, t, u))\in H, \tag{6}$$

где функция $\psi= \psi(s, t)\equiv \psi(s, t, u)$ является решением следующей вспомогательной краевой задачи:

$$\frac{\partial^2 \psi}{\partial t^2} = a^2 \frac{\partial^2 \psi}{\partial s^2}, \qquad (s, t)\in Q, \tag{7}$$

$$\frac{\partial \psi}{\partial s}(0, t)= \frac{\partial \psi}{\partial s}(l, t)= 0, \qquad 0< t< T,$$

$$\psi(s, T)= 2\beta_1\left(\frac{\partial x}{\partial t}(s,T,u)- z(s)\right),$$

$$\frac{\partial \psi}{\partial t}(s, T)= -2\beta_0\big(x(s,T,u)- y(s)\big), \quad 0\le s\le l.$$

По теореме 3.6 из [1] функционал (1) при условиях (2) –(5) достигает на U своей нижней грани J^*, т. е. существует оптимальное решение $u^*=(p^*(t), f^*(s, t)) \in U$ задачи (1)–(5). Для минимизации функционала (1) при условиях (2)–(5) будем использовать метод условного градиента, предполагая, что множество U состоит из управлений $u=(p(t), f(s, t)) \in H$, удовлетворяющих условиям

$$\int_0^T p^2(t)\,dt \le R_0^2, \qquad \iint_Q f^2(s,t)\,ds\,dt \le R_1^2, \tag{8}$$

где $R_0 > 0$, $R_1 > 0$ – заданные числа.

Рассмотрим конечно-разностный метод приближенного решения задачи (1)–(5). Введем в прямоугольнике Q равномерную сетку точек

$$\overline{\omega}_{h\tau} = \overline{\omega}_h \times \overline{\omega}_\tau = \{(s_i, t_j) \in Q:\ s_i = ih,\ i=0,1,\cdots,N;\ t_j = j\tau,\ j=0,1,\cdots,M\},$$

где h и τ – шаги сетки по направлениям s и t соответственно, $h= l/N$, $\tau= T/M$. Интегралы в (1) заменим квадратурными суммами по формуле прямоугольников, уравнение (2) – разностными уравнениями по неявной схеме [5], производные $\frac{\partial x}{\partial t}$, $\frac{\partial x}{\partial s}$ в (1), (3), (4) – их конечно-разностными отношениями. В результате прийдем к следующей задаче минимизации разностного функционала

$$J_{h\tau}(\tilde{u}) = \beta_0 \sum_{i=1}^{N-1}(x_{iM} - y_i)^2 h + \beta_1 \sum_{i=1}^{N-1}\left(\frac{x_{iM} - x_{i,M-1}}{\tau} - z_i\right)^2 h \tag{9}$$

при условиях

$$\frac{1}{\tau^2}(x_{i,j+1} - 2x_{ij} + x_{i,j-1}) = \frac{a^2}{h^2}\big[\sigma(x_{i+1,j+1} - 2x_{i,j+1} + x_{i-1,j+1}) +$$
$$+(1-2\sigma)(x_{i+1,j} - 2x_{ij} + x_{i-1,j}) + \sigma(x_{i+1,j-1} - 2x_{i,j-1} + x_{i-1,j-1})\big] + f_{ij}, \tag{10}$$
$$i= 1,2,..., N\text{-}1; \quad j= 1,2,..., M\text{-}1,$$

$$\frac{1}{h}(x_{1j} - x_{0j}) = p_j, \quad \frac{1}{h}(x_{Nj} - x_{N-1,j}) = 0, \quad j= 1,\cdots, M-1, \tag{11}$$

$$x_{i0} = \varphi_i, \quad \frac{1}{\tau}(x_{i1} - x_{i0}) = \mu_i, \quad i= 0,1,\cdots, N, \tag{12}$$

где $x_{ij} \approx x(s_i, t_j)$, $y_i= y(s_i)$, $z_i= z(s_i)$, $f_{ij}= f(s_i, t_j)$, $p_j= p(t_j)$, $\varphi_i= \varphi(s_i)$, $\mu_i= \mu(s_i)$,

$$\tilde{u} = (\tilde{p}, \tilde{f}) \in \tilde{U} = \left\{ \tilde{u} = (\tilde{p}, \tilde{f}) \in \tilde{H}_{h\tau} = H_\tau \times H_{h\tau} : \ \|\tilde{p}\|_\tau \le R_0, \ \left\|\tilde{f}\right\|_{h\tau} \le R_1 \right\} \quad (13)$$

H_τ и $H_{h\tau}$ – пространства сеточных функций $\tilde{p} = \{ p_j \}_{j=\overline{0,M}}$, определенных на сетке $\overline{\omega}_\tau$, и сеточных функций $\tilde{f} = \{f_{ij}\}_{i=\overline{0,N},\, j=\overline{0,M}}$, определенных на сетке $\overline{\omega}_{h\tau}$, с нормами $\|\tilde{p}\|_\tau = \sqrt{\sum_{j=1}^{M-1} p_j^2 \tau}$ и $\left\|\tilde{f}\right\|_{h\tau} = \sqrt{\sum_{i=1}^{N-1}\sum_{j=1}^{M-1} f_{ij}^2 \, h\tau}$ соответственно [5].

(10)-(12) есть разностная начально_краевая задача, представляющая собой при каждом фиксированном j $(j = 2,..., M)$ систему трехточечных уравнений, решение $x_{ij} \approx x(s_i, t_j, \tilde{u})$ которой при любом $\tilde{u}$ из (13) существует и единственно [5]. Разностная схема (10) –(12) на решении $x(s, t)$ задачи (2)-(4) имеет аппроксимацию $O(h+\tau)$; схема устойчива по начальным данным и по правой части, если выполняется соотношение

$$\sigma \ge \frac{1+\varepsilon}{4} - \frac{h^2}{4\tau^2}, \quad (14)$$

где $0 < \varepsilon < 1$ – любое число, не зависящее от h и τ [5].

Аналогично выписывается разностная схема для краевой задачи (7):

$$\frac{1}{\tau^2}(\psi_{i,j+1} - 2\psi_{ij} + \psi_{i,j-1}) = \frac{a^2}{h^2}\left[\sigma(\psi_{i+1,j+1} - 2\psi_{i,j+1} + \psi_{i-1,j+1}) + \right.$$
$$\left. + (1-2\sigma)(\psi_{i+1,j} - 2\psi_{ij} + \psi_{i-1,j}) + \sigma(\psi_{i+1,j-1} - 2\psi_{i,j-1} + \psi_{i-1,j-1})\right], \quad (15)$$
$$i = 1,2,..., N-1; \qquad j = 1,2,..., M-1,$$
$$\frac{1}{h}(\psi_{1j} - \psi_{0j}) = \frac{1}{h}(\psi_{Nj} - \psi_{N-1,j}) = 0, \qquad j = 1,\cdots, M-1,$$
$$\psi_{iM} = 2\beta_1\left[\frac{1}{\tau}(x_{iM} - x_{i,M-1}) - z_i\right], \qquad i = 0,1,..., N,$$
$$\frac{1}{\tau}(\psi_{iM} - \psi_{i,M-1}) = -2\beta_0(x_{iM} - y_i), \qquad i = 0,1,..., N,$$

где $\psi_{ij} \approx \psi(s_i, t_j, \tilde{u})$.

Так же, как и раньше, разностная схема (15) на решении $\psi(s, t)$ задачи (7) имеет аппроксимацию $O(h+\tau)$ и устойчива при выполнимости соотношения (14). Задача (15) при каждом фиксированном j $(j = M-2,\cdots, 0)$ имеет единственное решение [5]. Таким образом, для приближенного вычисления градиента (6) функционала (1) при условиях (2) –(5) требуется после-

довательно решить краевые задачи (10) -(12) и (15), используя, к примеру, метод разностной прогонки [5].

Метод конечных разностей решения задачи (1) -(5) является по существу, методом проекционного типа, согласно которому задача минимизации ограниченного снизу функционала $F(u)$, заданного на некотором множестве Ω вещественного гильбертова пространства H,

$$F(u) \to inf, \quad u \in \Omega, \tag{16}$$

аппроксимируется последовательностью задач минимизации «приближенных» функционалов $\tilde{F}_n(\tilde{u}_n)$, заданных в некоторых гильбертовых пространствах $\tilde{H}_n$, изоморфных подпространствам H_n исходного пространства $(H_1 \subset H_2 \subset ... \subset H_n \subset ... ... \subset H)$,

$$\tilde{F}_n(\tilde{u}_n) \to inf, \quad \tilde{u}_n \in \tilde{\Omega}_n \subset \tilde{H}_n, \quad n = 1,2,..., \tag{17}$$

где $\tilde{\Omega}_n = \{ \tilde{u}_n \in \tilde{H}_n : \tilde{u}_n = \Phi_n u_n, \ u_n \in \Omega_n = \Omega \cap H_n \}$, Φ_n – линейный оператор, взаимно однозначно отображающий H_n на $\tilde{H}_n$. В самом деле, в роли задачи (16) здесь выступает задача минимизации при условиях (2) -(4) функционала (1) ($F(u)=J(u)$), заданного на множестве (5), (8) ($\Omega = U$) гильбертова пространства $H = L_2[0,T] \times L_2(Q)$, а роль приближенных задач (17) играют задачи минимизации при условиях (10) -(12) функционалов (9) ($\tilde{F}_n(\tilde{u}_n) = J_{h_n\tau_n}(\tilde{u}_n)$), заданных на множествах вида (13) ($\tilde{\Omega}_n = \tilde{U}_n = \{ \tilde{u}_n : \tilde{u}_n = (\tilde{p}_n, \tilde{f}_n) \in \tilde{H}_{h_n\tau_n} = H_{\tau_n} \times H_{h_n\tau_n} : \|\tilde{p}_n\|_{\tau_n} \le R_0, \ \|\tilde{f}_n\|_{h_n\tau_n} \le R_1 \}$) из гильбертовых пространств $\tilde{H}_n = \tilde{H}_{h_n\tau_n}$, где H_{τ_n} и $H_{h_n\tau_n}$ – соответственно пространства сеточных функций $\tilde{p}_n = \{ p_j \}_{j=\overline{0,M_n}}$, определенных на сетке $\bar{\omega}_{\tau_n}$, и сеточных функций $\tilde{f}_n = \{ f_{ij} \}_{i=\overline{0,N_n}, \, j=\overline{0,M_n}}$, определенных на сетке $\bar{\omega}_{h_n\tau_n}$, с нормами

$$\|\tilde{p}_n\|_{\tau_n} = \sqrt{\sum_{j=1}^{M_n-1} p_j^2 \tau_n} \quad \text{и} \quad \|\tilde{f}_n\|_{h_n\tau_n} = \sqrt{\sum_{i=1}^{N_n-1} \sum_{j=1}^{M_n-1} f_{ij}^2 \, h_n \tau_n} \ ;$$

при этом $h_n = l/N_n$, $\tau_n = T/M_n$,

$$\bar{\omega}_{h_n\tau_n} = \bar{\omega}_{h_n} \times \bar{\omega}_{\tau_n} = \{ (s_i, t_j) \in Q: \ s_i = ih_n, \ i = 0,1,\cdots,N_n; \ \ t_j = j\tau_n, \ j = 0,1,\cdots,M_n \},$$

$$N_1 > 0, \ M_1 > 0, \ N_{n+1} \ge N_n, \ M_{n+1} \ge M_n, \ \ n = 1,2,....$$

Пространства $\tilde{H}_n = \tilde{H}_{h_n\tau_n}$ изоморфны подпространствам $H_n \subset H$ элементов $u_n = (p_n(t), f_n(s,t))$, составленных из кусочно-постоянных функций вида

$$p_n(t)=\sum_{j=1}^{M_n-1} p_j\,\chi_j(t),\qquad f_n(s,t)=\sum_{i=1}^{N_n-1}\sum_{j=1}^{M_n-1} f_{ij}\,\chi_{ij}(s,t), \tag{18}$$

где $\chi_j(t)$ и $\chi_{ij}(s,t)$ – характеристические функции соответственно частичного промежутка $\{t_{j-1}<t\le t_j\}$ сетки $\overline{\omega}_{\tau_n}$ и ячейки $\{s_{i-1}<s\le s_i,\ t_{j-1}<t\le t_j\}$ сеточной области $\overline{\omega}_{h_n\tau_n}$. Оператор Φ_n, отображающий H_n на $\tilde{H}_n$, определяется как оператор, который каждому элементу $u_n=(p_n(t),f_n(s,t))$ вида (18) ставит во взаимно однозначное соответствие элемент $\tilde{u}_n=(\tilde{p}_n,\tilde{f}_n)$, составленный из сеточных функций $\tilde{p}_n=\{p_j\}_{j=\overline{0,M_n}}\in H_{\tau_n}$, $\tilde{f}_n=\{f_{ij}\}_{i=\overline{0,N_n},\,j=\overline{0,M_n}}\in H_{h_n\tau_n}$ таких, что $p_j=p_n(t_j)$, $f_{ij}=f_n(s_i,t_j)$, $i=1,2,\ldots,N_n-1$; $j=1,2,\ldots,M_n-1$. Φ_n^{-1} – оператор, осуществляющий обратное отображение.

Условия сходимости проекционного метода для решения экстремальных задач с ограничениями сформулированы в [6].

Рассмотрим вопрос о применении к решению задачи (1)–(5) проекционно-итерационного метода, основанного на методе условного градиента [3]. Будем для каждой из «приближенных» задач минимизации (9)–(13) (или, что то же самое, «приближенных» задач (17)) строить итерационным методом условного градиента несколько (k_n) приближений $\tilde{u}_n^{(k)}$ ($k=1,2,\cdots,k_n$), последнее из которых с использованием кусочно-постоянного продолжения (18) будем принимать в качестве начального приближения к решению следующей, $(n+1)$-й, «приближенной» задачи:

$$\tilde{u}_n^{(k+1)}=\tilde{u}_n^{(k)}+\alpha_n^{(k)}(\bar{u}_n^{(k)}-\tilde{u}_n^{(k)}),\qquad \tilde{u}_{n+1}^{(0)}=\Phi_{n+1}\Phi_n^{-1}\tilde{u}_n^{(k_n)}, \tag{19}$$

$$(k=0,1,\cdots,k_n-1;\quad n=1,2,\cdots;\quad \tilde{u}_1^{(0)}\in\tilde{U}_1),$$

где $\bar{u}_n^{(k)}\in\tilde{U}_n$ – вспомогательное приближение, определяемое из условия

$$(\tilde{F}_n'(\tilde{u}_n^{(k)}),\bar{u}_n^{(k)})_{h\tau}=\min_{\tilde{u}_n\in\tilde{U}_n}(\tilde{F}_n'(\tilde{u}_n^{(k)}),\tilde{u}_n)_{h\tau}, \tag{20}$$

$\alpha_n^{(k)}$ – итерационный параметр, такой, что

$$\tilde{G}_n^{(k)}(\alpha_n^{(k)})=\min_{0\le\alpha_n\le1}\tilde{G}_n^{(k)}(\alpha_n),\quad \tilde{G}_n^{(k)}(\alpha_n)\equiv\tilde{F}_n(\tilde{u}_n^{(k)}+\alpha_n(\bar{u}_n^{(k)}-\tilde{u}_n^{(k)})). \tag{21}$$

Условие (20) с учетом (13) допускает [1] явное выражение для $\bar{u}_n^{(k)}=(\bar{p}_n^{(k)},\bar{f}_n^{(k)})$, $\bar{p}_n^{(k)}=\{\bar{p}_j^{(k)}\}_{j=\overline{0,M_n}}$, $\bar{f}_n^{(k)}=\{\bar{f}_{ij}^{(k)}\}_{i=\overline{0,N_n},\,j=\overline{0,M_n}}$:

$$\bar{p}_j^{(k)} = -\frac{R_o \psi_{0j}^{(k)}}{\sqrt{\sum_{j=1}^{M_n-1} \left|\psi_{0j}^{(k)}\right|^2 \tau_n}}, \quad \bar{f}_{ij}^{(k)} = -\frac{R_1 \psi_{ij}^{(k)}}{\sqrt{\sum_{i=1}^{N_n-1}\sum_{j=1}^{M_n-1} \left|\psi_{ij}^{(k)}\right|^2 h_n \tau_n}},$$

где $\psi_{ij}^{(k)} \approx \psi(s_i, t_j, \tilde{u}_n^{(k)})$ – решение разностной краевой задачи (15).

Что касается выбора итерационного параметра $\alpha_n^{(k)}$ из условия (21), то по результатам [1] $\alpha_n^{(k)} = min\{1; \alpha_n^{*(k)}\} > 0$, где

$$\alpha_n^{*(k)} = \frac{1}{2}\left[\sum_{j=1}^{M_n-1} a^2 \psi_{0j}^{(k)} (p_j^{(k)} - \bar{p}_j^{(k)})\tau_n + \sum_{i=1}^{N_n-1}\sum_{j=1}^{M_n-1} \psi_{ij}^{(k)} (f_{ij}^{(k)} - \bar{f}_{ij}^{(k)}) h_n \tau_n\right] \times$$

$$\times \left[\sum_{i=1}^{N_n-1}\left(\beta_0 \left|x_{iM_n}^{(k)} - \bar{x}_{iM_n}^{(k)}\right|^2 + \frac{\beta_1}{\tau}\left|(x_{iM_n}^{(k)} - x_{i,M_n-1}^{(k)}) - (\bar{x}_{iM_n}^{(k)} - \bar{x}_{i,M_n-1}^{(k)})\right|^2\right) h_n\right]^{-1},$$

причем если $\alpha_n^{*(k)} = 0$ или $x(s_i, T, \bar{u}_n^{(k)}) \equiv x(s_i, T, \tilde{u}_n^{(k)})$, то $\tilde{u}_n^{(k)} = (\tilde{p}_n^{(k)}, \tilde{f}_n^{(k)})$ – оптимальное решение задачи (9) –(13) и итерации при данном n заканчиваются.

В результате описанного процесса приближений (19) –(21) возникает последовательность $\{\tilde{u}_n^{(k_n)}\}$, $n=1,2,...$, которая при кусочно –постоянном восполнении (18) выступает в роли последовательности приближений к решению исходной задачи (1)-(5).

Приведем условия сходимости проекционно–итерационного метода (19)-(21) применительно к задаче минимизации (16) (или, что то же самое, задаче (1)-(5)).

Т е о р е м а [3]. *Пусть функционал $F(u)$ определен на выпуклом замкнутом ограниченном множестве $\Omega \subset H$, а функционалы $\tilde{F}_n(\tilde{u}_n)$ – на выпуклых замкнутых множествах $\tilde{\Omega}_n \subset \tilde{H}_n$, причем Ω и $\tilde{\Omega}_n$ связаны условием: для любого $u \in \Omega$ существует последовательность $\{\tilde{u}_n\}$, $n=1,2,...$, $\tilde{u}_n \in \tilde{\Omega}_n$ такая, что $\lim\limits_{n\to\infty} \Phi_n^{-1}\tilde{u}_n = u$. Предположим, что $\tilde{F}_n(\tilde{u}_n) \in C^1(\tilde{\Omega}_n)$, $\inf\limits_{u\in\Omega} F(u) = F^* > -\infty$ и $\left|\tilde{F}_n(\tilde{u}_n) - F(\Phi_n^{-1}\tilde{u}_n)\right| \le \tilde{\beta}_n$ для всех $\tilde{u}_n \in \tilde{\Omega}_n$ с $\tilde{\beta}_n \to 0$ при $n \to \infty$, $\sum\limits_{n=1}^{\infty} \tilde{\beta}_n < \infty$. Пусть, кроме того, $\sum\limits_{n=1}^{\infty} \tilde{\alpha}_n^2 < \infty$, $0 \le \tilde{\alpha}_n \le 1$, и градиент $\tilde{F}_n'(\tilde{u}_n)$ удовлетворяет на $\tilde{\Omega}_n$ условию Липшица. Если функционалы*

$F(u)$ и $\tilde{F}_n(\tilde{u}_n)$ выпуклы на Ω и $\tilde{\Omega}_n$ соответственно, то последовательность $\{\Phi_n^{-1}\tilde{u}_n^{(k_n)}\}$ является минимизирующей для $F(u)$ и любая ее слабая предельная точка есть точка минимума $F(u)$ на Ω, причем в случае единственности точки минимума к ней слабо сходится вся последовательность $\{\Phi_n^{-1}\tilde{u}_n^{(k_n)}\}$.

Выполнимость условий этой теоремы для задачи (1)–(5) проверяется так же, как в [6].

Рассмотрим применение проекционно-итерационного метода (19)–(21) к решению конкретной задачи оптимального управления процессом колебания однородной упругой гибкой струны с закрепленными концами. При этом в задаче (1)–(5) краевые условия (3) имеют вид

$$x(0, t)= x(l, t)= 0, \qquad 0\le t\le T,$$

управление ищется в скалярном виде $u= f(s, t)$, а основные параметры задачи принимают такие значения: $\beta_0=\beta_1=1$, $l=T=1$, $a^2=1$, $y(s)= s^2\left(1-\frac{s}{l}\right)$, $z(s)= s\,(s-l)$, $\varphi(s)= \sin s\cdot\sin(l-s)$, $\mu(s)= s\cdot tg(l-s)$, $R_0= 0$, $R_1= 1$. Совокупность вложенных вдвое сеток $\bar{\omega}_{h_n\tau_n}$, $n= 1,2,..., p$, определялась величиной первоначального разбиения $N_1= M_1= 4$ и для получения приближенного решения с точностью ε=0,001 оказалась состоящей из $p= 4$ сеток с размерностью последней $N_p= M_p= 32$. При этом количество k_n строящихся приближений на шаге с номером n определялось как наименьшее целое k, удовлетворяющее неравенствам

$$\left\|\tilde{u}_n^{(k+1)}-\tilde{u}_n^{(k)}\right\|_{h_n\tau_n}<\varepsilon_n,\ \left|J_{h_n\tau_n}(\tilde{u}_n^{(k+1)})-J_{h_n\tau_n}(\tilde{u}_n^{(k)})\right|<\varepsilon_n,\quad k=1,2,....,$$

где ε_n выбиралось величиной порядка разностной аппроксимации исходной задачи управления при данном n, а именно $\varepsilon_n= C(h_n+\tau_n)$, $C=const>0$. Критерием окончания проекционно-итерационного процесса являлось условие $\varepsilon_n\le\varepsilon$, $n=1,2,...$. Начальные значения для сеточной функции управления $\tilde{u}_1^{(0)}$ на первой сетке были приняты во всех ее узлах равными 0,9. Расчеты проводились на ЭВМ типа Pentium-166.

В ходе реализации описанного проекционно-итерационного метода было выполнено в общей сложности $k=\sum_{n=1}^{p}k_n=2+3+3+4=12$ итераций. Время счета при этом составило $t=$ 7,49 мин. Для минимизируемого функционала (1) было получено приближенное значение $J\approx 0{,}007509$.

Наряду с проекционно ‑итерационным методом для решения упомянутой конкретной задачи оптимального управления был применен обычный конечно‑разностный метод на одной последней сетке с $N = M = 32$. При тех же начальных значениях $\tilde{u}^{(0)}$ в итерационном процессе, равных в каждом узле $0{,}9$, приближенное решение задачи получено с точностью $\varepsilon = 0{,}001$ за 9 итераций, что потребовало 19,30 мин. машинного времени. Приближенное значение функционала (1) составило $J \approx 0{,}007614$.

Анализ результатов показывает, что проекционно ‑итерационный подход приводит к уменьшению вычислительных затрат на построение приближений и обеспечивает большую точность получаемых приближенных решений.

Библиографические ссылки

1. **Васильев Ф.П.** Методы решения экстремальных задач. / Ф.П. Васильев – М., 1981. – 400 с.
2. **Гарт Л.Л.** Проекционно‑итерационный метод решения задачи оптимального управления процессом колебания струны / Л.Л. Гарт, Н.В. Поляков // Математичне моделювання. – Дніпродзержинськ, 2001. – № 1(6). – С. 12–15.
3. **Тавадзе Э.Л.** Проекционно‑итерационный метод решения задачи минимизации с ограничениями, основанный на методе условного градиента / Э.Л. Тавадзе, Л.Л. Тавадзе // Математическое моделирование. – Днепродзержинск: ДГТУ, 1998. – № 3. – С. 128–134.
4. **Васильев Ф.П.** Лекции по методам решения экстремальных задач. / Ф.П. Васильев – М., 1974. – 374 с.
5. **Самарский А.А**. Теория разностных схем. / А.А. Самарский – М., 1977. – 656 с.
6. **Тавадзе Л.Л.** Применение проекционно ‑итерационного метода к решению задачи об оптимальном нагреве стержня. / Л.Л. Тавадзе – Днепропетровск, 1997. – 16 с. – Деп. в ГНТБ Украины 27.10.97, N 534 – Ук 97.

Надійшла до редколегії 01.05.10.

УДК 519.23

©**С.Н. Герасин**[*], **Е.А. Михайлов**[**], **С.В. Яковлев**[**]
[*]*Харьковский национальный университет радиоэлектроники,*
[**]*Харьковский институт управления*

НЕОДНОРОДНАЯ МАРКОВСКАЯ МОДЕЛЬ ПРОЦЕССОВ ПЕРЕРАСПРЕДЕЛЕНИЯ ДОХОДОВ В СИСТЕМЕ «ЗАПАС-ПОТРЕБЛЕНИЕ»

Запропонована стохастична інтерпретація моделі динамічного розподілу ресурсів. Формалізм моделі використовує мову неоднорідних марківських ланцюгів. Показана можливість використання запропонованої моделі при розв'язанні задач економічного прогнозування.

Предложена стохастическая интерпретация модели динамического распределения ресурсов. Формализм модели использует язык неоднородных марковских цепей. Показана возможность использования предложенной модели в задачах экономического прогнозирования.

Stochastic interpretation of model of dynamic allocation of resources is offered. Model formalism is utilized by the language of inhomogeneous Markov's chains. Possibility of the use of the offered model is solution in the tasks of economic prognostication.

Ключевые слова: марковские цепи, замкнутая система, исследование операций, стохастическая модель.

Общая постановка задачи и её актуальность. Задачи распределения и управления запасами относятся к числу наиболее важных задач теории исследования операций, при этом основной упор делается либо на детерминированные модели с дальнейшей оптимизацией, либо на стохастические с дальнейшим прогнозированием ресурсов. При этом довольно часто для формирования и анализа стохастической модели используется аппарат цепей Маркова, в связи с этим представляется перспективным построение соответствующей модели распределения базирующейся на аппарате неоднородных марковских цепей.

Цель данной статьи состоит в том, чтобы средствами теории вероятностей построить формальную модель перераспределения доходов в замкну-

той экономической системе и оценить доходность каждой из ее самостоятельных компонент.

Предлагаемая модель динамического распределения межпроизводственных потоков (в денежном выражении) отражает взаимосвязь между структурой производства и потребления (на уровне региона или в целом по народному хозяйству) и изменением дохода отдельных предприятий и суммарного дохода по региону в целом. В модели используется известный принцип представления структуры производства-потребления в виде матрицы коэффициентов затрат [1] замкнутой системы, а также метод отражения зависимости между объемом дохода и структурой производства-потребления с помощью матрицы коэффициентов прибыльности вложений [2].

Модель позволяет:

1) осуществить выбор оптимальной, с точки зрения роста суммарного дохода, структуры производства-потребления;

2) осуществить выбор оптимальной структуры производства для того или иного предприятия, с точки зрения роста его доходов, при заданной структуре производства других предприятий региона;

3) осуществить прогноз изменения суммарного дохода и дохода той или иной отрасли при данной структуре производства-потребления.

Настоящая работа опирается на предложенные в [3,4] модели массодинамики и позволяет достаточно точно описывать процесс распределения ресурсов с учетом времени.

Описание и общие характеристики модели. Введем некоторые обозначения:

$V(t)$ – суммарный доход на момент t;

$p(t) = (p_1(t), p_2(t), \ldots, p_n(t))$ – вектор коэффициентов распределения суммарного дохода по предприятиям региона:

$V_j(t) = V(t)p_j(t)$ – доход j-го предприятия на момент t;

$P(t) = \left\| p_{ij}(t) \right\|_{i,j=1}^{n}$ – матрица коэффициентов затрат, отражающая структуру производства-потребления:

$p_{ij}(t)$ – часть дохода i-го предприятия, направляемая на приобретение продукта j-го предприятия,

$$p(t+1) = p(t)P(t);$$

$R_{ij}(t) = \left\| r_{ij}(t) \right\|_{i,j=1}^{n}$ – матрица коэффициентов прибыльности вложений на момент t:

$r_{ij}(t)$ – величина прибыли, получаемой i-м предприятием на единицу затрат по приобретению продукции j-го предприятия, на момент t;

$k_i(t) = \sum_{j=1}^{n} r_{ij}(t)\, p_{ij}(t)$ – совокупная прибыль, получаемая i-м предприятием на единицу затрат, произведенных на момент t;

$k(t) = \sum_{i=1}^{n} p_i(t) k_i(t) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_i(t)\, p_{ij}(t) r_{ij}(t)$ – величина относительного изменения (увеличения при $k(t)>0$ и уменьшения при $k(t)<0$) суммарного дохода по региону на момент t при данной структуре производства:

$V(t+1) = (1+k(t))V(t)$,

$k(t) \geq -1$ (получаемый доход не может быть отрицательным).

Как отмечалось выше, предлагаемая модель является моделью замкнутой системы. На практике, однако, ни одна экономическая система не является замкнутой: всегда присутствует товарообмен предприятий данного региона или страны с внешними контрагентами. Для преодоления этого противоречия модель предусматривает введение дополнительного фиктивного предприятия, представляющего в агрегированной форме всех внешних контрагентов. Поскольку в реальной практике одним из контрагентов рынка всегда является государство или отдельные его органы, то в модель в качестве еще одного фиктивного предприятия введен элемент, представляющий государство в структуре производства потребления. Наконец, поскольку источником всякой стоимости является живой труд, а конечным потребителем продукции всякого предприятия (отрасли) является население, то в модели также использовано агрегированное представление последнего как еще одного фиктивного предприятия (отрасли). На основании вышеизложенного, для отдельных элементов матрицы коэффициентов затрат $P(t) = \left\| p_{ij}(t) \right\|_{i,j=1}^{n}$ можно предложить следующие интерпретации:

$p_{i1}(t)$ – часть дохода i-го предприятия, полученного на момент t, расходуемая на выплату заработной платы;

$p_{i2}(t)$ – часть дохода i-го предприятия, полученного на момент t, расходуемая на обязательные отчисления в государственный и местный бюджеты, а также в различные фонды;

$p_{in}(t)$ – часть дохода i-го предприятия, полученного на момент t, расходуемая на закупку продукции внешних контрагентов;

$p_{1i}(t)$ – часть доходов населения, полученных на момент t, расходуемая на приобретение продукции i-го предприятия;

$p_{2i}(t)$ – часть государственных доходов, полученных на момент t, расходуемая на а) закупку продукции i-го предприятия, б) государственные инвестиции в i-е предприятие;

$p_{21}(t)$ – часть государственных доходов, полученных на момент t, расходуемая на выплаты населению (зарплаты в бюджетной сфере, пенсии, стипендии, пособия, материальная помощь населению);

$p_{12}(t)$ – часть доходов населения, полученных на момент t, расходуемая на уплату налогов (подоходного, на недвижимое имущество, на транспортные средства);

$p_{2n}(t)$ – часть государственных доходов, полученных на момент t, расходуемая на осуществление платежей внешним контрагентам (закупка продукции внешних производителей, платежи по кредитам из внешних источников, кредитование внешних агентов, безвозмездная помощь внешним субъектам);

$p_{n2}(t)$ – часть доходов, полученных на момент t полученных от внешних контрагентов, получаемая государством (таможенные пошлины, кредиты государству из внешних источников, платежи по кредитам государства внешним агентам);

$p_{1n}(t)$ – часть доходов населения, полученных на момент t, расходуемая на выплаты внешним контрагентам (денежные переводы за пределы региона, расходы на туристические поездки, приобретение продукции непосредственно у внешних производителей);

$p_{n1}(t)$ – часть доходов, полученных на момент t полученных от внешних контрагентов, получаемая населением (денежные переводы из -за пределов региона, заработная плата, полученная за работу на внешних контрагентов и т. п.).

При определении коэффициентов прибыльности вложений для каждой отрасли следует исходить из следующих принципов:

1) поскольку источником всякой стоимости является живой труд, то расходы на заработную плату всегда прибыльны: $r_{i1}(t) > 0, 2 < i < n, \forall t > 0$;

2) отдача от живого труда обычно пропорциональна энерго - и технической вооруженности производства, поэтому для j, соответствующих расходам на энергообеспечение и закупку оборудования также следует положить $r_{ij}(t) > 0, 2 < i < n, \forall t > 0$;

3) затраты на приобретение сырья обычно сами по себе не приводят ни к увеличению, ни к уменьшению дохода (при нормальных пропорциях относительно других статей расходов);

4) расходы по уплате государственных сборов и налогов обычно непосредственно не продуктивны, поэтому для всех i следует положить $r_{i2}(t) \le 0$.

При определении конкретных значений этих величин следует учитывать, что они, в большой степени, зависят от пропорций расходов (то есть реально имеем $r_{ij}(t) = r_{ij}(p_{i1}(t), p_{i2}(t), ..., p_{in}(t))$), но в любом случае для любого i и j и всех $t>0$ должно выполняться неравенство $r_{ij}(t) \ge -1$ (то есть даже при самом неразумном вложении средств нельзя потерять больше, чем их имеется в наличии).

Стохастическая интерпретация модели. В классической модели коэффициентов затрат [1] предполагалось, что каждое предприятие на каждом шаге полностью расходует весь полученный на этот момент доход - такое предположение было оправданным на том основании, что при представлении взаимосвязей между отдельными производствами посредством статической модели (охватывающей большие промежутки времени порядка года) такое допущение действительно приближенно соответствовало реальному положению вещей. В предлагаемой модели, так как она, во - первых, является динамической, а во -вторых, поскольку ее временной масштаб имеет порядок недель или месяцев, более оправданным является предположение, что в матрицах коэффициентов затрат для различных периодов не все диагональные элементы будут равны нулю. Следует также отметить, что данное предположение, также как и предположение о неоднородности цепи Маркова с матрицами переходных вероятностей за единицу времени, совпадающими с матрицами коэффициентов затрат для отдельных периодов в рассматриваемой модели, обеспечивает отражение в этой модели различной длительности производственных циклов для различных производств. Кроме того, модель можно модифицировать таким

образом, чтобы в ней учитывалось изменение числа предприятий с течением времени: для этого можно использовать рассматривавшуюся в [5] модель цепей Маркова с переменным числом состояний.

В соответствии с принятыми в данной модели предположениями, для величины дохода $V(k,T)$, полученного с момента s в течении промежутка времени T имеем

$$V(s,T)=\sum_{t=0}^{T}\beta^t\sum_{i=1}^{n}p_i(s+t-1)\sum_{j=1}^{n}p_{ij}(s+t)r_{ij}(s+t)=$$

$$=\sum_{t=0}^{T}\beta^t\sum_{i=1}^{n}p_i(s+t-1)k_i(s+t), \qquad (1)$$

где β – коэффициент дисконтирования (величина, обратная норме банковского процента).

При $T\to\infty$ из (1) получаем

$$V(s,\infty)=\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{n}p_i(s+t-1)k_i(s+t). \qquad (2)$$

В случае, когда неоднородная цепь Маркова с матрицами переходных вероятностей за единицу времени, совпадающими с соответствующими матрицами коэффициентов затрат, является эргодической (в сильном смысле), для $V(s,\infty)$ можно получить оценку $\tilde{V}(s,\infty)$, которая включает в себя суммирование только конечного числа слагаемых. Рассмотрим вначале величину

$$V'(s,\infty)=\sum_{t=0}^{N}\beta^t\sum_{i=1}^{n}p_i(s,t-1)k_i(s+t)+\sum_{t=N+1}^{\infty}\beta^t\sum_{i=1}^{n}p_i^*q_i(s+t)=$$

$$=\sum_{t=0}^{N}\beta^t\sum_{i=1}^{n}p_i(s,t-1)k_i(s+t)+\sum_{i=1}^{n}p_i^*\sum_{t=N+1}^{\infty}\beta^t q_i(s+t),$$

где $p_i^*, 1\le i\le n,$ – компоненты вектора предельного распределения p^* соответствующей неоднородной цепи Маркова; N – число переходов, достаточное либо для сходимости распределения указанной цепи к предельному, так называемый процесс фокусировки (условия, которые для этого должны выполняться, приведены в [6]), либо для того, чтобы распределение цепи достаточно мало отличалось от своего предела (σ-фокусировка [6]).

В общем случае, использование величины $V'(s,\infty)$ вместо $V(s,\infty)$ не приводит к существенному упрощению расчетов, однако в некоторых частных случаях на основе этой величины можно получить оценки величины $V(s,\infty)$, достаточно приемлемые в смысле точности, получение которых гораздо менее трудоемко с вычислительной точки зрения.

Пусть $r_{ij}(t) \equiv r_{ij} = const$. В этом случае имеем

$$\left|\sum_{t=N+1}^{\infty}\beta^t\sum_{j=1}^{n}p_{ij}(s+t)r_{ij}\right| \le \sum_{t=N+1}^{\infty}\beta^t\sum_{j=1}^{n}p_{ij}(s+t)\left|r_{ij}\right| = \sum_{j=1}^{n}\left|r_{ij}\right|\sum_{t=N+1}^{\infty}p_{ij}(s+t)\beta^t \le$$

$$\le R_i\sum_{t=N+1}^{\infty}\beta^t = R_i\frac{\beta^{N+1}}{1-\beta} \qquad (3)$$

где

$$R_i = \sum_{j=1}^{n}\left|r_{ij}\right| < \infty, 1 \le i \le n.$$

Для всякого $\varepsilon > 0$ при достаточно больших N будет справедливо неравенство $\frac{\beta^{N+1}}{1-\beta} < \frac{\varepsilon}{R}$,

где $R = \max\limits_{1\le i\le n} R_i$. Тогда для выражения в правой части (3) имеем $R_j\frac{\beta^{N+1}}{1-\beta} < \varepsilon$.

То есть при подходящих значениях β в качестве оценки $\tilde{V}(s,\infty)$ величины $V(s,\infty)$ можно принять

$$\tilde{V}(s,\infty) = \sum_{t=0}^{N}\beta^t\sum_{i=1}^{n}p_i(s,t-1)k_i(s+t). \qquad (4)$$

Нетрудно заметить, что оценка (4) является удовлетворительной только при специальных значениях параметра β, что значительно сужает область ее применимости. Гораздо больший интерес представляет поэтому случай, когда отдача от вложений зависит только от направления расходов, но не от их источника (такое предположение будет оправдано в том случае, когда модель охватывает предприятия, которые делятся на две группы в зависимости от профиля – производителей и дистрибьютеров

однотипной продукции): $r_{ij}(t) = r_j(t)$. В этом случае, подставляя в (1) выражение для $r_{ij}(t)$, получим

$$V(s,T) = \sum_{t=0}^{T} \beta^t \sum_{i=1}^{n} p_i(s+t-1) \sum_{j=1}^{n} p_{ij}(s+t) r_{ij}(s+t) =$$

$$= \sum_{t=0}^{T} \beta^t \sum_{i=1}^{n} p_i(s+t-1) \sum_{j=1}^{n} p_{ij}(s+t) r_j(s+t) = \sum_{t=0}^{T} \beta^t \sum_{i=1}^{n} p_i(s+t) r_i(s+t).$$

Таким образом, при достаточно больших s можно использовать оценку

$$\tilde{V}(s,T) = \sum_{t=0}^{T} \beta^t \sum_{i=1}^{n} p_i^* r_i(s+t). \qquad (5)$$

Хотя такая оценка также требует выполнения весьма специфических условий, ее все же можно использовать и в ситуациях, когда предположение, на котором она основана само выполняется лишь приближенно. Примером такого «расширения» модели является модель, для которой имеют место соотношения

$$r_{ij}(t) = r_j(t) + \alpha_t \rho_{ij}(t),$$

$$\lim_{t\to\infty} \alpha_t = 0, \lim_{t\to\infty} \frac{\alpha_t}{\beta^{mt}} = 0 \; \forall m > 0 (m < \infty).$$

Очевидно, что при выполнении этих условий и при достаточно больших s, для таких моделей применима оценка (5).

В качестве входных данных для формирования соответствующей формальной модели необходимо иметь:

1) сведения о числе предприятий региона (или о среднем числе предприятий за достаточно продолжительный период, если выбрана модель с меняющимся числом предприятий);

2) сведения из балансовых отчетов предприятий о доходах и расходах за отчетный период;

3) сведения из бюджетной сметы о доходах и расходах бюджета за отчетный период;

4) статистические сводки о величине совокупного дохода по региону;

5) материалы социологических исследований о структуре доходов и расходов населения.

На основании этих данных можно построить матрицы коэффициентов затрат и коэффициентов прибыльности вложений.

При этом, по желанию пользователя, можно получить:

1) прогноз изменения совокупного дохода по региону с заданным упреждением;

2) прогноз изменения дохода того или иного предприятия при заданном упреждении;

3) оценку долгосрочных тенденций в экономике региона.

Следует отметить, что указанная модель представляет собой модель марковского типа не только по форме представления данных, поскольку ее действительно можно интерпретировать либо как цепь Маркова с постоянным или переменным числом состояний, либо как полумарковский процесс, описывающий эволюцию некоторой абстрактной «неделимой» денежной единицы, переходящей от одного предприятия к другому.

Выводы по результатам и направления дальнейших исследований. Предложенные в работе оценки величины доходов позволяют осуществлять прогнозируемую финансовую политику предприятия. В частности, на основании полученных результатов можно:

1) оценить влияние изменений в технологии на доход предприятия;

2) осуществить выбор оптимальной технологии из ряда альтернатив;

3) осуществить выбор оптимального решения при инвестировании в экономику региона.

Библиографические ссылки

1. **Аллен Р.** Математическая экономия / Р. Аллен – М., 1963. – 648 с.
2. **Валтер Я.** Стохастические модели в экономике / Я. Валтер – М., 1976. –232 с.
3. **Герасин С.Н.** Стабилизация решений в задачах динамического распределения ресурсов/ С.Н. Герасин // Доповіді НАН України. – 2001. – №10. – С.73–78.
4. **Герасин С.Н.** Неоднородная марковская модель массопереноса в условиях ограничений / С.Н. Герасин, Н.В.Гибкина // Математичне моделювання. – 2003. – №2(10). – С.7–10.
5. **Герасин С.Н.** Стабилизация распределений марковских цепей с переменным числом состояний за конечное время / С.Н. Герасин // Доповіді НАН України. – 2004. – №12. –С.59–63.

Надійшла до редколегії 15.03.10

УДК 519.1

©**Т. О. Гришанович**

Київський національний університет ім. Т.Шевченка

СКЛАДНІСТЬ АЛГОРИТМІВ РОЗКЛАДАННЯ ГРАФІВ

Досліджені часові складності алгоритмів декомпозиції графів за їх вершинами в залежності від способу подання графа. Розглянуто точні та наближені алгоритми.

Исследованы временные сложности алгоритмов декомпозиции графов по их вершинам. Рассмотрены точные и приближенные алгоритмы.

This work is dedicated to the analysis of the time complexity of the graphs decomposition algorithms. The exact and drawn near algorithms are considered.

Ключові слова: алгоритми розкладання графів, часова складність, оцінка складності алгоритму.

Вступ. Розвиток теорії дискретних та комбінаторних задач та практика їх розв'язання показали, що загальність методу розв'язання та його ефективність знаходяться в досить значному протиріччі. Тому виникає наступне питання: розбивати задачі на вузькі класи та, користуючись їх специфікою, розробляти для них ефективні алгоритми чи орієнтуватись на загальні, але, у свою чергу, менш ефективні алгоритми? [2]

У найширшому значенні поняття ефективності алгоритму розв'язання задачі пов'язане із обчислювальними ресурсами. Але найчастіше під найефективнішим алгоритмом розуміють найшвидший, оскільки обмеження в часі досить часто є домінуючим фактором, що визначає придатність його для використання на практиці. Визначення часової оцінки роботи алгоритму називають апріорним аналізом алгоритму. Цей аналіз ігнорує всі фактори, що залежать від конкретної обчислювальної платформи (розрахунки здійснюються на PAM – машині, що не передбачає паралельного виконання операцій [8]), мови програмування, та концентрується на визначенні типу функції, яка характеризує час виконання алгоритму [3].

Час роботи алгоритму виражається у вигляді функції від однієї змінної, яка характеризує розмір індивідуальної задачі, тобто об'єм вхідних даних, що необхідні для опису цієї задачі. Вхідна довжина індивідуальної задачі визначається як число символів у ланцюжку, що отримується застосуванням до конкретної задачі схеми кодування для масової задачі. Саме це число, вхідна довжина, і використовується в якості формальної характеристики розміру індивідуальної задачі. Спосіб вимірювання розміру входу залежить від конкретної задачі – це може бути число елементів на вході, число біт, необхідне для подання усіх вхідних даних. В окремих випадках, розмір входу визначається кількома числами. Наприклад, число вершин та число ребер графа [8].

Якщо при заданому розмірі входу в якості міри складності береться найбільша із складностей (за всіма входами даного розміру), то її називають складністю в найгіршому випадку. Якщо ж вибирається середня складність алгоритму за даним розміром входу, то її називають середньою або усередненою. Цей тип часової складності алгоритму знайти важче, ніж складність у найгіршому випадку [3].

Постановка задачі та метод її розв'язування. У даній роботі досліджуються часові складності алгоритмів розкладання графів за їх вершинами залежно їх від способу подання графа. Зокрема, розглядаються точні алгоритми, тобто такі, які гарантують відшукання оптимального розфарбування вершин та хроматичного числа довільного графа та наближені, які не завжди знаходять точний розв'язок, але є більш ефективними.

Часом роботи алгоритму називатимемо число елементарних кроків, які він виконує. Вважатимемо, що елементарним кроком є один пункт алгоритму, але в тому випадку, коли він не містить вказівок такого роду, як «відсортувати масив за зростанням». [7] Насамперед, нас буде цікавити час роботи T таких алгоритмів у найгіршому випадку з огляду на наступне:

- Знаючи час роботи в найгіршому випадку, можна гарантувати, що виконання алгоритму закінчиться за деякий час, не знаючи, вхід якого розміру буде використовуватись.
- На практиці «погані» входи (для яких час роботи близький до максимального) можуть траплятися досить часто.
- Середній час роботи може бути дуже близьким до часу роботи в найгіршому випадку [4].

Структури даних, які використано для подання графів, – це матриці інцидентності та суміжності, списки суміжності та двійковий код графа [3].

Аналіз отриманих результатів. Спочатку розглянемо наближені алгоритми розкладання, а саме: наближений та покращений алгоритми послідовного розфарбування вершин графа.

Алгоритм послідовного розфарбування. Довільній вершині v_1 графа G припишемо колір 1. Якщо вершини $v_1, v_2, \ldots v_i$ розфарбовані k кольорами, $k \le i$, то новій довільно вибраній вершині v_{i+1} припишемо мінімальний колір, що не використовувався при побудові розфарбування суміжних вершин [10].

for $v \in V$ **do**

C[v]:=0 {усі вершини не пофарбовані}

end for

for $v \in V$ **do**

A:={1, …p} {усі кольори}

for $u \in \Gamma^+$ **do**

A:=A\{C[u]} {зайняті всі кольори}

end for

C[v]:=min A {мінімальний вільний колір}

end for

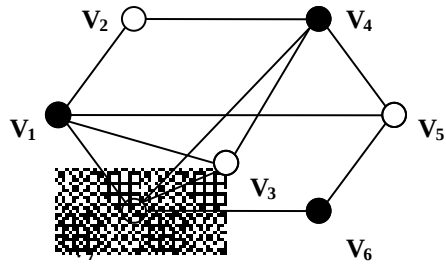


Рис. 1. Розкладання графа за допомогою алгоритму послідовного розфарбування

Знайдемо часову складність даного алгоритму, якщо граф G задається матрицею суміжності. Нагадаємо, що в такому випадку вершини графа G(V, E) нумерують числами 1, 2, … |V| та розглядається матриця $A=(a_{ij})$ розміру |V| × |V|, для якої

$$a_{ij} = \begin{cases} 1, (i\ \ j) \in E, \\ 0, (i\ \ j) \notin E. \end{cases}$$

1. Задати вхідну послідовність із n^2 елементів, де n – кількість вершин графа.
2. Вибрати довільним чином вершину. Наприклад, v_1.

Зауваження: Для зберігання розфарбування графа використаємо масив C, де номер елемента вказуватиме на номер вершини, а його вміст – на колір цієї вершини.

3. Для вибраної вершини v_1 приписуємо колір 1: C[1]:=1.
4. Перейти до наступної вершини, за даним випадком – v_2.
5. Переглядаємо всі суміжні v_2 вершини. Їх буде щонайбільше $(n-1)$.
6. У масиві C серед елементів з відповідними номерами знаходимо мінімальний елемент. Оскільки ми оцінюємо час роботи алгоритму в найгіршому випадку, то припустимо, що v_2 суміжна з усіма іншими вершинами графа. При пошуку мінімального кольору min переглянемо $(n-1)$ елемент.
7. Вершині v_2 приписуємо колір $(\min+1)$: C[2]:=min+1.
8. Перейти до кроку 2.

$$T_1(n) = n^2 + (n-1)((n-1)+n) = 3n^2 - 3n + 1.$$

Теорема. *Якщо* $A(k) = a_m k^m + \ldots + a_1 k^1 + a_0$ *є поліномом степеня m, тоді (*)*

$$A(k) = O(k^m).$$

Отже, часова складність вище описаного алгоритму становить $O(n^3)$.

Якщо граф G, який має n вершин та e ребер, задається матрицею інцидентності, то розмірність матриці становить n×e. Матриця $B=(b_{ij})$ формується із нулів та одиниць за наступним правилом [2]:

$$b_{ij} = \begin{cases} 1, \textit{якщо } v_i \textit{ інцидентна ребру } e_j \\ 0, \textit{у протилежному випадку} \end{cases}$$

У цьому випадку кількість елементарних операцій при виконанні послідовного алгоритму розфарбування графа становитиме $T_2(n)=2mn - m + 1$, а отже $T_2(n)$ – лінійна функція складності.

Наступний спосіб подання графа – двійковий код. Використовуючи матрицю суміжності графа, можна подати його у вигляді двійкового коду. Очевидно, якщо запам'ятати послідовність нулів та одиниць матриці, то за нею можна відновити весь граф. Так як матриця симетрична, то достатньо запам'ятати лише ті елементи, які розміщені над головною діагоналлю, тобто послідовність $a_{12}, a_{13}, a_{14}, \ldots a_{n-1\,n}$. Довжина цієї послідовності рівна $C_n^2 = \frac{n(n-1)}{2}$.

Число, яке отримаємо після виконання наступної операції

$$a_{1,2} \cdot 2^0 + a_{1,3} \cdot 2^1 + a_{2,3} \cdot 2^2 + a_{1,4} \cdot 2^3 + \ldots + a_{n-1,n} \cdot 2^{c_n^2 - 1},$$

назвемо двійковим кодом матриці. Оскільки різним нумераціям вершин графа відповідають різні матриці суміжності, то існує n! таких кодів. Найменший із них називають міні_кодом μ (G), а найбільший – максі_кодом μ (G)[2].

Таким чином, за умови подання графа за допомогою двійкового коду після переведення максі_коду μ (G) у двійкове число ми отримаємо матрицю суміжності графа. Оскільки виконання алгоритму за умови подання у вигляді матриці суміжності уже описано, то спробуємо визначити кількість операцій при перетворенні максі_коду μ (G) у двійкове число. Їх кількість становитиме $\frac{n(n-1)}{2}$.

Таким чином, часова складність алгоритму послідовного розкладання становитиме $O(n^3)$.

Одним із поширених способів подання графів є списки суміжності. Цей спосіб доцільно використовувати у випадку розріджених графів ($|E|$ значно менше від $|V^2|$). Подання графів у вигляді списків суміжності використовує масив Adj із $|V|$ списків – по одному на вершину. Для кожної

вершини v $\in$ V список суміжних вершин *Adj[v]* містить у довільному порядку (вказівники на) всі суміжні з нею вершини. Підрахунок кількості елементарних операцій, які здійснюються у ході виконання алгоритму за умови такого способу подання приводить до наступної оцінки складності

$$T_4(n)=n^3+2n^2+8n+3\lg n+24m+1.$$

Застосувавши теорему (*), отримали $O(n^4+3\lg n)$.

Покращений алгоритм послідовного розфарбування. Наступний алгоритм також будує допустиме розфарбування. Але на відміну від попереднього, розфарбування розпочинається із вершини з найбільшим степенем. Якщо ж їх розфарбовувати в останню чергу, то може виявитись, що для них немає вільного кольору [10].

```
Sort(v) {впорядкувати вершини за не зростанням степенів }
c:=1
for v ∈ V do
        C[v]:=0 {усі вершини не зафарбовані}
end for
while V≠ ø do
      for v ∈ V do
        for u∈ Γ+(v) do
              if C[u]=c then
                    next for v { вершину не можна пофарбувати в
                  колір c}
          end if
        end for
        c[v]:=c {зафарбовуємо вершину в колір c}
       V:=V\{v} {видаляємо вершину із переліку тих, які будуть розг-
       лядатись}
  end for
c:=c+1
```

end while

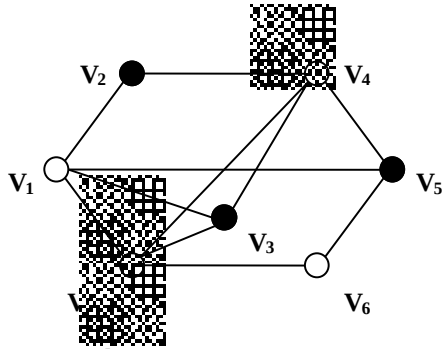


Рис. 2 Розкладання графа за допомогою покращеного алгоритму послідовного розфарбування

Оцінка часової складності проводиться методом, аналогічним до попереднього. Особливість даного алгоритму полягає в тому, що у ньому використовується сортування степенів вершин не по зростанню. Для його реалізації використано метод бульбашки, який використовує n^2 операцій, де n – кількість елементів, які сортують [7].

Оцінки часової складності покращеного алгоритму розфарбування вершин графа наступні:

Матриця суміжності – $O(n^4)$.

Матриця інцидентності – $O(n^4)$.

Двійковий код – $O(n^4)$.

Списки суміжності – $O(n^4+3\lg n)$.

Щодо точних алгоритмів розфарбування, то вони, на відміну від наближених, гарантують відшукання оптимального розфарбування вершин та істинного значення хроматичного числа довільного графа. [10]

Точний алгоритм розфарбування. Даний алгоритм базується на використанні незалежних множин для розкладання графа за його вершинами. Задача відшукання найбільшої незалежної множини вершин графа належить до класу трудомістких задач.

Основна ідея алгоритму полягає у використанні рекурсивного виклику процедури (в даному випадку вона носить назву P), що знаходить максимальну незалежну множину вершин графа та зафарбовує їх у допустимий

колір [10]. Множину вершин графа називають незалежною, якщо ніякі дві вершини у ній не є суміжними [1].

if V=ø **then**

return {розфарбування закінчено}

end if

s:=selectmax (G) {S – максимальна незалежна множина}

C[S]:=i {зафарбовуємо вершини множини у колір i}

P(G-S, i+1) {рекурсивний виклик}

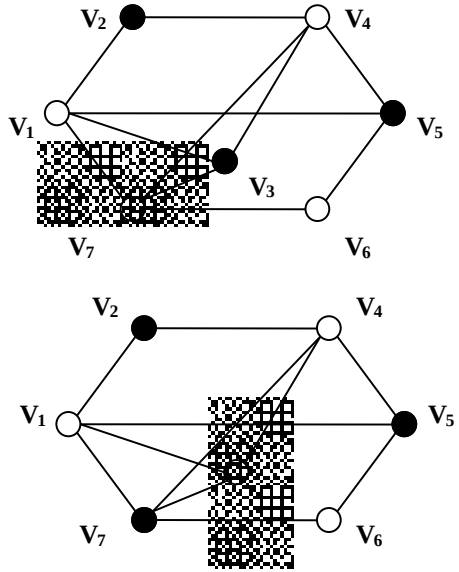


Рис. 3 Розкладання графа за допомогою покращеного алгоритму послідовного розфарбування

Для відшукання незалежних множин графа використаємо уже відомий алгоритм [10]. Побудова максимальної незалежної множини розпочинається із порожньої множини і у ході виконання алгоритму поповнюється вершинами із зберіганням незалежності.

k:=0 {кількість елементів у поточній незалежній множині}

S[k]:=ø {незалежна множина із k вершин}

$Q^-[k]$:=ø {множина вершин, що уже використані для розширення S[k]}

$Q^+[k]$:=V { множина вершин, які можна використати для розширення S[k]}

M1: {крок вперед}

select $v \in Q^+[k]$ {розширяюча вершина}

S[k+1]:=S[k] $\cup v$ {розширена множина}

$Q^-[k+1]:= Q^-[k] \setminus \Gamma[v]$ {вершина v використана для розширення}

$Q^+[k+1]:= Q^+[k] \setminus (\Gamma[v] \cup \{v\})$ {усі вершини, суміжні з v не можуть бути використані для розширення}

k:=k+1

M2:

for $u \in Q^-[k]$ **do**

if $\Gamma[u] \cap Q^+[k]= ø$ **then**

goto M3 {можна повертатись}

end if

end for

if $Q^+[k]:= ø$ **then**

if $Q^-[k]:= ø$ **then**

yield S[k] {множина S[k] максимальна}

end if

goto M3 {можна повертатись}

else goto M1 {можна іти вперед}

end if

M3: {крок назад}

v:=last(S[k]) {останній добавлений елемент}

k:=k-1

S[k]:=S[k+1] − {v}

Q^-[k]:= Q^-[k] ∪ {v}

Q^+[k]:= Q^+[k] \ {v} {вершина v уже додавалась}

if k=0 **and** Q^+[k]= ø **then**

stop {перебір завершено}

else goto M2 {перехід на перевірку}

enf if

Часову складність даного алгоритму оцінюватимемо разом із усіма операціями у алгоритмі.

Отже, часова оцінка точного алгоритму розкладання графа становить:

Матриця суміжності – $O(n^n)$.

Матриця інцидентності – $O(n^n)$.

Двійковий код – $O(n^n)$.

Списки суміжності – $O(n^4+8m+2m\lg n)$.

Алгоритм, що базується на використанні максимальних r-підграфів. Породжений підграф $<S_r[G]>$ графа $G=(X, Г)$, де $S_r[G] \subseteq X$, називають r-підграфом, якщо він r-хроматичний. Якщо не існує такої множини H, що $H \supset S_r[G]$ і підграф H є r-хроматичним, то $<S_r[G]>$ називають максимальним r-підграфом графа G.

Хроматичне число графа є найменшим значенням r, при якому $S_r[G]$ − множина вершин деякого максимального r-підграфа – співпадає із множиною вершин X графа. Тому процедури послідовної побудови r-підграфів можна використовувати для знаходження хроматичного індексу графа, якщо на кожному кроці процедури перевіряти, чи не міститься множина вершин графа в котромусь із знайдених r-підграфів [9]:

r:=1

Q:={$S^j_r[G] \mid j=1, \dots q_r$} {знаходимо максимальну множину r-підграфів}

label

k:=Selectmax(X − $S^j_r[G]$); {знаходимо максимальну незалежну множину}

if $k \neq ø$ **then**

$S := S^{j}_{r}[G] \cup S_1[G]$

if $S=X$ **then stop** {$(r+1)$ − хроматичне число, розфарбування знайдено}

else

case S **of**

$S \subseteq S'$ **then goto label** { $S' \in Q$}

$S \supset S'$ **then Q:=Q − {S}** { $S' \in Q$}

else $Q:=Q \cup \{S\}$ **goto** label

end case

end if

else

if $j<q_r$ **then** $j:=l+1$ **goto** label

end if

if $j=q_r$ **then** $j:=l+1$

$r:=r+1$

$q_r:=|Q|$ **goto** label

end if

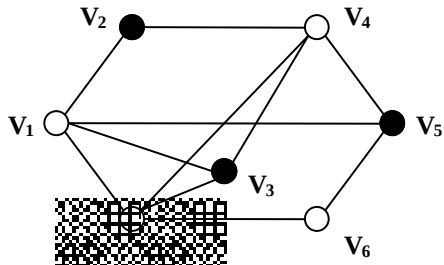


Рис. 4 Розкладання графа за допомогою покращеного алгоритму послідовного розфарбування

Для відшукання незалежних множин графа та його максимальних r-підграфів використано ту ж процедуру, що і для точного алгоритму. Ча-

сова складність алгоритму розкладання графа за допомогою його максимальних r-підграфів становить:

Матриця суміжності – $O(n^n + n^2 + c_1 n + c_2)$, де c_1, c_2 – сталі.

Матриця інцидентності – $O(n^n + nm + nc_1 + c_2)$, де c_1, c_2 – сталі.

Двійковий код – $O(n^n + n^2 + c_1 n + c_2)$, де c_1, c_2 – сталі.

Списки суміжності – $O(n^n + c_1 n + 8m + 2m \lg n)$, де c_1 – стала.

Слід зауважити, що останні два алгоритми використовують для побудови максимальних незалежних множин вершин алгоритм, який має переборний характер. Таким чином , кожен із цих алгоритмів в тій чи іншій мірі все ж містить перебір варіантів[10].

Далі розглянемо **алгоритм розкладання графа за допомогою його кістяків** [5] та спробуємо, як і вище, оцінити його часову складність у залежності від способу подання.

Ідея алгоритму полягає в наступному: для запропонованого графа будується кістяк. Він перетворюється у ході розв'язання задачі до тих пір, поки всі його вершини не задовольнятимуть відношення часткового порядку «<», що визначений на множині вершин. (Відношення часткового порядку «<» визначається наступним чином: вершина у < вершини z тоді і тільки тоді, коли найкоротший шлях від кореневої вершини x до вершини z проходить через вершину y.) Після того, коли такий кістяк знайдено, будується розфарбування самого графа. Обґрунтування даного методу міститься в [12].

Наближений алгоритм даного способу розкладання має такий вигляд.

Tr:=kraskala(G) {будуємо кістяк графа G за алгоритмом Краскала (можна вибрати довільний інший алгоритм)}

V_0:=SelectRoot {вибираємо кореневу вершину}

kistjak:=false;

repeat

BEGIN

{ побудова допоміжного масиву norm для визначення того, чи кістяк нормальний}

dejkstr(nn, ks, x, i, l1, k1, path); // процедура, що реалізує алгоритм Дейкстри відшукання мінімальної відстані між вершинами; x, i – номери вершин, відстань між якими обчислюється; l1 – допоміжна змінна; k1 – відстань між вершинами x та i; path – масив, що містить шлях від x до i.

```
dejkstr(nn, ks, x, j, l2, k2, path1);
    if path1[p]=i then
    norm[h].b11:=true else  norm[h].b11:=false;
```

{ітерація виправлення кістяка на нормальний; даний блок команд повторюється доки масив norm не міститиме жодного елемента b11=false }

```
if norm[i].b11 = false then
  dejkstr(nn, ks, x, norm[i].a11, l1, k1, path);
  dejkstr(nn, ks, x, norm[i].a12, l2, k2, path1);
  if k1<=K2 then
    j:=norm[i].a12;
    dejkstr(nn, ks, x, p, l1, k3, path);
    if (matrix[p, j]<>0) and (matrix[p, j]<>1000) and (ks[p, j]<>0)
and(k3<=k1-1) then
// видаляємо ребро (p, j)
      ks[norm[i].a11, norm[i].a12]:=1;
      ks[norm[i].a12, norm[i].a11]:=1;
      norm[i].b11:=true;
    else
    j:=norm[i].a12;
    dejkstr(nn, ks, x, p, l1, k3, path);
    if (matrix[p, j]<>0) and (matrix[p, j]<>1000) and (ks[p, j]<>0)
    and(k3<=k1-1) then
// видаляємо ребро (p, j)
      ks[norm[i].a11, norm[i].a12]:=1; // додаємо нове ребро
```

```
            ks[norm[i].a12, norm[i].a11]:=1;
            norm[i].b11:=true;
          {перевіримо, чи масив norm містить значення false}
          if norm[i].b11=false then kistjak:=false;
    until kistjak=false;
      {розфарбування отриманого нормального кореневого кістяка усіма можливими кольорами}
    hi[i]:= (k1)mod a;
  end.
```

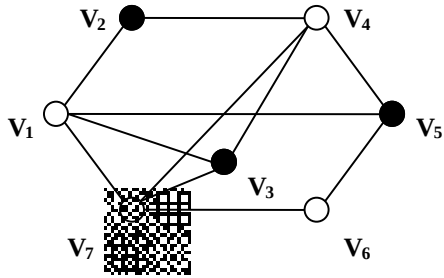


Рис. 5 Розфарбування графа за допомогою алгоритму розкладання графа за допомогою його кістяків

Оцінки часової складності описаного алгоритму наступні [6; 11]:

Матриця суміжності – $O(n^4)$.

Матриця інцидентності – $O(n^4+m \log m)$.

Двійковий код – $O(n^4)$.

Списки суміжності – $O((2n + 8m + 2m[\log_{10} n])^3)$.

Далі в таблиці подані оцінки часових складностей алгоритму розкладання графів.

Таблиця

Складність алгоритмів розкладання графів

АЛГОРИТМ	СПОСІБ ПОДАННЯ			
	Матриця суміжності	**Матриця інцидентності**	**Двійковий код**	**Списки суміжності**
1	**2**	**3**	**4**	**5**
Алгоритм послідовного розфарбування	$O(n^3)$	лінійна	$O(n^3)$	$O(n^2+n \lg n)$
Покращений алгоритм послідовного розфарбування	$O(n^4)$	$O(n^4)$	$O(n^4)$	$O(n^4+3\lg n)$
Точний алгоритм розфарбування	$O(n^n)$	$O(n^n)$	$O(n^n)$	$O(n^4+8m+2m\lg n)$
Алгоритм розфарбування з використанням максимальних r-підграфів	$O(n^n+n^2+nc_1+c_2)$	$O(n^n+nm+nc_1+c_2)$	$O(n^n+n^2+nc_1+c_2)$	$O(n^n+c_1n+8m+2m \lg n)$
Алгоритм розфарбування за допомогою кістяків	$O(n^4)$	$O(n^4)$	$O(n^4)$	$O((2n + 8m+2m[\log_{10} n])^3)$

Висновки: Оцінки часової складності алгоритмів розкладання графів залежно від способів подання графів свідчать про те, що запропонований алгоритм забезпечує не гіршу, а для точних алгоритмів навіть і кращу,

часову складність. Крім того, час роботи кожного із алгоритмів повністю залежить від кількості вершин і лише в кількох випадках (алгоритм розкладання за допомогою кістяків та максимальних r-підграфів) від кількості ребер графа. Отже, за умови подання графа за допомогою списків суміжності такий алгоритм доцільно використовувати для розріджених графів – в яких кількість вершин значно перевищує кількість ребер. Хоча для випадку алгоритму розкладання, що базується на r-підграфах, величини $2m$, nm та $8m$ (n – кількість вершин графа, m – кількість його ребер) не можуть суттєво впливати на ситуацію, оскільки яким би великим не було число ребер графа, його значно перевищуватиме число n^n.

Бібліографічні посилання

1. **Асанов М.О.** Дискретная математика: графы, матроиды, алгоритмы. / М.О. Асанов, В.А. Баранский, В.В. Расин – М., 2001.
2. **Асельдеров З.М.** Представление и восстановление графов. / З.М. Асельдеров – К., 1991.
3. **Ахо А.** Построение и анализ вычислительных алгоритмов. / А. Ахо, Дж. Хопкрофт, Дж. Ульман - М., 1979.
4. **Ахо А.** Структуры данных и алгоритмы. / А. Ахо, Дж. Хопкрофт, Дж. Ульман – М., 2000.
5. **Гришанович Т.О.** «Алгоритм розкладання графів» / Т. О. Гришанович // III-я Міжнародна науково практична конференція «Моделювання та комп'ютерна графіка 2009», Донецьк.
6. **Дегтярьов В.О.** Розробка методології прокладки комунікацій в САПР для складних технічних об'єктів. /В.О. Дегтярьов //Вестник ХНТУ. – 2005. – №1(21). – С. 190-195.
7. **Катленд Н.** Вычислимость. Введение в теорию рекурсивных функций. / Н. Катленд – М., 1990.
8. **Кормен Т.** Алгоритмы: построение и анализ. /Т. Кормен, Ч. Лайзерсон, Р. Риверст. – М., 2004
9. **Кристофидес Н.** Теория графов. Алгоритмический поход. / Н. Кристофидес – М., 1978.
10. **Новиков Ф.А.** Дискретная математика для программистов. / Ф.А. Новиков – С.Пб., 2000.
11. **Погорілий С.Д.** Формування та аналіз паралельних схем алгоритму Дейкстри. / С.Д. Погорілий, Ю.В. Бойко, Р.В. Білоус // Математичні машини і системи. – 2008. – №4. – С.59-65.

12. **Протасов І.В.** Розкладність графів: Навчальний посібник. / І.В. Протасов, К.Д. Протасова – К., 2003.

Надійшла до редколегії 12.12.09.

УДК 519.852.6:004.421.2:514.747

©**А.І. Зінченко**
Запорізький національний університет

ВИКОРИСТАННЯ ЕКВІДИСТАНТНИХ ФІГУР У ЗАДАЧАХ ОПТИМАЛЬНОГО РОЗКРОЮ

При розв'язанні задач оптимального розкрою необхідно враховувати, що деталі не можуть розташовуватися на листі впритул, що пов'язано з технологічними особливостями процесу штампування. Пропонується замість заданих фігур розглядати розширені фігури, які можна розташовувати впритул. Описано процес побудови розширених фігур і наведені приклади роботи програми, в якій реалізовано розроблений алгоритм.

При решении задач оптимального раскроя необходимо учитывать, что детали не могут распол агаться на листе вплотную, что связано с технологичес кими особенностями процесса штамповки. Предлагается вместо заданных фигур рассматривать расширенные фигуры, которые можно располагать вплотную. Описан процесс построения расширенных фигур и приведены примеры работы программы, в которой реализован разработанный алгоритм.

While solving the problem of optimal cutting it is necessary to take into account that details cannot locate on sheet closely that is connected with technological features of process of punching. It is offered to consider the expanded figures which it is possible to locate closely instead of the given figures. Process of construction of the expanded figures is described and examples of work of the program in which the developed algorithm is realised are resulted.

Ключові слова: задача оптимального розкрою, еквідистантні фігури, процес штампування.

Одним з основних шляхів зниження витрат листового прокату на автомобільних, комбайнових та тракторних заводах є вдосконалювання технологічних процесів розкрою. При цьому найбільш істотне значення для зниження витрат має етап пошуку оптимального розташування плоских деталей у прямокутних листах або рулонах.

Кількість наукових публікацій за цією тематикою останнім часом швидко зростає. Вичерпну інформацію можна знайти на сайті ESICUP (EURO Special Interest Group on Cutting and Packing) [1] Існує широкий

спектр методів розв'язування таких задач: класичні (методи лінійного цілочислового та нелінійного програмування, наприклад, методи, що ґрунтуються на теорії R-функцій [2]), і розроблені відносно недавно (генетичні алгоритми [3], метод Лагранжевих релаксацій [4], пошук із заборонами [5], метод комашиної колонії [6], і т. д.). Огляд сучасного стану проблеми можна знайти в [7; 8].

Через особливості процесу штампування, деталі на листі не можуть розташовуватись впритул, а повинні знаходитись одна від одної на відстань, яка не менша ніж задана міждетальна перемичка. Зауважимо, що майже всі методи вказані у попередньому абзаці шукають такі варіанти розкрою, при якому фігури щільно прилягають одна до одної. Дана робота пропонує замість заданих фігур розглядати еквідистантні їм фігури. Параметр еквідистантності зручно брати рівним половині заданої міждетальної перемички. Розширені фігури вже можна укладати на листи щільно і таким чином можна буде застосовувати апробовані алгоритми.

Побудова коду заданої та розширеної фігур. На рис.1 зображено приклад побудови еквідистантної фігури (пунктир) для вихідної фігури (неперервна лінія).

Контур реальних заготовок для штампування зазвичай складається з відрізків прямих та дуг кіл. Ці геометричні об'єкти будемо називати «лінійними елементами», а точки, які є спільними для сусідніх лінійних елементів – кутовими точками. Нехай n кількість елементів контуру фігури. Елементи будемо нумерувати від 1 до n, рухаючись вздовж контуру фігури проти ходу годинникової стрілки.

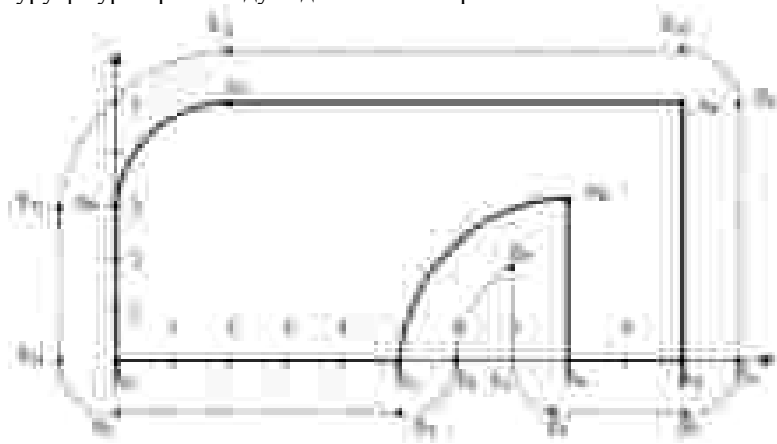


Рис.1. Еквідистантні фігури

Віднесемо фігуру до правої декартової системи координат з осями X та У. Кожний лінійний елемент контуру фігури будемо кодувати послідовністю, яка складається з трьох чисел. Якщо елемент контуру є відрізком прямої, то відповідна трійка чисел має вигляд (0, X, Y), де 0 – ознака відрізка, а точка з координатами (X, Y) є початковою точкою відрізка. Якщо ж елемент контуру є дугою кола, то трійка має вигляд ($\pm R, X, Y$), де R – радіус кола. Знак перед R має наступний сенс: «+» відповідає опуклій дузі кола, «–» відповідає угнутій дузі кола. Точка з координатами (X,Y) є початковою точкою дуги кола. Як приклад, наведемо код фігури, зображеної на рис 1. (0, 0, 0; -3, 5, 0; 0, 8, 3; 0, 8, 0; 0, 10, 0, 0, 10, 5, 2, 2, 5; 0, 0, 3)

Для подальшого будемо позначати елементи цієї послідовності чисел через p_i, а саму послідовність літерою P. Наприклад $p_1 = 0, p_4 = -3$. Елементи послідовності, які описують контур еквідистантної фігури, будемо позначати через q_i, а саму послідовність – літерою Q.

Розглянемо спосіб побудови еквідистнатної фігури для фігури на рис.1 для $\varepsilon = 1$.

Якщо позначити через $A_k(x_k, y_k)$ k- ту вершину контуру заданої фігури, то із попередньої домовленості випливає, що $x_k = p_{3k-1}, y_k = p_{3k}$. Будемо розрізняти вхідні кути (кут з вершиною в A_3, A_7, A_8), та вихідні кути (кути з вершинами в A_1, A_2, A_4, A_5, A_6). Розгорнутий кут зручно вважати вхідним.

Перейдемо до знаходження коду розширеної фігури за відомим кодом вихідної фігури. Оскільки $p_1 = 0$, то лінійний елемент A_1A_2 є відрізком. Відповідним лінійним елементом еквідістантної фігури буде відрізок, B_1B_2. Це означає, що $q_1 = 0$. Щоб визначити значення членів q_2 та q_3 послідовності q знайдемо координати точки B_1. Вектор $\overline{A_1B_1}$ можна отримати, якщо вектор $\overline{m} = (m_1, m_2) = \varepsilon \dfrac{\overline{A_1A_2}}{\left|\overline{A_1A_2}\right|}$ повернути на прямий кут за годинниковою стрілкою. Щоб отримати координати точки B_1 потрібно до координат точки A_1 додати координати вектора $\overline{A_1B_1}$:

$$\overline{A_1A_2} = (p_5 - p_2, p_6 - p_3) = (5 - 0{,}0 - 0) = (5{,}0),$$

$$\overline{m} = 1 \cdot \frac{1}{5}(5{,}0) = (1{,}0),\ \overline{A_1B_2} = (m_2, -m_1) = (0, -1).$$

Щоб отримати координати точки B_1 потрібно до координат точки A_1 додати координати вектора $\overline{A_1B_1}$:

$$B_1(q_2, q_3) = (p_2, p_3) + (m_2, -m_1) = (0{,}0) + (0, -1) = (0, -1).$$

Таким чином, $q_2 = 0, q_3 = -1$, і початок послідовності q матиме вигляд $0{,}0{,}-1{,}....$

Оскільки $p_1 = 0$, а $p_4 = -3$, то маємо кут між відрізком та дугою кола. Визначимо тип кута між першим і другим лінійними елементами контуру заданої фігури. Для цього спочатку визначимо координати точки C – центру кола з дугою A_2A_3. Оскільки $p_4 < 0$, то це коло є угнутим. Позначимо через D – середину відрізка A_2A_3 (рис 2). Координати цієї точки такі:

$$(x_d, y_d) = \left(\frac{x_2 + x_3}{2}, \frac{y_2 + y_3}{2}\right) = \left(\frac{p_5 + p_8}{2}, \frac{p_6 + p_9}{2}\right) = \left(\frac{5+8}{2}, \frac{0+3}{2}\right) = \left(\frac{13}{2}, \frac{3}{2}\right)$$

.

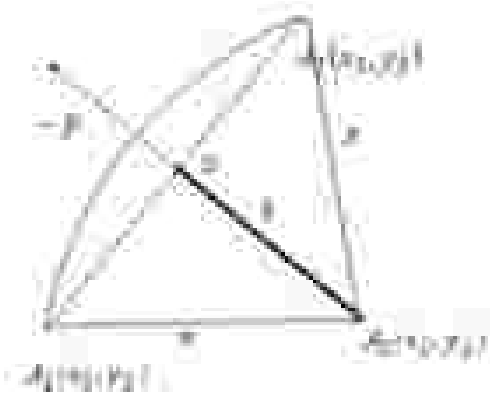


Рис.2. До задачі визначення центра кола

За теоремою Піфагора

$$\left|\overline{DC}\right| = \sqrt{R^2 - \overline{A_2D}^2} = \sqrt{p_4^2 - (x_d - x_2)^2 - (y_d - y_2)^2} = \sqrt{9 - \frac{9}{4} - \frac{9}{4}} = \frac{3}{\sqrt{2}}.$$

Знайдемо тепер координати одиничного вектора $\bar{r}$ хорди A_2A_3.

$$\bar{r} = \frac{\overline{A_2A_3}}{\left|\overline{A_2A_3}\right|} = \frac{1}{\left|\overline{A_2A_3}\right|}(p_8 - p_5, p_9 - p_6) = \frac{1}{\sqrt{3^2 + 3^2}}(3,3) = \frac{1}{\sqrt{2}}(1,1).$$

Якщо вектор $\bar{r}$ помножити на $\left|DC\right| = \frac{3}{\sqrt{2}}$ і повернути на 90 ^{0} за рухом годинникової стрілки (у випадку опуклої дуги, потрібно було б обертати проти годинникової стрілки), то отримаємо вектор $\overline{DC}$.

Тобто $\overline{DC} = \frac{3}{\sqrt{2}}\frac{1}{\sqrt{2}}(1,-1) = \frac{3}{2}(1,-1)$.

Координати точки C знайдемо, якщо до координат точки D додамо координати вектора $\overline{DC}$: $\left(\frac{13}{2}, \frac{3}{2}\right) + \left(\frac{3}{2}, -\frac{3}{2}\right) = (8,0)$. Таким чином, центром кола з дугою A_2A_3 є точка $C(8,0)$.

Вектор $\overline{CA_2}(-3,0)$ повернемо на прямий кут вздовж руху годинникової стрілки. Будемо мати вектор $\bar{n}(0,3)$. Якщо вектори $\overline{A_1A_2}(5,0)$ та $\bar{n}(0,3)$ перенести в одну точку, то найкоротший поворот від першого з цих векторів до другого буде здійснюватися проти годинникової стрілки. Це випливає з того, що число $\begin{vmatrix} 5 & 0 \\ 0 & 3 \end{vmatrix}$ є додатним (це число є аплікатою векторного добутку цих векторів, якщо їх розглядати як вектори тривимірного простору, які лежать у площині $z = 0$). За умови, що система координат $Oxyz$ є правою. А це означає, що кут $A_1A_2A_3$ є вихідним, а значить еквідистантна фігура буде містити опуклу дугу B_2B_3 радіуса $\varepsilon = 1$, і $q_4 = 1$.

Координати точки B_2 знайдемо, якщо до координат точки A_2 додамо координати вектора $\overline{A_2B_2} = \overline{A_1B_1} = (0,-1)$. Будемо мати $B_2(5+0,0-1)=(5,-1)$.

Таким чином, $q_5 = 5, q_6 = -1$, і послідовність Q приймає вигляд $0,0,-1,1,5,-1,...$

Дуга B_3B_4 є угнутою дугою кола радіус якого на ε менший за радіус дуги A_2A_3. Тобто $q_7 = -(|p_4| - \varepsilon) = -(3-1) = -2$.

Координати точки B_3 можна знайти, якщо до координат точки A_2 додати координати вектора $\frac{\varepsilon}{|\overline{A_2C}|}\overline{A_2C} = \frac{1}{\sqrt{3^2+0^2}}(3,0) = (1,0)$. Отримаємо $B_3(5+1,0+0)=(6,0)$, і відповідно $q_8 = 6, q_9 = 0$.

Послідовність Q на даний час приймає вигляд $0,0,-1,1,5,-1,-2,6,0,...$

Точка A_3 є кутовою точкою для дуги A_2A_3 і відрізка A_3A_4. Вектор $\overline{CA_3}(0,3)$ повернемо на прямий кут за рухом годинникової стрілки. Будемо мати вектор $\bar{n}(3,0)$. Якщо вектори $\bar{n}(0,3)$ та $\overline{A_3A_4}(p_{11} - p_8, p_{12} - p_9) = (0,-3)$ перенести в одну точку, то найкоротший поворот від першого з цих векторів до другого буде здійснюватися за рухом годинникової стрілки. Це випливає з того, що число $\begin{vmatrix} 3 & 0 \\ 0 & -3 \end{vmatrix}$ є від'ємним. Тому точка B_4 також є вершиною вхідного кута. Оскільки елемент A_3A_4 є відрізком прямої, то елемент B_4B_5 також є відрізком прямої. А отже $q_{10} = 0$. Координати точки B_4 знайдемо з умови, що вона належить колу з центром у точці $C(8,0)$ і радіусом $|q_7| = 2$, і віддалена на відстань $\varepsilon = 1$ від прямої A_3A_4. Такій умові задовольняють чотири точки на площині. Нам потрібно обрати ту, сума відстаней від якої до точок B_3 та A_3 найменша. Рівняння прямої A_3A_4:

$\frac{x-x_3}{x_4-x_3}=\frac{y-y_3}{y_4-y_3}$, або, після підстановки чисельних значень $\frac{x-8}{0}=\frac{y-3}{-3}$. Загальне рівняння цієї прямої $x-8=0$. Відстані від точки $B_4(X,Y)$ до цієї прямої $\varepsilon=1=\frac{|X-8|}{\sqrt{1}}$. Підводячи до квадрату, будемо мати $(X-8)^2=1$.

Умова, що ця точка належить колу с центром у точці $C(8,0)$ і радіусом $|p_7|=2$ аналітично записується у вигляді $(X-8)^2+Y^2=4$.

Отримаємо систему

$$\begin{cases}(X-8)^2=1,\\(X-8)^2+Y^2=4.\end{cases}$$

Розв'язавши систему, будемо мати наступні розв'язки. $(7,\sqrt{3}),(7,-\sqrt{3}),(9,\sqrt{3}),(9,-\sqrt{3})$.

Серед них умові мінімальності відповідає точка $(7,\sqrt{3})$. Отримаємо $B_4(7,\sqrt{3})$, і, отже $q_{11}=7, q_{12}=\sqrt{3}$.

Послідовність Q тепер приймає вигляд $0,0,-1,1,5,-1,-2,6,0,0,7,\sqrt{3}...$

Продовжуючи процес визначення коду еквідистантної фігури, остаточно отримаємо послідовність:

(0,0,−1;1,5,−1;−2,6,0;0,7, $\sqrt{3}$;3,1,7;0,0,8;−1,1,9;0,0,11;0,1,11;5,11,6;3,2,6;0, −1,3; 1,−1,0)

Зауважимо, що кількість елементів послідовності, що відповідає еквідистантній фігурі, на величину $3s$ (s – кількість вихідних кутів заданої фігури) перевищує кількість елементів послідовності, що відповідає заданій фігурі.

Застосування еквідистантних фігур у задачах пошуку оптимального регулярного однорядного розкрою прямокутних листів. Розроблений спосіб побудови контуру фігури еквідистантної заданій є складовою частиною алгоритму пошуку оптимального регулярного

однорядного розкрою прямокутних листів на однотипні заготовки. Сам алгоритм складається з наступних пунктів:

1. Визначення коду контуру фігури.
2. Визначення площі заданої фігури по її коду.
3. Побудова коду розширеної еквідистантної фігури з параметром еквідистантності ε.
4. Обчислення габаритів розширеної фігури вздовж осей x і y.
5. Обчислення ширини смуги, в якій розміщається один ряд деталей.
6. Обчислення кількості смуг у листі.
7. Визначення кроку штампування (він дорівнює довжині фігури в напрямку осі Ox).
8. Обчислення кількості фігур у смузі.
9. Обчислення кількості фігур у листі.
10. Визначення контуру розширеної фігури після повороту її на кут Δ.
11. Повторення пунктів 4–10 до тих пір, поки фігура не повернеться до початкового положення.
12. Вибір такого кута ϕ, при якому кількість фігур у листі буде максимальною.

Цей алгоритм реалізовано у програмі, яка дозволяє розв'язувати задачі оптимального розкрою. Нижче на рисунках наведено оптимальні карти розкрою для однієї і тієї ж фігури при різних розмірах листа.

Рис 3. Оптимальне розміщення фігур при ширині листа 200 одиниць

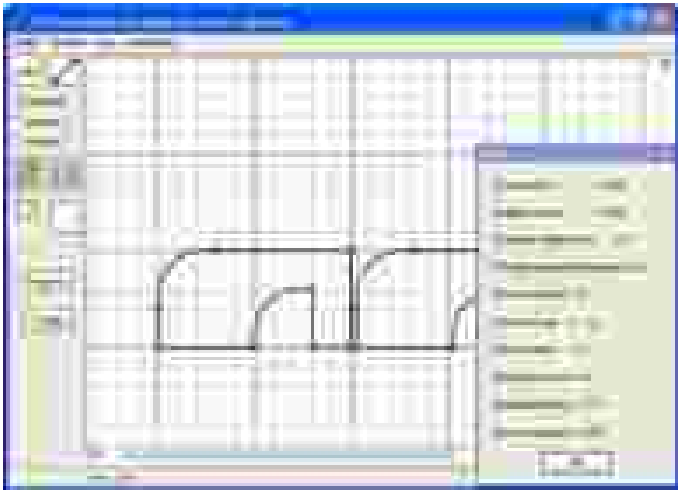


Рис. 4. Оптимальне розміщення фігур при ширині листа 400 одиниць

Рис. 5. Оптимальне розміщення фігур при ширині листа 666 одиниць

Як можна бачити на рис. 3-5 , оптимальний кут повороту може суттєво змінюватися при зміні розмірів прямокутного листа. Звернемо увагу на такий цікавий факт. У випадку розкрою, наведеного на рис 3 із листа отримали 5 деталей. У випадку розкрою, наведеного на рис 4, із листа отримали 12 деталей. При цьому довжина листа не змінювалася, а

ширина була збільшена у два рази. Таким чином, при збільшенні площі листа вдвічі кількість деталей, які можна отримати з нього, збільшилась у 2,4 рази.

Висновки та перспективи. Застосування еквідистантних фігур дозволяє задачу пошуку оптимального розташування фігур при наявності міждетальної перемички звести до задачі пошуку оптимального варіанта щільного укладання. У випадку, коли контур складається з відрізків та дуг кіл, задача знаходження координат кутових точок еквідистантної фігури розв'язується за допомогою апарату аналітичної геометрії. Отримані у статті формули входять до алгоритму пошуку оптимального варіанта однорядного регулярного розкрою листового матеріалу на однотипні заготовки, який реалізовано у вигляді програмного продукту.

У подальшому планується результати, отримані у статті, узагальнити для випадку коли елементи вихідного контуру є кривими, відмінними від відрізків прямих та дуг кіл.

Бібліографічні посилання

1. http://paginas.fe.up.pt/~esicup/tiki-index.php
2. **Рвачев В.Л.** Теория R-функций и некоторые ее приложения/ В.Л. Рвачев – К., 1982.- 552 с
3. **Скобцов Ю.А.** Решение задачи раскроя на основе генетического программирования /Ю.А. Скобцов, А.М. Фонотов // Вестник Херсонского государственного технического университета. – 2003. – № 2(18). – С. 137-142.
4. **Литвинчев И.С.** Улучшение лагранжевых оценок в задачах оптимизации / И.С. Литвинчев // Журнал вычислительной математики и математической физики, – Т 47, № 7, Июль 2007, С. 1151-1157.
5. **Glover, F., Laguna, M**., Tabu search. Kluwer Academic Publishers, –Boston, 1997.
6. **Levine, J., Ducatelle, F**., Ant colony optimization and local search for bin packing and cutting stock problems. Journal of the Operational Research Society, 55(7), 2004: 705-716.
7. **Скобцов Ю. А.** К вопросу о применении метаэвристик в решении задач рационального раскроя и упаковки / Ю. А. Скобцов, В. Н. Балабанов// Вісник Хмельницького національного університету. — 2008. — Т. 1, № 4. — С. 205—217.
8. **Гоменюк С.И.** Методы описания геометрических объектов / С.И. Гоменюк, Д.Н. Морозов, Ю.А. Сысоев, А.А. Лисняк // Вестник ЗНУ, – 2008, Т.1.с.36–44

Надійшла до редколегії 01.06.10

УДК 004.942

©**А.В. Иванченко, И.П. Гамаюн**
Национальный Технический Университет «Харковский политехнический институт»

ИДЕНТИФИКАЦИЯ ЗНАНИЙ В КОЛЛЕКТИВЕ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПОМОЩИ АППАРАТА НЕЧЕТКИХ СОЦИАЛЬНЫХ СЕТЕЙ

Запропонований метод ідентифікації неформальних знань в колективі розробників програмного забезпечення. Показано практичне використання підходу соціальних мереж для опису взаємодії співробітників в процесі реалізації проекту, а також використання міри упевненості у виборі для опису вхідних даних.

Предложен метод идентификации неформальных знаний в коллективе разработчиков программного обеспечения. Показано практическое использование подхода социальных сетей для описания взаимодействия сотрудников в процессе реализации проекта, а также использование меры уверенности выбора для описания входных данных.

The method of informal knowledge authentication in a software developer team is offered. The practical using of the social networks approach for a informal interaction between employees in a developer process and certainty measure using for input data describing are shown.

Ключевые слова: Идентификация знаний, Социальная сеть, Индекс Ямашита, Нечеткая социограмма.

Постановка проблемы. В статье «Проблема выявления полезных знаний при реализации программных проектов» показана проблема влияния субъективных знаний сотрудников на время реализации программных проектов. [1] Другими словами, в процессе создания программного обеспечения, в частности, программных модулей систем эксплуатационной поддержки телекоммуникационного оператора, осуществляется определенный набор работ $i, i \in I = \overline{1,n}$. В коллективе разработчиков можно выделить специалистов в каждой из работ, при том, отдельные сотрудники больше будут разбираться в некоторых вопросах, чем другие. Метод управления знаниями «COOLISON & PARCELL» в вопросе управления

субъективными знаниями предлагает сосредоточиться на управлении средой, в которой находятся сотрудники, обладающие определенными ценными знаниями для осуществления поставленным и профессиональным целям, а также объединить данных сотрудников в сети. [2] В то же время, стандартные подходы к формализации объективных профессиональных знаний не дают возможности учета неявных знаний, формирующихся в процессе человеческого общения. Подход социальных сетей был выбран для описания взаимодействия сотрудников в процессе реализации поставленных рабочих целей. Ввиду классических жестких двухполюсных способов описания коммуникаций (формой выбора элемента группы), в выше упомянутой статье был предложен подход описания данных взаимодействий с указанием дополнительной информации – степени уверенности в выборе. Применительно к контексту описания знаний данный параметр будет показывать степень участия другого сотрудника в процессе получении от него необходимой информации. Т. е. в анкете респондент указывает степень удовлетворенности в полученной от носителя информации, или, например, степень участия коллеги в решении конкретных задач по работам. Степень, оцененная в один бал, является максимальной.

В данной статье ставится задача реализации практического примера описанного подхода.

Обзор литературы. Начиная с 30 -х годов XX века проблемой описания взаимосвязи людей, а впоследствии – пробелами социальных сетей и их анализом занималось множество исследователей, среди которых можно выделить работы Я.Л. Морено, Дж. Барсена, Л. Фримана, Д. Ноука, П. Марсдена, С. Вассермана, Б. Веллмана, С. Берковица, а также А.А. Давыдова, Г.В. Градосельской и др.

В последние годы получила развитие концепция Social Networks Mining (добыча социальных сетей), как решение проблематики получения информации из социальных сетей, а именно из Large-Scale Web Social Networks, что отражено в работах международных конференций International Conference on Advanced in Social Networks Analysis and Mining (http://www.asonam.org), First International Workshop on Mining Social Networks for Decision Support (http://im.nuk.edu.tw/~iting/MSNDS2009 /index.php), International Workshop on Social Networks Mining and Analysis for Business Applications (SNMABA2009) (http://im.nuk.edu.tw/~iting/ SNMABA2009), 1st International Conference on Computational Collective Intelligence - Semantic Web, Social Networks & Multiagent Systems (ICCCI2009)(http://isccci.org/iccci-09).

В статье А. Давыдова «Системная социология: Social Networks Mining» данная концепция представляется как системная интеграция Social Net-

works Analysis (анализа социальных сетей), Multi-Agent-Based Social Simulations (MABSS) (компьютерного моделирования многоагентных систем), которое включает в себя множество компьютационных стандартов MABSS, методологий и языков моделирования, алгоритмов моделирования и имитационных моделей, платформ и компьютерных систем (http://jasss.soc.surrey.ac.uk/12/2/2.html) для проведения имитационного моделирования, анализа, визуализации и т. д.; технологии, методологии и методы Knowledge Discovery and Data Mining (KDD), в частности, Mining Graph Data; Web Mining; Text Mining , в частности, Opinion Mining; Multimedia Mining; Spatial Mining; Temporal Mining; Streams Mining и т.д.

Использование многолетних разработок в описанных сферах дает отличный инструментарий для решения задачи идентификации знаний в коллективе разработчиков программного обеспечения.

Сбор данных. Согласно предложенного Х. Ямашита подхода, процесс составления социограммы (социоматрицы) дополняется дополнительной оценкой респондентов, показывающей степень уверенности в выборе другого члена группы в некотором контексте опроса. Данный дополнительный параметр позволяет перейти от жесткого «да» или «нет». Предложенный подход использования аппарата нечеткой логики к реализации описанной оценки респондентов не противоречит традиционным методам анализа структуры взаимоотношений малой социальной группы и позволяет расширить представление данных до т. н. нечеткого графа, нечеткой социоматрицы и нечеткого социограммы.

Нечеткая логика. Основные определения. Основы нечеткой логики были заложены в конце 60_х годов в трудах ученого Лотфи А. Заде.

В основе алгебры нечеткой логики лежат два основных понятия: нечеткого множества и нечетких операций над ними. [3] При этом считается, что логика мышления при принятии решений человеком отлична от двоичной и многозначной логики, так как она имеет дело с нечеткими, размытыми классами понятий и отношений человеческого языка.

Опишим ряд определений, которые будут использованы.

Определение 1. Нечеткое подмножество А универсального множества U характеризуется функцией принадлежности f(u;A), которая ставит в соответствие каждому элементу «u» число f(u;A) из отрезка [0;1].

Определение 2. Лингвистической переменной называют переменную, принимающую в качестве своих значений нечеткие множества.

В ситуациях, связанных с оценкой истинности некоторого утверждения «p» через употребление человеком выражений «очень верно», «близко к истине» и др., целесообразным считается трактовать оценку через лингвистическую переменную с соответствующими значениями. Каждое такое

значение моделируется с помощью соответствующей функции принадлежности.

Принято значение истинности утверждения «p» обозначать через v(p), выраженное в виде v(p)=f(u;A).

В нечеткой логике операции: дизъюнкция (or), конъюнкция (and), отрицание (nod) и ипликации (=>) обозначаются и определяются следующим образом: v(p or q)=max(v(p),v(q)), v(p and q)=min(v(p),v(q)), v(not p)=1 − v(p) и v(p = >q)=min(1,1−v(p)+v(q)).

Определение 3. Нечеткое отношение R на универсальном множестве Z характеризуется функцией принадлежности R(x,y), число R(x,y) из отрезка [0;1]. При этом задать нечеткое бинарное отношение R на Z означает сначала указать пары элементов (x,y), определенно связанных отношением R, затем пары, которые не связаны с данным отношением и пары, имеющие промежуточные градации принадлежности или связаны с этим отношением.

Нечеткие бинарные отношения удобно задавать в виде матриц, называемых нечеткими матрицами.

Нечеткие операции над нечеткими отношениями: дизъюнкция, конъюнкция, отрицание и импликация определяются также как и для значений истинности в нечеткой логике. Для моделирования операций конъюнкции в нечеткой логике применяют т. н. треугольные нормы. [Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. – М, 1981]

Определение 4. Треугольной формой называется бинарная операция вида
T(x,y): [0,1]x[0,1]→[0,1] и удовлетворяющая следующим аксиомам: b≤c→T(a,b)≤T(a,c) и T(b,a)≤T(c,a) (монотонность); T(T(a,b),c) = T(a,T(b,c)) (ассоциативность); T(a,b) = T(b,a) (коммутативность); T(a,1) = T(1,a) = a (граничное условие).

Определение 5. Треугольная норма T (t -норма) является архимедовой, если она непрерывна и для любого нечеткого множества A выполнено неравенство T(A,A)<A. Она называется строгой, если функция T строго возрастает по обоим аргументам.

Для *архимедовой* t- нормы имеет место следующее функциональное представление:

$$T(x,y) = \frac{1}{\min(f(0), f(x) + f(y))}, \qquad \text{где } f: [0,1]\to[0,\infty].$$

Примерами t-норм являются:

$$T(x,y)=xy; \quad (1)$$

$$T(x,y)=\min(x,y); \quad (2)$$

$$T(x,y)=\max(0,x+y-1); \quad (3)$$

$$T(x,y)=\begin{cases} x, \text{åñëè } y=1 \\ y, \text{åñëè } x=1 \\ 0, \text{èíà÷å}. \end{cases} \quad (4)$$

Определение 6. Оператором осреднения называется непрерывная, неубывающая функция двух переменных M: [0,1]x[0,1]→[0,1], M(0,0)=0, M(x,x)=x, min(x,y)<M(x,y)<max(x,y). Примеры:

$$M(x,y)=0.5(x+y) \ (5); \ M(x,y)=0.5(1/x+1/y); \quad (6)$$

$$M(x,y)=\sqrt{xy}. \quad (7)$$

Сбор первичной информации. Для сбора первичной информации с целью изучения структуры в малой социальной группе применяют анкеты, содержащие один или несколько социометрических вопросов или критериев. Далее полученная информация обрабатывается на последовательности шагов для получения нечеткой социометрической матрицы.

Шаг 1. Полученные результаты от респондентов представляются в виде матрицы $K=(K_{ij}), i=1,L; \ j=1,L;$ L – количество респондентов.

Шаг 2. Переходят к оценочной матрицы $R=(R_{ij}), R_{ij}=N-K_{ij}+1, R_{ij}>0$, где $N=[\sum_{1}^{L}(\max K_{ij}/(L+1))+0.5]$ – используется для перехода от шкалы исходных степеней предпочтения выбора респондентов, которые были указаны в анкете.

Шаг 3. Формируют нечеткую социометрическую матрицу F=(Fij), Fij=Rij/N, 1>=Fij>0, Fij=1, если i=j.

Анализ социометрических данных. Нечеткую социометрическую матрицу A дополняют индексом Gij, учитывающим степень взаимности выборов, который рассчитывается по формуле

$$Gij = \frac{Fij \times Fji}{0.5(Fij + Fji)}.$$

Как видно из выражения характеристика степени взаимности выбора Gij выражается через треугольную норму T(x,y) и операторов осреднения M(x,y): $Gij = T(Fij, Fji) / M(Fij, Fji)$, если $M(Fij, Fji) \neq 0$ и нулевое значение если $M(Fij, Fji) = 0$. В качестве треугольной формы выбрана (1), а в качестве оператора осреднения (5) . Данный индекс Gij был предложен X. Ямашита. [4] Исходя из свойств представленных математических выражений делаются следующие выводы:

- если оба значения Fij, Fji равны единице, то индекс Ямашиты G примет значение единицы. Что означает о наличии полного взаимного предпочтения между рассматриваемыми двумя членами группы;
- если оба значения Fij, Fji будут положительными и меньшими единицы, то индекс равен нулю. Во всех других случаях 0<Gij<1. Т. е. возможно выражение с помощью данного индекса жестких требований по какому_либо отношению (например, влиянию);

Также в такой ситуации выбора, кроме наличия взаимности выбора не равного нулю, среди двух членов группы, по крайней мере один является лидером, что говорит о наличии у данного члена группы определенных знаний в данной ситуации.

Рассчитанные степени взаимности выбора формируют нечеткую матрицу Gij, на основе которой строится нечеткая дендрограмма P, являющаяся одним из инструментов кластерного анализа, позволяет визуализировать характер образования подструктур в зависимости от проявления интенсивности степени взаимности выбора.

Общий алгоритм построения нечеткой дендрограммы.

Шаг 1. Задаются уровни интенсивности степени взаимности выбора. Или разделяют единичный интервал на равные части, или упорядочивают значения Gij элементов нечеткой матрицы G в порядке возрастания.

Шаг 2. Выбирается наибольшая из степеней.

Шаг 3. Все значения матрицы G, равные или большие этому значению степени, заменяются на единицу, а те которые меньше обнуляются.

Шаг 4. В реконструированной матрице G выделяют подструктуры.

Шаг 5. Выбирают следующую по иерархии степень взаимности выбора предпочтения.

Шаг 6. Переходят к шагу 3.

Графическое изображение динамики формирования подструктур в зависимости от роста значений степени выбора представляет собой нечеткую дендрограмму.

Использование других видов треугольных норм позволяет расширить получаемые оценки при анализе малой социальной группы.

Нечеткая социограмма. Окончательные результаты удобно представить в виде нечеткой социограммы. [5, 6]

Нечеткая социограмма H представляет собой совокупность социограмм Hq, где q – обозначает степень взаимности выбора. Из такого определения можно выделить алгоритм построения нечеткой социограммы:

- выбирается наибольшее значение q;
- в соответствии с этим значением из нечеткой дендрограммы отбираются соответствующие подструктуры;
- элементы, образующие подструктуру, изображаются одинаковыми геометрическими фигурами;
- взаимосвязи между различными элементами подструктур строятся на основании нечеткой социометрической матрицы Аю

Недостатком данного подхода является его сложная формализация ввиду нестандартного и субъективного процесса построения.

Аналогично строятся социограммы для остальных значений q.

Преимущество построения такой социограммы в том, что в ней почти полностью используется полученная первичная информация, содержащаяся в социоматрице. Сведения, содержащиеся в нечеткой социограмме H=(Hq) значительно расширяют информационное обеспечение задачи моделирования процессов, происходящих в малых группах.

Пример расчета. Изучаемой группой является коллектив разработчиков программного обеспечения в некоторой прикладной области в составе из пяти человек. Согласно контексту выполнения программных проектов в данной группе находятся участники, обладающие различными ролями необходимыми для достижения цели проекта. Тем не менее, специфика реализации конкретной работы, выполняемой ранее, привела к тому, что некоторые из них в большей степени участвовали в исполнении, и, значит, приобрели некоторый опыт и знания. Ставится задача выявления данных знаний для дальнейшего использования в принятии управленческих решений.

Предложенные респондентам анкеты содержали вопрос «К кому из членов коллектива Вы обращались в процессе реализации некоторого задания?», степень оценки отражала полученное удовлетворение от коммуникации и измерялась в номинальной шкале от 1 до 7.

В таблице 1 представлена матрица К.

Таблица 1

Матрица ответов К и оценочная R

K	1	2	3	4	5
1	*		6	2	
2		*			3
3	2		*		
4		1	7	*	
5		4			*

→R:

R	1	2	3	4	5
1	*		1,33	5,33	
2		*			4,33
3	5,33		*		
4		6,33	0,33	*	
5		3,33			*

Для перехода к N -бальной реверсной шкале было получено значение

$$N = [\sum_{1}^{L} (\max K_{ij} /(L+1)) + 0.5] = 6{,}33.$$

$_{j=1,L}$

После формируется нечеткая социометрическая матрица F.

Таблица 2

Нечеткая социометрическая матрица F и G

F	1	2	3	4	5
1	1		0,21	0,84	
2		1			0,7
3	0,84		1		
4		1	0,05	1	
5		0,53			1

G	1	2	3	4	5
1	1		0,34	0	
2		1			0,59
3	0,34		1		
4		0	0	1	
5		0,59			1

Нечетка дендограмма и нечеткая социограмма Un, для уровня n=0.4представлена на рисунке 1.

(3) (1) (5) (2) n=1,00
| | **V**
(3) (1) (5,2) n=0.59
V |
(3,1) (5,2) n=0.34
V
(3, 1, 5, 2) n=0.00

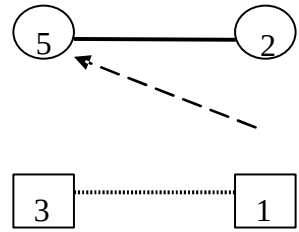


Рис. 1. Нечеткая дендограмма Р и нечеткая социограмма

Таким образом, можно сделать вывод, что для некоторого анализиру - емого типа работы, которую выполняет небольшая группа разработчиков основными подгруппами наличия знаний будут являться коммуникации

между респондентами 5 и 2, и 3 и 1. Первая пара респондентов является более информированной в данном вопросе. Также можно сделать вывод о наличии влияния данных сотрудников на остальных членов группы. При дальнейшем анализе и принятии решений согласно выполнению некоторого проекта, для которого необходима реализация данной работы, полученные характеристики сосредоточения знаний в приведенном коллективе будут являться дополнительной информационной базой.

Выводы. В статье показан подход применения аппарата социальных сетей для идентификации полезных знаний в коллективе разработчиков программного обеспечения. В отличии от понятий, используемых в социологии, для анализа малых социальных групп, в данном примере было предложено идентифицировать полезные знания для выполнения определенного значения через оценку удовлетворенности полученной информации от своих коллег в рамках рабочей группы. При этом степень удовлетворенности отождествлялась со степенью предпочтения того или иного коллеги. Выраженная в некоторой бальной шкале, оценка удовлетворенности использовалась в построении нечеткой социограммы и реализации, предложенного Х. Ямашита подхода для анализа полученных данных.

Библиографические ссылки

1. **Гамаюн И.П.** Проблема выявления полезных знаний при реализации программных проектов. / И.П. Гамаюн, А.В. Иванченко // Вісник Національного технічного університету «Харківський політехнічний інститут». Проблемы информатики и моделирования. – Харьков, – 2009.
2. **Chris Collison, Geoff Parcell.** Learning to Fly: Practical Knowledge Management from Leading and Learning Organizations. — 2000, 2005. — 332 с.
3. **Раскин Л.Г.** Нечеткая математика. Основы теории. / Л.Г. Раскин, О.В. Серая. Парус, - Харьков, 2008.
4. **Yanai I., Yanai A., Tsuda E., Okuda Y., Yamashita H., Inaida J.** Fuzzy Sociogram analysis applying shapely value//Biomedical Fuzzy and Human Sciences, 1996. V.2. .1. P. 63-68.
5. **Yanai I., Yanai A., Tsuda E., Okuda Y., Yamashita H., Inaida J.** Fuzzy Sociogram analysis applying shapely value//Biomedical Fuzzy and Human Sciences, 1996. V.2. .1. P. 63-68.
6. Методы сбора информации в социологических исследованих. Кн 1. М., 1990

Надійшла до редколегії 12.01.10

УДК 534.4: 621.391

©О.Н. Карпов, О.И. Лучинкина
Днепропетровский национальный университет имени Олеся Гончара

ПОСТРОЕНИЕ ОБЩЕЙ СХЕМЫ СИСТЕМЫ РАСПОЗНАВАНИЯ РЕЧИ КАК СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Розглянуто схему побудови системи розпізнавання мови як системи штучного інтелекту. Наведено принцип побудови фрейму опису фонетичної структури мови.

Рассмотрена схема построения системы распознавания речи как системы искусственного интеллекта. Приведен принцип построения фрейма описания фонетической структуры речи.

The scheme of construction of the system of speech recognition as intelligence systems is considered. Principle of construction of frame of description of phonetic structure of speech is cited.

Ключевые слова: искусственный интеллект, распознавание речи, принятие решений.

Постановка задачи. Пусть задана некоторая реализация слитной речи. Необходимо построить некоторую схему системы распознавания речи, позволяющую наиболее эффективно принять решение относительно входных данных.

Во многих языках, в том числе и в русском, существуют достаточно регулярные правила формирования более крупных речевых единиц (слов, фраз) из более мелких (фонем, морфем, слогов). Однако процесс осмысления данных довольно сложен. Так сочетание одних и тех же морфем могут соответствовать разным фразам.

Возникает задача построения некоторой системы искусственного интеллекта, позволяющей принять решение относительно результата распознавания [3].

Фонетическое строение речи . Необходимым условием образования звуков является поток выдыхаемого воздуха. Опираясь на [1 ;6] дадим характеристику некоторым классам фонем русского языка.

Произношение согласных обязательно связано с преодолением препятствия, созданного в ротовой полости на пути воздушной струи. При произношении гласных голосовые связки колеблются, а для воздушной струи обеспечен свободный, беспрепятственный проход через ротовую полость

Струя воздуха, выходящая из трахеи, должна пройти через гортань, в которой находятся голосовые связки. Если связки напряжены и сближены, то выдыхаемый воздух вызовет их колебание, в результате чего возникнет голос, то есть музыкальный звук, тон.

В лингвистике тон – это использование высоты звука для смыслоразличения в рамках слов. Он обязателен при произношении звонких согласных и гласных.

Если голосовые связки расслаблены и раздвинуты, то воздушная струя не вызовет их колебания и тон не возникнет. Такое положение органов речи присуще произношению глухих согласных.

Глухими согласными называются звуки, произносимые без участия тона. К ним относятся [п], [т], [к], [ф], [с], [х] и их мягкие пары.

К звонким согласным относятся [б], [д], [г], [в], [з] и их мягкие пары, а также звуки [ж], [ж':]. При их производстве голосовые связки вибрируют и образуется тон.

Если при производстве согласного звука имеет место преобладание тона над шумом, то мы говорим о сонорных звуках (лат. sonorus звучный), или сонантах. Они наиболее близки к гласным, однако в процессе их образования на пути прохождения воздушной струи появляется преграда. К таким звука относятся [л], [л'], [м], [м'], [н], [н'], [р], [р'], [j].

Все согласные можно разделить на фрикативные, взрывные и аффрикаты.

Взрывные согласные – это смычные согласные, при артикуляции которых нёбная занавеска поднята, и воздух проходит в ротовую полость (в отличие от носовых смычных), а размыкание смычки происходит резко и напоминает взрыв (в отличие от аффрикат).

К взрывным согласным русского языка относятся [п], [б], [т], [д], [к], [г] и их мягкие пары.

Все фрикативные (за исключением в некоторых случаях [w] и [j] й) относятся к шумным согласным и поэтому бывают в двух разновидностях: глухие [ф], [с], [ш], [х] и звонкие [в], [з], [ж], [γ] ([γ] – звонкое [х]); к шумным относятся также взрывные и аффрикаты: а именно – глухие взрывные

[п], [т], [к], [?] ([?] – гортанный взрыв) и аффрикаты [пф], [ц], [ч]; звонкие взрывные [б], [д], [г] и аффрикаты [дз], [дж].

Решение задачи. Распознавание речи – многоуровневый процесс. Основными уровнями являются разбор, анализ и синтез речи. Этих уровни, в свою очередь, разбиваются на подуровни, на каждом из которых решается некоторая конечная подзадача.

Рассмотрим схему описания звуковой системы языка (рис. 1).

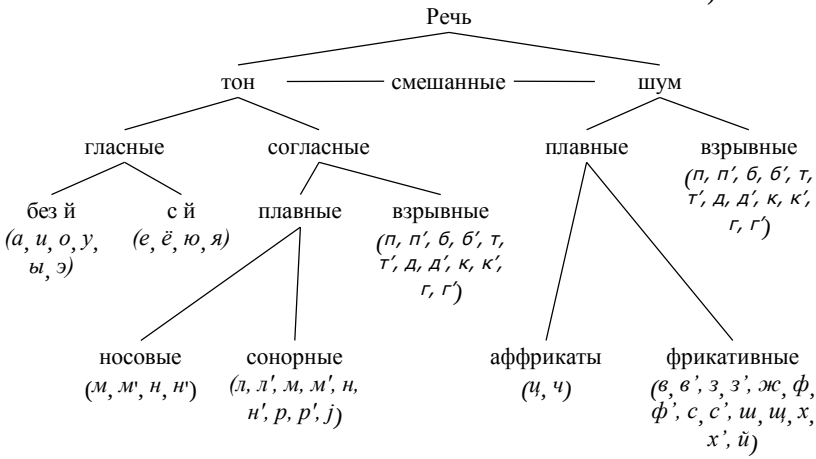


Рис. 1. Граф-схема фрейма описания фонетического строения речи

Представим схему, изображенную на рис.1, на языке предикатов (табл.1).

Таблиця 1

Схема фрейма описания фонетического строения речи на языке предикатов

$s(\Omega, \omega, t, \textit{список характеристик})$ – речевой сигнал
$\textit{тон}[F_0, s(\Omega, \omega, t)]$
$\textit{гласные}[\textit{Тон}, F_1, F_2, ..., F_5]$
$\textit{согласные тон плавные}[\textit{тон}, E_1 \approx E_2]$
$\textit{носовые}[\textit{согласные тон плавные}, NASA]$
$\textit{согласные тон взрывные}[\textit{тон}, \textit{тон смычные}, \textit{всплеск}, \tau_{\min}]$
$\textit{шум}[\rho, s(\Omega, \omega, t)]$
$\textit{шум взрывные}[\textit{шум}, \textit{глухие смычне}, \textit{всплеск}, \tau_{\min}]$

Закінчення таблиці 1

шум плавные $[шум, F_{ШУМ}, \tau \cong 100\ Мсек]$
шум аффрикаты $[шум\ плавные]$
шум фрикативные $[шум\ плавные, щель]$
смешанные $[тон, \rho]$

На рис.1 представлена граф -схема одного из нижних подуровней системы, на котором решается одна из задач, возникающая в процессе распознавания. Подобной схемой можно описать всю систему распознавания речи. В качестве ее узлов могут браться не отдельные объекты, а целые подсистемы. Данная граф -схема в качестве объектов содержит не только некоторую сущность речевого (акустического сигнала), но и отображение этой сущности в лингвистическое представление, т. е. элементы и их совокупности смыслового представления.

Если подниматься по дугам снизу вверх, то в итоге получим речь как совокупность этих элементов. Само движение как вниз (дедукция) и вверх (индукция) – это процессы анализа -синтеза, которые реализуются как соответствие информационного представления, так и программы обработки информации о речевом сигнале.

Представленная граф-схема в качестве списка характеристик содержит формальные правила и характеристики. Правила – это уровень шума, способ произнесения (темп), дикторы, микрофоны т. д. Ограничения и условия – интервал анализа, частота квантования, разрядность, длительность интервала анализа и т. д. Каждая из компонент табл.1 имеет своё информационное или алгоритмическое представление (на самом деле эти понятия зависят от сложности построения системы анализа информации о речевом сигнале или анализа информации , содержащейся в речевом сообщении).

Одной из особенностей фреймовых моделей является наследование свойств верхних уровней. Рассмотрим варианты возможного представления предикатов граф-схемы рис.1.

Речевой сигнал $s(\Omega, \omega, t, список\ характеристик)$.

А. Ввод речевого сигнала. Определение латентных периодов. Нормирование сигнала [2; 5]. Блок анализа. Спектральный анализ.

Данный этап можно разбить на следующие подзадачи:

1. Первичное описание.

Первичное описание может содержать базисные функции:

- Фурье;

- Матье;
- Лежандра;
- Чебышева;
- Уолша;
- Лаггера;
- Эрмита.

2. Вторичное описание. Спектрально_полосное представление.
Обработка спектрально_полосного представления на основе:

- сглаживающих сплайнов;
- фильтров ФНЧ – с ядром $\frac{\sin x}{x}$;
- фильтров с ядром тригонометрического полинома ТП –

$$y_m(t_i) = \sum_{j=1}^{N} y(t_j) \frac{\sin \frac{N}{m}\left(\frac{t_j - t_i}{2}\right)}{\sin\left(\frac{t_j - t_i}{2}\right)},$$

где m – параметр фильтрации высокочастотных компонент, $y(t_j)$ – исходный сигнал, $y_m(t_i)$ – отфильтрованный сигнал с параметром m.

Б. Выбор и формирование типов речевых единиц.
В качестве речевых единиц могу выступать
а) сегменты;
б) слоги;
в) фонемы;
г) слова.
В. Сегментация.

Сегментация может проводиться одним из следующих методов:
а) фильтрация огибающих спектра;
б) применение пороговых методов по групповым признакам;
в) дифференцирование функций параметров;
г) верификация по совокупности параметров.

В итоге $s(\Omega, \omega, t, \textit{список характеристик})$ содержит совокупность процедур, решающих задачи ввода, нормирования, спектрального анализа, спектрально_полосного описания, фильтрации сигналов на низких частотах Ω, сегментации и формирования речевых единиц как соответствующего спектрально_временного описания.

Рассмотрим узлы описанной в табл.1 схемы.

Tîí $[F_0, s(\Omega, \omega, t)]$ – содержит тональный сигнал с частотой основного тона F_0 и обладает свойствами $s(\Omega, \omega, t, \textit{список характеристик})$.

Cîãëàñíûå òîí $[\text{òîí}, E_1 \approx E_2]$ – обладают высоким уровнем энергии на низких частотах до 400 Гц и примерно равной энергией в полосах (400_1000)Гц и (1000_4000) Гц.

Hîñîâûå $[\text{ñîãëàñíûå òîí ïëàâíûå}, NASA]$ – содержат носовую компоненту.

Cîãëàñíûå òîí âçðûâíûå $[\text{òîí}, \text{òîí ñìû÷íûå}, \text{âñïëåñíê}, \tau_{\min}]$ – в качестве паузы содержат низкочастотный гул перед звуком, а сами звуки имеют длительность около $\tau_{\min} = 15$ Мсек.

Øóì $[\rho, s(\Omega, \omega, t)]$ характеризуется высокой частотой ρ перехода через ноль сигнала $s(\Omega, \omega, t, \textit{список характеристик})$ и обладает свойствами $s(\Omega, \omega, t, \textit{список характеристик})$.

Øóì âçðûâíûå $[\text{øóì}, \text{ãëóõèå ñìû÷íå}, \text{âñïëåñíê}, \tau_{\min}]$ – характеризуются глухой паузой звуком, а сами звуки имеют длительность около $\tau_{\min} = 15$ Мсек.

Øóì ïëàâíûå $[\text{øóì}, F_{\text{øóì}}, \tau \cong 100\ Mcek]$ – характеризуются достаточно большой длительностью и параметром ρ.

Øóì àôôðèêàòû $[\text{øóì ïëàâíûå}]$ – характеризуются достаточно большой длительностью и параметром ρ и широким диапазоном частот от 2 Кгц до 7 Кгц для разных звуков (ц_тс, ч_тщ).

Øóì ôðèêàòèâíû å$[\text{øóì ïëàâíûå}, \text{ùåëü}]$ – характеризуются достаточно большой длительностью и параметром ρ и своими частотми для (х, ш, щ, ф, с).

Ñìåøàííûå $[\text{òîí}, \rho]$ – характеризуются достаточно большой длительностью, параметром ρ и тональной компонентою F_0.

Выводы. В каждой подсистеме, входящей в общую систему распознавания речи, решается некоторая отдельная конечная подзадача. Так на рис.1 рассмотрена технология проектирования некоторого устройства, позволяющего решить задачу описания звуковой системы языка.

Всю систему распознавания речи можно представить в виде подобной схемы, в качестве узлов которой могут выступать не только отдельные

объекты, но и целые подсистемы. Реализация подобной схемы требует разработки системы искусственного интеллекта.

Первоочередной задачей при разработке подобной системы распознавания речи является определение существенных для принятия решений признаков и, если необходимо, выделение их поиска в отдельные подзадачи.

Библиографические ссылки

1. **Акишина А. А.** Русская фонетика на фоне общей. / А. А. Акишина, С. А. Барановская. – М., 2007 – 104 с.
2. **Карпов О.М.** Вступ до курсу «Технологія проектування пристроїв розпізнавання промови» /О.М. Карпов // Навч. посіб. – Д., 2000. – 64 с.
3. Компьютерные технологии распознавания речевых сигналов. Монография / О.Н. Карпов, А.Г.Габович, Б.Г.Марченко и др.– К., 2005. – 138 с.
4. Методи та алгоритми оцінки ситуативних відхилень параметрів мови людини. Навч. посібн. / О.М.Карпов, Г.В.Зирнєєва, А.О.Чугай, В.О. Асадулін. // – Д., 2007 – 64 с.
5. **Карпов О.Н.** Технология построения устройств распознавания речи. Монография. / О.Н. Карпов –Д., 2001. – 184 с.
6. **Попова Т.В.** Практическая фонетика русского языка. Учеб. пособие / Т.В. Попова, Т.В. Губанова – Тамбов, 2001.

Надійшла до редколегії 26.11.09

УДК 378:001.891

©**І.С. Катеринчук[*], В.В. Кравчук[**], В.М. Кулик[**], Р.В. Рачок[*]**
[*] Національна академія ДПСУ, [**] Хмельницький ЦНТЕІ

НЕЧІТКЕ ПОРІВНЯННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ У СИСТЕМАХ ТЕСТОВОГО КОНТРОЛЮ ЗНАНЬ

Постановка проблеми. Останнім часом для автоматизації проведення як поточного, так і підсумкового контролю рівня засвоєння знань, використовуються системи тестового контролю. Однак, за своєю функціональністю існуючі системи суттєво обмежують можливості неформалізованої побудови тестових завдань. Зручність програмної реалізації привела до того, що сучасні програмні засоби, які використовуються для тестування, дозволяють будувати питання лише певних типів. Переважно це питання виду «одне питання – декілька варіантів відповідей серед яких одна вірне», «одне питання – декілька варіантів відповідей серед яких декілька вірних», «співставлення варіантів відповідей». Усіх їх об'єднує те, що в них надаються варіанти відповідей, серед яких є вірна або вірні відповіді. У багатьох випадках той хто тестується, не має міцних знань, але має добре розвинену інтуїцію, він може «вгадати» багато вірних відповідей. З іншого боку, наявність декількох близьких варіантів відповідей може розгубити того хто тестується, і він може помилитися. Тобто психологічні особливості особистості, можуть в окремих випадках суттєво вплинути на результат тестування.

Усе це зумовлено тим, що жорстка, шаблонна організація сучасних систем тестування, звичайно, визначається не потребою в об'єктивному оцінюванні знань, а зручністю програмної реалізації.

Звичайно, менш формалізованою є надання текстової відповіді на питання в довільній формі. Деякі тестові системи дозволяють формувати такі питання, однак для перевірки відповідей на них залучається експерт – людина. Необхідність залучення людини є недоліком такого комп'ютерного тестування, що суттєво погіршує його ефективність.

Для автоматизації перевірки відповіді, наданої в текстовому форматі природною людською мовою, необхідно розробити методику порівняння такої відповіді зі зразком (зразками) правильної відповіді закладеної при розробці тестових завдань. Звичайно, традиційне порівняння строк для

вирішення цієї задачі застосувати неможливо. На нашу думку, первинна причина в необхідності нечіткого порівняння строк, полягає у потребі врахувати (компенсувати) нечіткість, яка властива людському мозку і природній мові. Ця нечіткість, зокрема, проявляється в тому, що одне і те ж висловлювання може бути представлене людиною в текстовому вигляді по різному. І всі ці представлення, з точки зору їх інформаційного наповнення, будуть тотожні.

Метою даної роботи є розробка алгоритмів нечіткого порівняння текстової інформації для застосування у системах оцінювання знань.

Робота виконується в рамках Державної програми «Інформаційні та комунікаційні технології в освіті і науці» на 2006 -2010 роки. На відміну від існуючих систем оцінювання з використанням тестів, розроблювана система надаватиме можливість оцінювати письмові відповіді студентів. Письмова відповідь студента представлятиме нечітку лінгвістичну змінну, яка за розробленим алгоритмом буде здійснювати перевірку граматики, та орфографії, на основі чого, буде сформовано образ відповіді та його порівняння з еталонними лінгвістичними змінними бази знань. База знань представлятиме собою розгалужену структуру інформації предметної області.

Слід відмітити, що для повного вирішення даного завдання необхідно забезпечити порівняння текстової відповіді зі зразком за змістом.

Однак такий підхід потребує створення окремих експертних систем для порівняння фраз різної тематики, що є складною задачею. Більш простою альтернативою є використання спрощених методів нечіткого порівняння строк.

Один з можливих підходів, який може бути використаний для нечіткого порівняння строк передбачає визначення метрики і обчислення відстані між строками [1; 2]. Чим більша відстань, тим більшою є відмінність. Оскільки в комп'ютері текстова інформація кодується числами, кожний текстовий рядок представляє собою вектор у N-мірному просторі, де N – кількість символів у рядку.

Функція $d(x,y)$ для обчислення відстані (метрики) між двома векторами x та y має мати наступні властивості:

1. Невід'ємність: $d(x,y) \geq 0 \ \forall x,y$;
2. Властивість нуля: $d(x,y) = 0 \Leftrightarrow x = y$;
3. Симетричність: $d(x,y) = d(y,x) \ \forall x,y$;
4. Нерівність трикутника: $d(x,z) \leq d(x,y) + d(y,z) \ \forall x,y,z$.

Можливо побудувати багато різних метрик, які б відповідали цим властивостям. Наприклад може бути використана Евклідова метрика:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \ . \qquad (1)$$

Однак при обробці текстової інформації така метрика не завжди є зручною. Звичайно, кількість символів, які будуть введені у відповідь на тестове питання не є константою. Тому необхідно мати можливість порівнювати рядки різної довжини і, відповідно, розмірності просторів в яких вони знаходяться. Тому, для нечіткого порівняння текстової інформації доцільно використання метрик, які оцінюють максимальну «вартість» перетворення одного текстового рідка в інший [1; 3].

Однією з найпростіших є відстань (метрика) між рядками за Хеммінгом, яка визначається як число позицій в яких символи не співпадають. Більш складною є метрика Левенштейна, з використанням якої можливим є порівняння рядків різної довжини.

Однак, проведені дослідження застосування метрики Левенштейна показали, що безпосереднє її застосування в задачі нечіткого порівняння відповіді зі зразком у ході тестування є малоефективним. При незначних відхиленнях відповіді від зразка відстань за Левенштейном є невеликою. Проте, якщо в реченні присутній зайвий пропуск, зсув, перестановка слів або інші спотворення, несуттєві з точки зору змісту, інколи отримується значна відстань. У той же час, на коротких текстових рядках (одне слово), даний підхід до порівняння показав задовільні результати. Усе це свідчить про те, що безпосереднє застосування метрики Левенштейна для перевірки і оцінки відповідей у тестових системах не є ефективним.

Тому для нечіткого порівняння текстової інформації у відповідях за ходом тестування, було розроблено алгоритм в якому і зразок і відповідь розбиваються на окремі слова. Після чого проводиться нечіткий пошук співпадінь за словами між зразком і відповіддю, для чого застосовується алгоритм Левенштейна. На основі інформації про відповідність за словами будується оцінка в межах від 0 до 100. Значення 100 відповідає повній збіжності за словами, 0 коли жодного слова з оригіналу немає у відповіді.

Зрозуміло, що порівняння лише за нечітким співпадінням окремих слів не дає об'єктивної оцінки. Тому окрім порівняння за співпадінням слів, проводиться перевірка відповідності структури речення відповіді і зразка. З'ясовується, наскільки порядок слів у відповіді відповідає порядку слів зразка і оцінюється в межах від 0 до 100.

На основі двох оцінок формується інтегральна зважена оцінка, яка нараховується за відповідь. Вагові коефіцієнти були визначені з залученням експертів. Дослідження розробленого алгоритму показали, що він дозволяє оцінити відповідь на питання теста надану в текстовому вигляді. При

незначних відхиленнях у відповіді від зразка виставляється оцінка близька до максимальної. Навіть при незначних спотвореннях речення, коли його зміст не втрачається, оцінка відповіді є високою. З іншої сторони, коли надана відповідь не відповідає зразку виставляється низька оцінка, яка за 100 бальною шкалою прямує до 0.

Слід відмітити, що в окремих випадках, коли питання не пов'язане з використанням у відповіді чіткої термінології, надана текстова відповідь довільної форми за змістом може відповідати зразку, однак використання алгоритму нечіткого порівняння приводить до виставлення низької оцінки.

Для покращення алгоритму потрібно врахувати всі можливі синоніми слів і різні підходи до конструювання речень (можливо, навіть і помилки побудови речень).

Можливим перспективним напрямом подальшого вдосконалення систем тестового контролю, може бути використання методів штучного інтелекту. При використанні природної мови, однакове за змістом висловлювання може бути описане різними способами. Звичайно структура і елементи (окремі слова) текстових представлень може суттєво відрізнятись від зразка. У цьому випадку, використання описаного вище підходу буде некоректним. Тому для порівняння за змістом текстової відповіді зі зразком потрібно виділити цей «зміст» (знання) як з відповіді, так і зі зразка і провести порівняння. Вирішення даної задачі можливе з використанням методів штучного інтелекту.

На жаль, аналіз сучасного програмного забезпечення в цій галузі показує, що програм які на основі використання методів штучного інтелекту були б здатні повною мірою нетривіально опрацьовувати (порівнювати) отримані з тексту елементи «знань» сьогодні не існує навіть для англійської мови [4]. Ця ситуація обумовлена двома причинами [4]. Перша, полягає в незначному розповсюдженні систем лінгвістичного аналізу тексту, здатних інтерпретувати відношення між словами і, відповідно, дійсно видобувати знання як певні елементи з внутрішньою структурою і придатні для змістовної обробки «штучним розумом». Подібні системи лише розпочали з'являтись (Net Owl (www.netowl.com), Attensity (www.attensity.com), RCO Fact Extractor (www.rco.ru)) і їх ще не встигли інтегрувати у прикладних застосуваннях. Друга причина полягає в низькій достовірності автоматично видобуваємих з тексту «знань». Це пов'язано з недосконалістю сучасних алгоритмів інтерпретації тексту і, в окремих випадках, з низькою якістю джерел інформації.

Для «виділення» змісту з текстової інформації, може бути використаний семантичний аналіз. Він дає можливість з довільного тексту на при-

родній мові виділити змістовну структуру (знання). При цьому відбувається виявлення змісту речень, або окремих їх частин. Звичайно, семантичний аналіз є складним завданням. Він важко піддається формалізації. Звичайно, кожна мова має свої особливості, і їх необхідно враховувати при проведенні семантичного аналізу.

Для визначення семантичних відношень між окремими словами, звичайно, використовується тезаурус мови. Природно, що даний тезаурус є специфічним для кожної мови. Він задає набір бінарних відношень на множині слів природної мови. Проблема створення якісного тезарусу є однією з основних при використанні алгоритмічного підходу. Хоча зараз існують комерційні продукти в яких використовуються тезауруси для різних мов, вони насправді включають лише їх підмножини (англійської, російської).

За результатом семантичного аналізу будується семантична мережа – структура для представлення знань у вигляді вузлів пов'язаних дугами (зв'язками). Властивості отриманої семантичної мережі:

- вузли мережі представляють собою поняття, предмети, події, стани;
- дуги семантичних зв'язків створюють відношення між вузлами – поняттями (відношення можуть бути різних типів);
- деякі відношення між вузлами є лінгвістичними, просторовими, часовими, логічними та ін.;
- поняття організовані за рівнями у відповідності до ступеня узагальненості.

Таким чином, у результаті аналізу тексту з нього автоматично видобувається інформація («знання») у вигляді мережі основних понять і зв'язків з ваговими коефіцієнтами. В якості змістовного «портрету» тексту при подальшому порівнянні розглядається не просто список ключових слів, а мережа понять, яка у певному сенсі є «відбитком» змісту тексту. Кожне поняття має певну вагу, яка відображає значимість цього поняття в тексті. Зв'язки між поняттями також мають вагу.

Порівняння семантичних мереж двох текстів дозволить провести їх порівняння за змістом. Незалежно від побудови речень, наявності додаткових суджень, несуттєвих якісних характеристик, які можуть бути присутні у відповіді з неї виділяється основний «зміст» у вигляді семантичної мережі. Аналогічна процедура проводиться зі «зразком» вірної відповіді. Порівняння двох семантичних мереж (тексту відповіді і зразка) дозволить достовірно оцінити ступінь їх тотожності і в результаті виставити об'єктивну оцінку.

Висновок. Таким чином, у роботі запропоновано алгоритм, який може використовуватись для проведення оцінювання текстових відповідей при проведенні комп'ютерного тестування. Для нечіткого порівняння окремих слів у відповіді в алгоритмі використовується метрика Левенштейна, однак для покращення ефективності перевірки додатково проводиться аналіз структури речення. Експериментальна перевірка показала достатню ефективність запропонованого алгоритму при перевірці питань тесту, в яких відповідь потрібно навести у вільному викладенні. Однак для подальшого розвитку пропонується використання семантичного аналізу з подальшим порівнянням семантичних мереж зразка і текстової відповіді.

Бібліографічні посилання

1. **Graham A. Stephen**, String Searching Algorithms // World Scientific Publishing Company, 1994.
2. **Hirata M., Yamada H., Nagai H., Takahashi K**. A versatile data string-search VLSI, // IEEE Journal of Solid-State Circuits, – April 1988. Vol. 23, No. 2, p. 329-35, April 1988.
3. **Жидких А.Д.** Исследование алгоритмов текстового поиска. / А.Д. Жидких, В.И. Костин // Международная студенческая конференция «Информатика и комьютерные технологии», ДонНТУ – 2005.
4. **Ермаков А. Е.** Извлечение знаний из текста и их обработка: состояние и перспективы / А. Е. Ермаков // Информационные технологии – М, 2009. – С. 50 – 55.

Надійшла до редколегії 12.01.10

УДК 519.6

©**О. М. Кісельова*, О. Ю. Лебідь****
**Дніпропетровський національний університет імені Олеся Гончара*
***Академія митної служби України*

МЕТОД РОЗВ'ЯЗАННЯ ЗАДАЧІ НЕЧІТКОГО РОЗБИТТЯ МНОЖИН

Сформульовано нову постановку задачі нечіткого розбиття множин із фіксованими центрами підмножин без наявності обмежень та запропоновано підхід до її розв'язання. Подано метод та алгоритм розв'язання означеної задачі.

Сформулирована новая постановка задачи нечеткого разбиения множеств с фиксированными центрами подмножеств без наличия ограничений и предложен подход к ее решению. Представлены метод и алгоритм решения указанной задачи.

The new statement of set partitioning problem with fixed centers of subsets and without restrictions is formulated, and the approach solving that is proposed. Method and algorithm solving mentioned problem is cited.

Ключові слова: оптимальне розбиття множини, нечітке розбиття, степень належності.

Постановка проблеми. Як зазначено в [1] вимога знаходження однозначного (чіткого) розбиття елементів Ω з E_n може опинитись достатньо грубою та жорсткою при розв'язуванні задач з погано або слабко структурованою вихідною інформацією, тобто задач, в яких невизначеність має нечітку можливісну природу. Послаблення цієї вимоги виконується за рахунок уведення до розгляду нечітких підмножин Ω_i, $i = 1,..., N$, множини Ω та відповідних їм функцій належності, які приймають значення з $[0,1]$.

Одним з варіантів нечіткої задачі оптимального розбиття множин (ОРМ) стає задача знаходження такого нечіткого розбиття множини Ω на його нечіткі підмножини, які в деякому сенсі «мінімізують» деякий цільовий функціонал. Така задача буде зведена до задачі знаходження степені належності елементів множини Ω шуканим нечітким підмножинам $\Omega_1, ..., \Omega_N$, які за сукупністю визначають нечітке розбиття множини Ω.

Аналіз досліджень та публікацій. В [1] наведено постановку задачі нечіткого розбиття множин з фіксованими центрами підмножин без наявності обмежень. У даній роботі запропонуємо інший підхід до розв'язування даної задачі.

Постановка задачі. Для формулювання постановки задачі необхідно ввести деякі означення. Сформулюємо означення нечіткого розбиття множини.

Нечітким розбиттям чіткої множини Ω з E_n, де Ω – обмежена, вимірна за Лебегом, опукла множина, будемо називати систему нечітких підмножин $P = P(\Omega, N) = \{\Omega_i : \Omega_i \subseteq \Omega,\ \forall i = 1,..., N\}$, де $\Omega = (\Omega, \mu_\Omega(x))$, для яких виконується умова

$$\sum_{k=1}^{N} \mu_{\Omega_k}(x) = \mu_\Omega(x) \geq 1,\ \forall x \in \Omega. \qquad (1)$$

Остання умова вводиться з наступних суджень. Якщо вимагати виконання умови $\sum_{k=1}^{N} \mu_{\Omega_k}(x) = \mu_\Omega(x) = 1,\ \forall x \in \Omega$, то перетин нечітких множин буде в тих точках $x \in \Omega$, значення функцій належності в яких дорівнюватиме 0,5, або нижче. У теорії нечітких множин точки переходу небажані. Якщо ж матимемо перетин функцій належності на рівні, нижче ніж 0,5, то ми штучно збільшуємо кількість точок множини Ω, які з малим ступенем належать певній множині Ω_i, $i = 1,..., N$.

Клас усіх можливих нечітких розбиттів множини Ω на N нечітких підмножин позначимо через $\Re = \Re(\Omega, N)$.

На множині можливих нечітких розбиттів $\Re$ вводиться *цільовий функціонал* $(F : \Re \to R^1)$ у вигляді

$$F(P) = \sum_{i=1}^{N} \int_{\Omega_i} (c(x, \tau_i) + a_i)\rho(x)dx, \qquad (2)$$

де функції $c(x, \tau_i)$ – задані, дійсні, обмежені, визначені на Ω, вимірні за x за будь-якого фіксованого $\tau_i = (\tau_i^{(1)},..., \tau_i^{(n)})$ з Ω для усіх $i = 1,..., N$; $P \in \Re$; $\rho(x)$ – задана, обмежена, вимірна на Ω функція; $a_1,..., a_N$ – задані дійсні невід'ємні числа.

Наведемо постановку задачі нечіткого розбиття множинбез обмежень з фіксованими центрами.

Задача 1. Знайти

$$F(P) \to \min_{P \in \Re},$$

де цільовий функціонал $F(P)$ має вигляд (2).

У задачі 1 маємо нечіткі інтеграли (інтеграли за нечіткою множиною), які визначаються таким чином [2]:

$$\int\limits_{\Omega_i} \rho(x)dx = \int\limits_{\Omega_i} \rho(x)\mu_{\Omega_i}(x)dx,\ i=1,...,N,$$

де у правій частині маємо інтеграл Лебега за звичайною множиною.

Для того, щоб мати можливість ідентифікувати нечітке розбиття P множини Ω, необхідно знати для кожної точки x з Ω ступінь її належності до кожної з множин Ω_i, $i=1,...,N$. Тобто, необхідно знайти вектор-функцію належності вигляду:

$$\mu(x)=(\mu_{\Omega_1}(x),...,\mu_{\Omega_N}(x)),\ x\in\Omega.$$

Надалі, для спрощення функцію належності множини Ω_i, $i=1,...,N$ — $\mu_{\Omega_i}(\cdot)$ будемо позначати через $\mu_i(\cdot)$, $i=1,...,N$.

Функціонал (2) набуває наступного вигляду

$$I(\mu(\cdot))=\int\limits_{\Omega}\sum_{i=1}^{N}(c(x,\tau_i)+a_i)\rho(x)\mu_i(x)dx. \qquad (3)$$

Переформулюємо задачу 1 у термінах функцій належності $\mu_i(\cdot)$, $i=1,...,N$.

Задача 2. Знайти

$$I(\mu(\cdot))\to\min_{\mu(\cdot)\in M},$$

за умов

$$\sup_{x\in\Omega}\mu_{\Omega_i\cap\Omega_j}(x)<1,\ i\neq j,\ i,j=1,...,N, \qquad (4)$$

де

$$M=\left\{\mu(x)=(\mu_1(x),...,\mu_N(x)):0\le\mu_i(x)\le 1,\ i=1,..,N,\sum_{k=1}^{N}\mu_k(x)\ge 1,\ \forall x\in\Omega\right\}.$$

Функціонал $I(\mu(\cdot))$ лінійний та неперервний відносно $\mu(\cdot)$ на M.

Задача 2 має розв'язок, оскільки множина M замкнена, обмежена та опукла і належить до множини гільбертового простору $L_2(\Omega)$, функціонал $I(\mu(\cdot))$ неперервний та опуклий на M. Таким чином, з огляду на узагальнену теорему Вейєрштрасса задача 2 має глобальний розв'язок.

В якості функцій належності $\mu_i(\cdot)$, $i=1,...,N$ оберемо функцію аналітичного вигляду, наприклад, функцію Гауса

$$\mu_i(x) = e^{-\frac{\|x-\tau_i\|^{2b_i}}{2\sigma_i^2}}, \quad i = 1,...,N, \tag{5}$$

де b_i – параметр, який відповідає за ядро нечіткої множини Ω_i, $i = 1,...,N$; σ_i – параметр, який відповідає за ширину функції $\mu_i(\cdot)$, тобто за границю нечіткої множини Ω_i, $i = 1,...,N$; τ_i – центр нечіткої множини Ω_i, $i = 1,...,N$; $x \in \Omega$.

На рис. 1 наведено вигляд двох функцій Гауса для випадку евклідової метрики та наступних значень параметрів: $x \in E^1$, $b = (b_1, b_2) = (1,1)$, $\sigma = (\sigma_1, \sigma_2) = (1,1)$, $\tau = (\tau_1, \tau_2) = (0.25, 0.75)$.

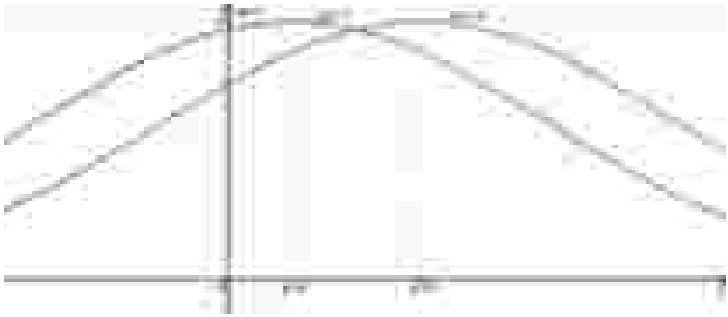


Рис. 1. Вигляд функцій належності для одновимірного випадку

Вдамося до аналізу функції належності вигляду (5). Значення b_i, $i = 1,...,N$ повинні бути цілими та $b_i \geq 1$, $i = 1,...,N$. При таких значеннях b_i відповідна функція належності має «дзвіноподібну» форму, тобто буде вгнутою функцією на Ω. Чим більше значення b_i, $i = 1,...,N$, тим більше ядро відповідної нечіткої множини.

З аналітичного виразу функцій Гауса видно, що для таких функцій завжди буде виконуватись умова $0 \leq \mu_i(x) \leq 1$, $i = 1,..,N$ для усіх $x \in \Omega$.

Залишилося питання виконання умови (4). Сформулюємо відповідь наступним чином. Для того, щоб для функцій належності виконувалась умова (4), необхідно, щоб перетин ядер нечітких множин був порожнюю множиною.

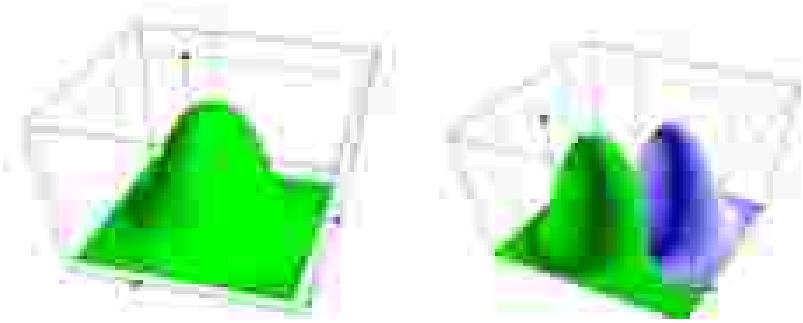
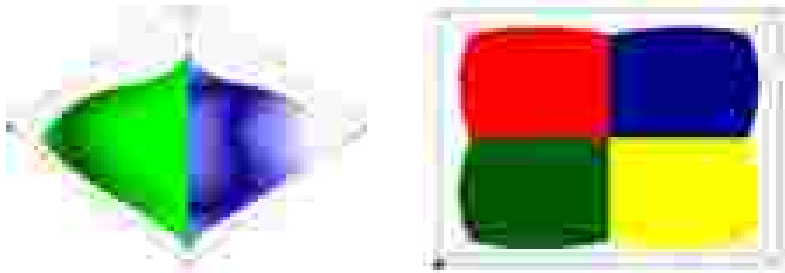


Рис. 2. Вигляд функцій належності Гауса вдвовимірному випадку

Якщо ввести поняття рівня значущості, то зрозуміло, що при рівні значущості близькому до одиниці нечітке розбиття приймає вигляд точок (рис. 5, б).



а) б)

Рис. 3. Нечітке розбиття одиничного квадратапри нульовому рівні значущості (вигляд зверху): а) на 2 множини; б) на 4 множини

Задачу 2 можна переформулювати наступним чином. Знайти такі значення параметрів b_i таσ_i, $i=1,...,N$, щоб функціонал (3) приймав мінімальне значення при виконанні умови (4). Фактично нам потрібно, щоб при рівні значущості, близькому до одиниці нечітке розбиття прямувало до вигляду чіткого оптимального розбиття множин (рис. 3, б).

На рис. 4 та 5 вказано нечітке розбиття одиничного квадрата на чотири множини при різних рівнях значущості.

а) б)

Рис. 4. Нечітке розбиття одиничного квадрата на 4 множини (вигляд зверху) а) рівень значущості 0,7; б) рівень значущості 0,8

а) б)

Рис. 5. Нечітке розбиття одиничного квадрата на 4 множини (вигляд зверху) а) рівень значущості 0,9; б) рівень значущості 0,99

Наведемо алгоритм розв'язання задачі 2. Для цього область Ω вкладаємо у n-вимірний паралелепіпед П, стороні якого паралельні осям декартової системи координат, зазначаючи, що $\rho(x)=0$ для усіх $x\in\Pi\setminus\Omega$. Паралелепіпед П вкриваємо прямокутною сіткою.

1. Фіксуємо $b=(b_1,...,b_N)$, причому $b_i=1$, $i=1,...,N$, $l=1$.
2. $t=1$, $\sigma_i^{(t)}=0{,}01$, $i=1,...,N$.
3. При фіксованих значеннях b_i, $i=1,...,N$ та $\sigma_i^{(t)}$, $i=1,...,N$ розв'язуємо задачу знаходження вектор-параметрів $\sigma=(\sigma_1,...,\sigma_N)$ таких, щоб функціонал (3) набував мінімального значення. Для цього використовуємо модифікований r-алгоритм Шора в Н-формі, а саме

$$\sigma^{k+1}=\sigma^k+h_k\frac{H_{k+1}g_G(\sigma_k)}{\sqrt{(H_{k+1}g_G(\sigma_k),g_G(\sigma_k))}},\ k=0,1,2,... \quad (6)$$

де

$$H_{k+1}=H_k+(1/\alpha_k^2-1)\frac{H_k\Delta_k\Delta_k^TH_k}{(H_k\Delta_k,\Delta_k)},\ \Delta_k=g_G(\sigma_k)-g_G(\sigma_{k-1}),$$

$$g_G(\sigma)=(g_G^{\sigma_1}(\sigma),...,g_G^{\sigma_N}(\sigma)),$$

$$g_G^{\sigma_i}(\sigma) = \int_{\Omega} \rho(x) g_\mu^{\sigma_i}(\sigma, x) c(x, \tau_i) dx, \quad i = 1, 2, ..., N,$$

$$g_\mu^{\sigma_i}(\sigma, x) = \begin{pmatrix} g_\mu^{\sigma_i^{(1)}}(\sigma, x) \\ ..... \\ g_\mu^{\sigma_i^{(n)}}(\sigma, x) \end{pmatrix},$$

$$g_\mu^{\sigma_i}(\sigma, x) = \left(g_\mu^{\sigma_i}(\sigma, x) \right) = \int_{\Omega_i} \rho(x) c(x, \tau_i) \mu_i(x) \frac{\|x - \tau_i\|^{2b_i}}{\sigma_i^3} dx, \quad i = 1, 2, ..., N. \quad (7)$$

4. Інтеграли, що входять до формули (7), обчислюються за допомогою квадратурної формули трапеції. Коефіцієнт розтягнення простору α_k приймається рівним 3. Для крокового множника h_k застосовується адаптивний засіб регулювання.

5. Якщо $\left\| \sigma^{(t+1)} - \sigma^{(t)} \right\| \le \varepsilon_1$, де ε_1 – задана точність, то перехід на наступний пункт. Інакше $t = t + 1$ та перехід на п. 3.

6. Для знайдених значень $\sigma = (\sigma_1, ..., \sigma_N)$ перевіряємо умову перетину ядер нечітких множин. Якщо отримана множина порожня, то $b_i = b_i + 1$, $i = 1, ..., N$, і перехід на п. 6. Інакше для тих нечітких множин, в яких є спільні точки ядра, зменшуємо на одиницю значення $b_i = 1$ і переходимо на п. 2.

7. Якщо $\left| I_l^* - I_{l-1}^* \right| > \varepsilon_2$, де ε_2 – задана точність, то $l = l + 1$ і перехід на п. 2. Інакше за розв'язок обираємо те розбиття, для якого досягнуто I_l^*. Алгоритм описано.

Висновки. У роботі наведено постановку задачі нечіткого розбиття без обмежень з фіксованими центрами підмножин. Для поставленої задачі наведено підхід та алгоритм розв'язання. Надалі планується узагальнити алгоритм на випадок розв'язання задачі нечіткого розбиття з обмеженнями.

Бібліографічні посилання

1. **Киселёва Е. М.** Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е. М. Киселёва, Н. З. Шор. – К., 2005. – 564 с.
2. **Бочарников В. П.** Fuzzy- технология : Математические основы. Практика моделирования в экономике. – СПб., РАИ, 2001. – 328 с.

*Надійшла до редколегії 30.01.*10

УДК 519.8

О. М. ©Кісельова[*], М. С. Сазонова[**]
[*]Дніпропетровський національний університет імені Олеся Гончара,
[**]Національна металургійна академія України

ПРОГРАМНИЙ КОМПЛЕКС NZORM ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ІЗ РІЗНИХ КЛАСІВ СІМЕЙСТВА НЕПЕРЕРВНИХ НЕЛІНІЙНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН

Розглядаються неперервні нелінійні задачі оптимального розбиття множин (ОРМ) із фіксованими центрами підмножин, а також із відшуканням координат центрів підмножин. Пропонується і описується система NZORM для програмної реалізації алгоритмів розв'язання цих задач у постановках без обмежень та при наявності обмежень.

Рассматриваются непрерывные нелинейные задачи оптимального розбиения множеств (ОРМ) с фиксированными центрами подмножеств, а так же с отысканием координат центров подмножеств. Предлагается и описывается система NZORM для програмной реализации алгоритмов решения этих задач в постановках без ограничений и при наличии ограничений.

Some continuous nonlinear problems of optimal set partition both with fixed subset centers and subset centers arrangement are examined. NZORM system, being an implementation of the algorithms for solving the posed problems in mathematical statements both without and under constraints, is proposed and described.

Ключові слова: нескінченновимірне математичне програмування, оптимальне розбиття множин, нелінійні задачі.

Вступ. Дослідження нових розділів загальної теорії нескінченновимірного математичного програмування, до якої відносяться неперервні задачі ОРМ, є передумовою створення ефективних методів та чисельних алгоритмів розв'язання важливих задач оптимізації, які є актуальними при моделюванні різних економічних, виробничих, соціальних та інших процесів. У даній статті авторами продовжуються дослідження [11)1] неперервних нелінійних задач ОРМ, а саме: пропонується і описується програмний комплекс NZORM, який є унікальним, бо дозволяє програмно розв'язувати нелінійні задачі ОРМ у різних постановках: як із фіксованими центрами підмножин, так і з розташуванням центрів підмножин, невідомих заздале-

гідь. До цього були програмно реалізовані лише алгоритми розв'язання лінійних задач ОРМ [2] та алгоритми розв'язання нелінійних задач із фіксованими центрами підмножин [4].

Постановка задачі. Система NZORM розроблена для розв'язання неперервних нелінійних однопродуктових задач оптимального розбиття множин (ОРМ) у чіткій постановці.

Функціональність програми забезпечує розв'язання задач з 4 класів сімейства неперервних нелінійних задач ОРМ [1]:

1) задача А2 – неперервна нелінійна задача оптимального розбиття множини $\Omega \in E_2$ на її неперетинні підмножини $\Omega_1, \ldots, \Omega_N$ (серед яких можуть бути й порожні) без обмежень із заданим положенням центрів $\tau_1, \ldots, \tau_N$ цих підмножин відповідно;

2) неперервна нелінійна задача оптимального розбиття множини $\Omega \in E_2$ на її неперетинні підмножини $\Omega_1, \ldots, \Omega_N$ (серед яких можуть бути й порожні) при обмеженнях у формі рівностей і нерівностей з відшуканням координат центрів $\tau_1, \ldots, \tau_N$ цих підмножин;

3) задача А5 – неперервна нелінійна задача оптимального розбиття множини $\Omega \in E_2$ на її підмножини $\Omega_1, \ldots, \Omega_N$ (серед яких можуть бути порожні) при обмеженнях у формі рівностей і нерівностей із заданим положенням центрів $\tau_1, \ldots, \tau_N$ цих підмножин відповідно;

4) задача А6 – неперервна нелінійна задача оптимального розбиття множини $\Omega \in E_2$ на її підмножини $\Omega_1, \ldots, \Omega_N$ (серед яких можуть бути порожні) при обмеженнях у формі рівностей і нерівностей з відшуканням координат центрів $\tau_1, \ldots, \tau_N$ цих підмножин відповідно.

Надається можливість розв'язання задач ОРМ, у цільовий функціонал [1] яких входять:

– опуклі функції $\varphi_{ii}(Y)$, $i = 1, \ldots,$;

– увігнуті функції $\varphi_{ii}(Y)$, $i = 1, \ldots,$.

Метою роботи є розробка й створення програмного комплексу NZORM для реалізації відповідних алгоритмів [1] розв'язування всіх цих задач на ЕОМ, отримання чисельних розв'язків та графічних візуалізацій результатів розв'язання названих задач.

Для реалізації розв'язання кожної з перелічених вище задач АХ (А2, А4, А5, А6) використуються окремі функції Task_AX(), а саме: Task_A2(), Tast_A4(), Task_A5(), Task_A6() відповідно. Ці функції розташовані в однойменних *.h файлах. У випадку розв'язання задачі АХ з опуклими фун-

кціями $\varphi_{ii}(Y)$, $i\overline{1,...,}$, функція типу Task_AX() визиває функцію Convex:: Fun_AX() (реалізовану в файлі fun.h), а у випадку задачі АХ з увігнутими функціями $\varphi_{ii}(Y)$, $i\overline{1,...,}$, – функцію Unconvex:: Fun_AX()(реалізовану в файлі fun1.h) відповідно.

При виборі засобів реалізації програмного продукту для розв'язання неперервних нелінійних задач оптимізації потрібно було врахувати наступні *фактори*:

– система NZORM повинна стати потужним засобом для розв'язання всіх задач із класу неперервних нелінійних однопродуктових задач оптимізації як універсальна система, що орієнтована на широке коло кваліфікованих спеціалістів, які досліджують та розв'язують нелінійні задачі ОРМ або ті, що зводяться до них;

– система NZORM повинна використовувати існуючі бібліотеки процедур, що реалізують розв'язання задач оптимізації за допомогою r-алгоритму Н.З Шора [5];

– система NZORM повинна надавати зручний інтерфейс користувача та наочну візуалізацію розв'язання задач оптимізації.

Метод розв'язування. *Реалізація.* Система NZORM розроблена у програмному середовищі Borland Builder 6 мовою C++. Система функціонує на IBM-сумісних комп'ютерах під керуванням Windows 98 (NT) і вище. NZORM є інструментом, який у доступній формі забезпечує чисельний розв'язок задач ОРМ, а також його графічну візуалізацію.

Структурно система NZORM складається з інтерфейсної і обчислювальної частин.

Функціональність системи NZORM. Інтерфейсна (пункт 1) і обчислювальна (пункти 2,3) частини системи NZORM дозволяють реалізовати наступні функції:

1) функції інтерфейсу користувача (введення даних, збереження даних задачі у файлі, відкриття задачі, що вже існує, зі збереженого файлу, запуск задачі на розв'язання, функції, що реалізують графічну візуалізацію розв'язку (відображення рисунку результату у вікні, збереження до файла графічного формату));

2) функції, що безпосередньо реалізують математичні алгоритми розв'язання перелічених вище задач (функції сімейства Task_AX, а також допоміжні Fun_AX , що використовують Task_AX відповідно);

3) функція ralgb4, що містить вже існуючу бібліотеку процедур, що реалізує розв'язання задачі оптимізації за допомогою r-алгоритму Н.З. Шора [5] ralgb4.

Усі дані задачі, з якими працює, заповнює, та обробляє система NZORM при виклику та роботі перелічених функцій, є змінними полями задачі та оголошені у файлі taskDataDefinition.h.

Розглянемо далі кожний з перелічених вище класів функцій докладно.

1) Функції інтерфейсу користувача містять у собі оброблювачі подій. У користувача є можливості наведені на рис. 1:

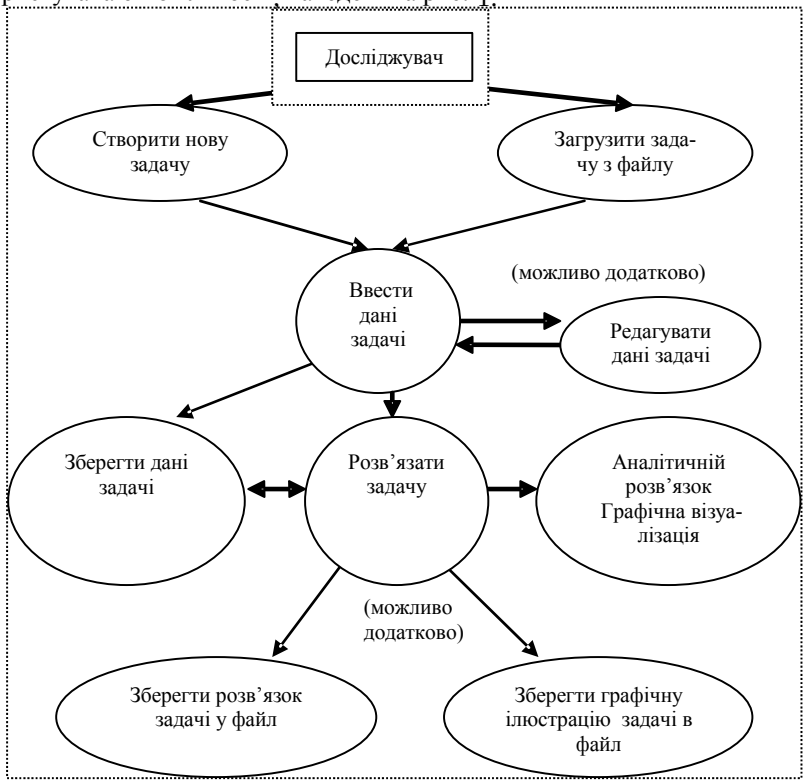


Рис. 1. Базовий варіант використання системи NZORM

Відзначимо, що при введенні даних задачі заповнюються значення TaskParameters[0], TaskParameters[2] масиву byte TaskParameters[4] – параметри задачі, що використовуються для визначення того, до якого з перерахованих вище типів задач (A2, A4, A5, A6) відноситься подана:

- TaskParameters[0]=1 – задача без обмежень;
- TaskParameters[0]=0 – задача з обмеженнями;
- TaskParameters[2]=1 – задача без розташування центрів;

– TaskParameters[2]=0 – задача з розташуванням центрів.

У тілі функції введення даних є розгалуження за наступною умовою:

– якщо задача з обмеженнями (if (TaskParameters[0] == 0)), то здійснюється перехід на форму TabSheet6, яка використується для введення обмежень;

– інакше, якщо задача без обмежень, перехід здійснюється на наступну форму (форма TabSheet6 пропускається).

2) Функції, що безпосередньо реалізують математичні алгоритми розв'язання перелічених вище задач A2, A4, A5, A6:

а) *Функції сімейства Task_AX* . Відповідно зі значеннями масиву TaskParameters, який заповняється в результаті роботи інтерфейсної частини системи NZORM функціями користувача, визивається одна з функцій сімейства Task_AX:

– Task_A2 – у випадку розв'язання задачі A2;

– Task_A4 – у випадку розв'язання задачі A4;

– Task_A5 – у випадку розв'язання задачі A5;

– Task_A6 – у випадку розв'язання задачі A6 відповідно.

Зупинимося на роботі цих функцій більш детально.

Алгоритм роботи будь_якої з функцій сімейства Task_AX:

1) Виділення пам'яті та початкове заповнення (обнуління) вектора узагальненого псевдоградієнту g ; таблиці розбиттів ResSubsets ; вектора psixy величин, за якими у подальшому буде відбуватися оптимізація (функцією ralgb4, що реалізує r-алгоритм Шора); результуючих (у подальшому) величин ResPsi (тобто Ψ_i), ResY (тобто Y_i) та ResCenCoord (двовимірний масив результуючих центрів $\tau_1, ..., \tau_N$) (розмірності всіх векторів згідно алгоритмів [1]);

2) відкриття для перезапису файлу iX.txt (X – одна з цифр «2», «4», «5», «6» залежно від того, яка з функцій Task_AX працює відповідно);

3) викликається функція, що реалізує обчислення двоїстого функціонала на нульовій ітерації, тобто одна з функцій:

– ResDFunctional_0=Convex::Fun_AX(NumCenters, g, psixy) – у випадку розв'язання задач ОРМ, до цільового функціонала яких входять опуклі функції $\varphi_{ii}(Y)$, $ii = 1,...,$, [3];

– ResDFunctional_0=Unconvex::Fun_AX(NumCenters, g, psixy) – у випадку розв'язання задач ОРМ, до цільового функціонала яких входять увігнуті функції $\varphi_{ii}(Y)$, $ii = 1,...,$, [3];

4) результати нульової ітерації (значення двоїстого функціонала, градієнта та (згаданих в пункті 1 алгоритму) результуючих величин) видаються у файл iX.txt;

5) викликається функція ralgb4, яка реалізує розв'язання задачі оптимізації за допомогою r-алгоритму Шора [5], на описанні якої зупинимося пізніше, одним з вихідних параметрів в яку подається «Convex::Fun_AX» – у випадку розв'язання задачі ОРМ, до цільового функціонала якої входять опуклі функції $\varphi_{ii}(Y)$, $i = 1,...,$, або «Unconvex::Fun_AX» – у випадку розв'язання задачі ОРМ, до цільового функціонала якої входять опуклі функції $\varphi_{ii}(Y)$, $i = 1,...,$, на описанні функцій Fun_AX зупинимося пізніше;

6) результати останньої ітерації (значення двоїстого функціонала, градієнта та (згаданих у пункті 1 алгоритму) результуючих величин), а також значення цільового (прямого) функціонала задачі, який обчислюється за допомогою функції Functional(), видаються у файл iX.txt.

б) Функції сімейства Fun_AX.

Як було відмічено у пункті 1, функція сімейства Task_AX (X – одна з цифр «2», «4», «5», «6» в залежності від того, яка з функцій Task_AX працює відповідно) викликає відповідну функцію сімейства Fun_AX. Розроблено два два варіанта виконання функцій Fun_AX:

1) для випадку розв'язання нелінійних задач ОРМ, у цільовий функціонал яких входять опуклі функції $\varphi_{ii}(Y)$, $i = 1,...,$;

2) для випадку розв'язання нелінійних задач ОРМ, у цільовий функціонал яких входять увігнуті функції $\varphi_{ii}(Y)$, $i = 1,...,$.

У першому випадку всі функції Fun_AX реалізовані і фізично розташовані у файлі fun.h, і доступ до відповідної функції сімейства здійснюється (за допомогою використання простору імен) наступним чином: Convex::Fun_AX().

У другому випадку всі функції Fun_AX реалізовані і фізично розташовані у файлі fun1.h, і доступ до відповідної функції сімейства здійснюється (за допомогою використання простору імен) наступним чином: Unconvex::Fun_AX().

Функції сімейства Fun_AX дозволяють по поточних значеннях величин ResPsi (тобто Ψ_i), ResY (тобто Y_i) та ResCenCoord (двовимірний масив результуючих центрів $\tau_1, ..., \tau_N$) обчислити значення вектора узагальненого псевдоградієнта g та повертають значення двоїстого функціонала задачі [3].

3) Функція ralgb4 містить вже існуючу бібліотеку процедур, що реалізує розв'язання задачі оптимізації за допомогою r-алгоритму Шора [5].

Під час роботи функція видає проміжні результати в текстовий файл, ім'я якого визначається під час роботи інтерфейсної частини системи NZORM, на кожній (введеній користувачем) за рахунком ітерації.

Керівництво користувачу

NZORM складається з одного виконавчого файла NZORM.exe, який не містить додаткових бібліотек. Для запуска програми достатньо лише скопіювати її на жорсткий диск і запустити. Вимоги до системи – Windows 9x/ME/XP. Під час роботи системи створюється додаткова директорія, яка містить інформацію про розв'язані вже задачі, що зберігається у файлах користувача. Не рекомендується власноруч змінювати вміст цієї директорії, оскільки це може привести до невірної роботи системи.

Під час роботи системи при розв'язанні одної (обраної дослідником - користувачем) з задач A2, A4, A5, A6, тобто при роботі одної з функцій, що безпосередньо реалізують математичні алгоритми розв'язання перелічених вище задач (функції Task_A2, Task_A4, Task_A5, Task_A6 відповідно, а також допоміжні Fun_AX , що використовуються Task_AX, відповідно) створюється один з файлів i2. txt, i4.txt, i5.txt, i6.txt (відповідно), що міститиме вихідну інформацію про данні задачі, що розв'язується, а після її розв'язання містиме ще й отримані чисельні результати.

Також, під час роботи програми розв'язуючий модуль створює текстовий файл *. txt, ім'я якого задається дослідником _користувачем програмного продукту, що міститиме інформацію, щодо самого процесу розв'язання задачі оптимізації r-алгоритмом Н.З. Шора (функція ralgb4). Ця інформація може видаватися у файл на кожній ітерації, або на кожній n-ій ітерації – залежно від бажання дослідника.

Система є гнучкою у плані завдання вихідних даних. Надає можливості табличного й аналітичного завдання конфігурації вихідної області, вагової функції, містить бібліотеку метрик, а також дає можливість задати початкові наближення для ітераційного алгоритму.

Програмний комплекс має зручний інтерфейс користувача, який з одного боку дозволяє звести до мінімуму зусилля при введенні вихідних даних, подальшій роботі з отриманими результатами, їх візуалізації та збереженні (вхідних даних задачі, що розв'язується, її чисельного розв'язку та його графічної інтерпретації), а з іншого – дозволяє оптимізувати час виконання розрахунку за допомогою визначення конкретного типу задачі, що розв'язується, й виконання відповідної підпрограми.

Висновки. Розроблено, протестовано й налагоджено програмний комплекс NZORM, який дозволяє розв'язувати нелінійні задачі ОРМ у різних постановках, і може бути застосований при розв'язуванні нескінченнови-

мірних задач розбиття множини споживачів деякої однорідної продукції, розподілених у цій області із заданою щільністю, на сфери обслуговування підприємствами, що виготовляють однорідну продукцію, з метою мінімізації нелінійного функціонала сумарних витрат на виробництво та доставку продукції споживачу при обмеженнях на потужності підприємств у формі рівностей та нерівностей чи без обмеженень на потужності підприємств, і коли залежності вартостей виробництва продукції підприємств від їх потужностей можуть бути описані у вигляді опуклих чи ввігнутих функцій; а також нескінченновимірних задач розміщення підприємств із одночасним розбиттям даного регіону, неперервно заповненого споживачами, на області споживачів, кожна з яких обслуговується одним підприємством, із метою мінімізації транспортних і виробничих витрат. У ролі споживачів тут можуть виступати також телефонні, радіо -, телеабоненти, школяри, виборці, точки зрошуваної території, пацієнти для діагностики захворювань і т. д. За допомогою системи NZORM одержано чисельні розв'язки та графічну візуалізацію результатів таких модельних задач . Система NZORM є унікальною, бо дозволяє програмно розв'язувати нелінійні задачі ОРМ у різних постановках, причому алгоритми розв'язання нелінійних задач ОРМ із розташуванням центрів підмножин реалізовано вперше.

Бібліографічні посилання

1. **Дунайчук М.С.** Методи та алгоритми розв'язання неперервних нелінійних задач оптимального розбиття множин: автореф. дис. на здобуття наукового ступ. канд. фіз. -мат. наук : спец. 01.05.01 «Теоретичні основи інформатики та кібернетики» / М.С. Дунайчук. – Д., 2008.
2. **Киселёва Е.М.** Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М. Киселёва, Н.З. Шор . – К., 2005. 564 с.
3. **Кісельова О.М.** Про отримання векторів узагальнених псевдоградієнтів двоїстих функціоналів неперервних нелінійних задач оптимального розбиття із розташуванням центрів підмножин / О.М. Кісельова, М.С. Дунайчук // Питання прикладної математики і математичного моделювання. – Д., – 2009. – С. 142-149.
4. **Кісельова О.М.** Система NZORM для розв'язання неперервної нелінійної задачі оптимального розбиття множин / О.М. Кісельова, М.С. Дунайчук // Питання прикладної математики і математичного моделювання. – Д., – 2006. – С. 49–61.
5. **Шор Н.З.** Методы минимизации недифференцируемых функций и их приложение. / Н.З. Шор – К., 1979. – 200 с.

Питання прикладної математики і математичного моделювання. Д., 2010

Надійшла до редколегії 04.03.10

УДК 519.85

О.М. ©Кісельова, В.О. Строєва
Дніпропетровський національний університет імені Олеся Гончара

РОЗВ'ЯЗАННЯ НЕЛІНІЙНОЇ НЕПЕРЕРВНОЇ БАГАТОПРОДУКТОВОЇ ЗАДАЧІ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН З РОЗТАШУВАННЯМ ЦЕНТРІВ ПІДМНОЖИН

Розглядається неперервна нелінійна багатопродуктова задача оптимального розбиття множини Ω з n-вимірного евклідового простору E_n на її неперетинні підмножини з розташуванням центрів при обмеженнях у формі рівностей та нерівностей з опуклим цільовим функціоналом. Здійснено перехід від названої нескінченновимірної задачі оптимізації до задачі пошуку сідлової точки функціонала Лагранжа.

Рассматривается непрерывная нелинейная многопродуктовая задача оптимального разбиения множества Ω из n-мерного евклидового пространства E_n на его непересекающиеся подмножества с размещением центров при ограничениях в форме равенств и неравенств с выпуклым целевым функционалом. Выполнен переход от названной бесконечномерной задачи оптимизации к задаче поиска седловой точки функционала Лагранжа.

The continuous nonlinear multigrocery problem of optimum splitting of set Ω from n-dimensional Euclidean spaces E_n on its not crossed subsets with placing of the centres is considered at restrictions in the form of equalities and inequalities with convex target functional. Transition from named infinity-dimensional optimisation problems to a search problem saddle points functional Lagrangian is executed.

Ключові слова: нескінченновимірне математичне програмування, оптимальне розбиття множин, багатопродуктова задача, негладка оптимізація.

Вступ. Відомо, що до задач неперервного оптимального розбиття множин (ОРМ) зводиться широкий клас теоретичних та практичних задач оптимізації. Наприклад, це нескінченновимірні транспортні задачі, а також більш загальні – нескінченновимірні задачі розташування підприємств з одночасним розбиттям даного регіону, неперервно заповненого

споживачами, на області споживачів. У ролі споживачів можуть бути телефонні абоненти, виборці, школярі, пацієнти для

діагностики захворювань та інші.

Розгляду саме нескінченновимірних задач розбиття потребують випадки з дуже великою кількістю споживачів. У даному разі розглядання дискретної моделі приводить до значних труднощів, що пов'язані з розмірністю таких задач. Також існують задачі, в яких множина, що розбивається, неперервна одразу. Наприклад, це задача знаходження областей притягання локальних мінімумів деякої багатоекстремальної функції. З іншої сторони, неперервні моделі цікаві тим, що вони породжують негладкі задачі оптимізації.

Дана робота присвячена подальшому розвиненню теорії неперервних задач ОРМ для випадку нелінійних багатопродуктових задач оптимального розбиття множин з розташуванням центрів підмножин при обмеженнях у вигляді рівностей та нерівностей.

У [1] викладається математичну теорію неперервних задач оптимального розбиття множин (ОРМ) n -вимірного евклідового простору, які є некласичними задачами нескінченновимірного математичного програмування з булевими змінними.

Запропонована в [1] математична теорія неперервних задач оптимального розбиття множин полягає у зведенні початкових нескінченновимірних задач оптимізації через функціонал Лагранжа до негладких, як правило, скінченновимірних задач. Для чисельного розв'язку яких застосовуються сучасні ефективні методи недиференційованої оптимізації, а саме різні варіанти r -алгоритму, розробленого в інституті кібернетики ім. В. М. Глушкова НАН України під керівництвом Н. З. Шора.

У [2; 3] задачі з [1] узагальнюються на нелінійний випадок. Метою дослідження даної роботи є узагальнення однопродуктової нелінійної задачі ОРМ з розташуванням центрів при обмеженнях із [2; 3] на випадок багатопродуктової нелінійної задачі з розташуванням центрів при обмеженнях, постановка якої наведена в [4], а також теоретичне обґрунтування методу її розв'язання.

Постановка задачі. Розглянемо задачу з [4]. Нехай Ω – обмежена, вимірна за Лебегом, опукла множина у n-вимірному евклідовому просторі E_n. Сукупність вимірних за Лебегом підмножин $\Omega_1^1,...,\Omega_N^1$;

$\Omega_1^2,...,\Omega_N^2;\cdots;\ \Omega_1^M,...,\Omega_N^M$ із $\Omega \subset E_n$ назвемо можливим розбиттям множини,якщо $\bigcup_{i=1}^{N} \Omega_i^j = \Omega,\ j=1,2,...,M,\ mes(\Omega_i^j \cap \Omega_k^j) = 0,\ i \neq k,\ i,k = 1,...,N,$

$j = 1,...,M,$ де $mes(\cdot)$ – міра Лебега.

Позначимо клас всіх можливих розбиттів множини Ω через $\Sigma_\Omega^{N\times M}$,

тобто $$\Sigma_\Omega^{N\times M} = \left\{ \begin{array}{l} (\Omega_1^1,...,\Omega_N^1;...;\Omega_1^M,...,\Omega_N^M) : \bigcup_{i=1}^{N} \Omega_i^j = \Omega, \\ mes(\Omega_i^j \cap \Omega_k^j) = 0,\ i \neq k,\ i,k = 1,...,N,\ j = 1,...,M \end{array} \right\}.$$

Введемо функціонал

$$F(\{\Omega_1^1,...,\Omega_N^1;\ \Omega_1^2,...,\Omega_N^2;\cdots;\ \Omega_1^M,...,\Omega_N^M\},\{\tau_1,\tau_2,...,\tau_N\}) =$$
$$= \sum_{j=1}^{M}\sum_{i=1}^{N}\left[\varphi_i^j\left(\int_{\Omega_i^j} \rho^j(x)dx\right) + \int_{\Omega_i^j} c^j(x,\tau_i)\rho^j(x)dx\right].$$

Тут і у подальшому інтеграли розуміються у сенсі Лебега. Будемо вважати, що міра множини граничних точок підмножин $\Omega_i^j,\ i=1,...,N,\ j=1,...,M$ дорівнює нулю.

Функції $c^j(x,\tau_i)$ – дійсні, обмежені, визначені на $\Omega\times\Omega$, вимірні за аргументом x при будь-якому фіксованому $\tau_{\overline{i}} \left(\tau_i^{(1)},...,\tau_i^{(n)}\right)$ з Ω для будь-яких $i=1,...,N;$; функції $\rho^j(x)$ – дійсні, обмежені, вимірні і невід'ємні на Ω для будь-яких $j=1,...,M$; $\tau_{\overline{i}} \left(\tau_i^{(1)},...,\tau_i^{(n)}\right)$ – задана точка підмножини Ω_i^j, одна й та ж для усіх $j=1,...,M$, яку ми називаємо спільним центром підмножин $\Omega_i^1,...,\Omega_i^M$, не повинна належати кожному $\Omega_i^1,...,\Omega_i^M$; $\varphi_i^j(\cdot),\ i=1,...,N,\ j=1,...,M$ – дійсні, обмежені, опуклі, двічі неперервно-диференційовані функції свого аргументу.

Тоді під неперервною нелінійною багатопродуктовою задачею оптимального розбиття множини Ω з E_n на її неперетинні підмножини $\Omega_1^1,...,\Omega_N^1;\ \Omega_1^2,...,\Omega_N^2;\cdots;\ \Omega_1^M,...,\Omega_N^M$ із знаходженням координат центрів підмножин при обмеженнях у формі рівностей та нерівностей називатимемо наступну задачу.

Задача А.

$$\min_{(\{\Omega_1^1,\dots,\Omega_N;\dots;\Omega_1^M,\dots,\Omega_N\},\{\tau_1,\dots,\tau_N\})} F(\{\Omega_1^1,\dots,\Omega_N;\dots;\Omega_1^M,\dots,\Omega_N\},\{\tau_1,\dots,\tau_N\})$$

при обмеженнях

$$\sum_{j=1}^{M}\int_{\Omega_i^j}\rho^j(x)dx = b_i,\ i=1,\dots,p,$$

$$\sum_{j=1}^{M}\int_{\Omega_i^j}\rho^j(x)dx \le b_i,\ i=p+1,\dots,N,$$

$$\{\Omega_1,\dots,\Omega_N\}\in\Sigma_\Omega^{N\times M},\ \tau=(\tau_1,\dots,\tau_N)\in\underbrace{\Omega\times\dots\times\Omega}_{N}\in\Omega^N,$$

де $x=(x^{(1)},\dots,x^{(n)})\in\Omega$; $\tau_i=(\tau_i^{(1)},\dots,\tau_i^{(n)})\in\Omega$; $b_1,\dots,b_N$ – задані невід'ємні числа, причому виконуються умови розв'язності задачі

$$S=M\cdot\int_\Omega\rho^j(x)dx\le\sum_{i=1}^{N}b_i,\ 0<b_i\le S,\ i=1,\dots,N \qquad (1)$$

Переформулюємо задачу А в термінах характеристичних функцій $\lambda_i^j(x)$ підмножин Ω_i^j, $i=1,\dots,N_j$, $j=1,\dots,M$, а саме функцій виду:

$$\lambda_i^j(x)=\begin{cases}1, & x\in\Omega_i^j,\\ 0, & x\in\Omega\setminus\Omega_i^j,\end{cases}\quad i=1,\dots,N_j,\quad j=1,\dots,M$$

Розглянемо функціонал

$$I(\lambda(\cdot),\tau)=\sum_{j=1}^{M}\sum_{i=1}^{N}\left[\varphi_i^j\left(\iint_{\Omega\Omega}\rho^j(x)\lambda_i^j(x)dx\right)+\int_\Omega c^j(x,\tau_i)\rho^j(x)\lambda_i^j(x)dx\right]. \qquad (2)$$

де $\lambda(x)=(\lambda_1^1(x),\dots,\lambda_N^1(x);\dots;\lambda_1^M(x),\dots,\lambda_N^M(x))$.

Очевидно, що $I(\lambda(\cdot),\tau)=F(\{\Omega_1^1,\dots,\Omega_1^N;\dots;\Omega_1^M,\dots,\Omega_N^M\},\tau)$,

тоді *задача А* матиме вигляд:

Задача В.

$$\min_{(\lambda(\cdot),\tau)\in\Gamma_1\times\Omega^N} I(\lambda(\cdot),\tau),$$

де $\Gamma_1=\{\lambda(x):\lambda(x)\in\Gamma_1'$ майже всюди (м.в.) для $x\in\Omega$;

$$\sum_{j=1}^{M}\int_\Omega\rho\lambda_{ii}(x)\ (x)dx = b,\ i=1,\dots,p,$$

$$\sum_{j=1}^{M}\int_{\Omega}\rho_{ij}(x)\,\lambda_{ij}(x)\,dx \le b,\ i=p+1,\dots,N,\};$$

$$\Gamma_1' = \{\lambda(x)=(\lambda_1^1(x),\dots,\lambda_N^1(x);\dots;\lambda_1^M(x),\dots,\lambda_N^M(x)):\lambda_i^j(x)=0\vee 1 \text{ м.в. для } x\in\Omega,$$
$$i=1,\dots,N;\ j=1,\dots,M,$$

$$\sum_{i=1}^{N}\lambda_i^j(x)=1 \text{ м.в. для } x\in\Omega,\ j=1,\dots,M\};\ \tau=(\tau_1,\dots,\tau_N)\in\Omega^N.$$

Від нескінченновимірної *задачі B* математичного програмування з булевими значеннями змінних $\lambda_i^j(x)$, $i=1,\dots,N;\ j=1,\dots,M$ перейдемо до відповідної задачі зі значеннями $\lambda_i^j(x)$ з відрізка $[0,1]$:

Задача C.

$$\min_{(\lambda(\cdot),\tau)\in\Gamma_2\times\Omega^N} I(\lambda(\cdot),\tau),$$

де $\Gamma_2=\{\lambda(x):\lambda(x)\in\Gamma$ м.в. для всіх $x\in\Omega$;

$$\sum_{j=1}^{M}\int_{\Omega}\rho_{ij}(x)\,\lambda_{ij}(x)\,dx = b,\ i=1,\dots,p,$$

$$\sum_{j=1}^{M}\int_{\Omega}\rho_{ij}(x)\,\lambda_{ij}(x)\,dx \le b,\ i=p+1,\dots,N\};$$

$$\Gamma=\{\lambda(x)=(\lambda_1^1(x),\dots,\lambda_N^1(x);\dots;\lambda_1^M(x),\dots,\lambda_N^M(x)):0\le\lambda_i^j(x)\le 1,\ x\in\Omega,$$
$$i=1,\dots,N;\ j=1,\dots,M,$$

$$\sum_{i=1}^{N}\lambda_i^j(x)=1 \text{ м.в. для всіх } x\in\Omega\};\quad \tau=(\tau_1,\dots,\tau_N)\in\Omega^N.$$

Очевидно,
$$I(\lambda_*(\cdot),\tau_*)=\min_{(\lambda(\cdot),\tau)\in\Gamma_2\times\Omega^N} I(\lambda(\cdot),\tau)=\min_{\tau\in\Omega^N}\left(\min_{\lambda(\cdot)\in\Gamma_2} I(\lambda(\cdot),\tau)\right) \quad (3)$$

З результатів, отриманих в [1], [2] для аналогічної неперервної однопродуктової задачі оптимального розбиття множин з розташуванням центрів підмножин при кожному фіксованому $\tau\in\Omega^N$ та за узагальненою теоремою Вейерштрасса [5] внутрішня задача з (3) глобально розв'язна відносно $\lambda(\cdot)$ на обмеженій, замкненій, опуклій множині Γ_2 гільбертова простору $L_2^{N\times M}(\Omega)$.

Для обґрунтування методу розв'язку початкової задачі розглянемо для *задачі С* функціонал Лагранжа:

$$h(\{\lambda(\cdot),\tau\},\psi) = I(\lambda(\cdot),\tau) + \sum_{i=1}^{N}\psi_i\left[\sum_{j=1}^{M}\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx - b_i\right] =$$

$$= -\sum_{i=1}^{N}\psi_i b_i + \sum_{j=1}^{M}\sum_{i=1}^{N}[\varphi_i^j(\iint_{\Omega\Omega}\rho^j(x)\lambda_i^j(x)dx) + (c^j(x,\tau_i) + \psi_i\rho^j(x))\lambda_i^j(x)dx] \qquad (4)$$

де $\psi = (\psi_1,...,\psi_N)$ – N-вимірний вектор з дійсними компонентами, при чому $\psi_1,...,\psi_p$ – довільного знаку, а $\psi_{p+1},...,\psi_N$ – невід'ємні; $\lambda(x) \in \Gamma$ для $x \in \Omega$; $\tau = (\tau_1,...,\tau_N) \in \Omega^N$.

$(\{\lambda_*(\cdot),\tau_*\},\psi^*)$ – назвемо сідловою точкою функціонала (4) на множині $\{\Gamma\times\Omega^N\}\times\Lambda$, де $\Lambda = \{\psi = (\psi_1..\psi_N) \in E_N : \psi_i \geq 0, i = p+1,...,N\}$, якщо

$$h(\{\lambda_*(\cdot),\tau_*\},\psi) \leq h(\{\lambda_*(\cdot),\tau_*\},\psi^*) \leq h(\{\lambda(\cdot),\tau\},\psi^*) \qquad (5)$$

для будь-яких $\lambda(x) \in \Gamma$, $\tau \in \Omega^N$, $\psi \in \Lambda$.

Також можна записати: $h(\{\lambda_*(\cdot),\tau_*\},\psi^*) =$

$$= \max_{\psi\in\Lambda}\min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} h(\{\lambda(\cdot),\tau\},\psi) = \min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N}\max_{\psi\in\Lambda} h(\{\lambda(\cdot),\tau\},\psi), \qquad (6)$$

або $$h(\{\lambda_*(\cdot),\tau_*\},\psi^{**}) = \min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} h(\{\lambda(\cdot),\tau\},\psi) = \max_{\psi\in\Lambda} h(\{\lambda_*(\cdot),\tau_*\},\psi) \qquad (7)$$

Без доведення існування сідлової точки перейдемо до розв'язання задачі

$$\max_{\psi\in\Lambda}\min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} h(\{\lambda(\cdot),\tau\},\psi).$$

Якщо позначити $G(\psi) = \min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} h(\{\lambda(\cdot),\tau\},\psi)$, $\psi \in \Lambda$,

то двоїстою до *задачі С* буде задача:

$$G(\psi) \to \max, \ \psi \in \Lambda \qquad (8)$$

Для знаходження $$\min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} h(\{\lambda(\cdot),\tau\},\psi)$$

розглянемо $$\min_{\tau\in\Omega^N}\min_{\lambda(\cdot)\in\Gamma} h(\{\lambda(\cdot),\tau\},\psi) \quad \text{при} \quad \psi\in\Lambda. \qquad (9)$$

Позначимо в (9) $G_1(\tau,\psi) = \min_{\lambda(\cdot)\in\Gamma} h(\{\lambda(\cdot),\tau\},\psi)$, $\tau\in\Omega^N$, $\psi\in\Lambda$ (10)

Підставимо в (10) вираз для $h(\{\lambda(\cdot),\tau\},\psi)$ з (4), тоді отримаємо

$$G_1(\tau,\psi) = -\sum_{i=1}^{N}\psi_i b_i + \min_{\lambda(\cdot)\in\Gamma}\sum_{j=1}^{M}\sum_{i=1}^{N}\left[\varphi_i^j\left(\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right) + \left(c^j(x,\tau_i)+\psi_i\right)\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right] \tag{11}$$

$$\tau\in\Omega^N,\ \psi\in\Lambda\ .$$

Узагальнюючи результати, отримані в [1], [2], на випадок багатопродуктової нелінійної задачі, мінімальне значення по $\lambda(\cdot)$ функціоналу $h(\{\lambda(\cdot),\tau\},\psi)$ досягається для кожних $\tau\in\Omega^N$ і $\psi\in\Lambda$ при

$$(\lambda_1^1(x),\ldots,\lambda_i^j(x),\ldots,\lambda_N^M(x)) = (\lambda_{*1}^1(x),\ldots,\lambda_{*i}^j(x),\ldots,\lambda_{*N}^M(x)),$$

де

$$\lambda_{*i}^j(x) = \begin{cases} 1, & c^j(x,\tau_i)+\psi_i+\varphi'^j_{iY_i^j}\left(\int_{\Omega}\rho^j(x)\lambda_{*i}^j(x)dx\right) \le c^j(x,\tau_k)+\psi_k+\varphi'^j_{kY_k^j}\left(\int_{\Omega}\rho^j(x)\lambda_{*k}^j(x)dx\right), \\ i\neq k \text{ м.в. для } x\in\Omega \text{ (іншими словами, } i=k \text{ тільки на множині міри нуль,} \\ \text{тобто у точках границі між підмножинами } \Omega_i^j \text{ та } \Omega_k^j) \\ i,k=1,\ldots,N, \\ 0, \quad \text{в інших випадках.} \end{cases} \tag{12}$$

$$j=1,\ldots,M,$$

а функціонал $G_1(\tau,\psi)$ приймає вигляд

$$G_1(\tau,\psi) = -\sum_{i=1}^{N}\psi_i b_i + \min_{\lambda(\cdot)\in\Gamma}\sum_{j=1}^{M}\sum_{i=1}^{N}\left[\left(\varphi_i^j\left(\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right)-\varphi'^j_{iY_i^j}\left(\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right)\cdot\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right) + \left(\int_{\Omega}c^j(x,\tau_i)+\psi_i+\varphi'^j_{iY_i^j}\left(\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right)\right)\int_{\Omega}\rho^j(x)\lambda_i^j(x)dx\right], \tag{13}$$

$$\tau\in\Omega^N,\quad \psi\in\Lambda$$

де $Y_{ii}^j = \int_{\Omega}\rho\lambda_i(x)\ \ ^j(x)dx$

Підставляючи вираз для $\lambda_{*i}^{j}(x)$ з (12) в ту частину формули (13), яка лінійно залежить від $\lambda(\cdot)$ та залишаючи поки що змінною величину Y_i^j, $i = \overline{1,...,N}$, $j = \overline{1,...,M}$, отримаємо наступний вираз для $G_1(\tau,\psi)$

$$G_2(Y,\tau,\psi) = -\sum_{i=1}^{N}\psi_i b_i + \sum_{j=1}^{M}\sum_{i=1}^{N}\left[\left(\varphi_i^j(Y_i^j) - \varphi_{iY_i^j}'(Y_i^j)\cdot Y_i^j\right) + \int_{\Omega}\min_{k=\overline{1,N}}\left(c^j(x,\tau_k) + \psi_k + \varphi_{kY_k^j}'(Y_k^j)\right)\rho^j(x)dx\right], \quad (14)$$

$$\tau = (\tau_1,...,\tau_N) \in \Omega^N, \quad \psi = (\psi_1,...,\psi_N) \in \Lambda, Y \in U = \left\{Y = (Y_1^1,...,Y_N^1;...;Y_1^M,...,Y_N^M) \in E_{NM} : 0 \le \sum_{j=1}^{M} Y_i^j \le b_i,\ i = \overline{1,...,N}\right\}$$

Таким чином, двоїста задача (8) має вигляд:

$$\max_{\psi\in\Lambda}\min_{\tau\in\Omega^n}\max_{Y\in U} G_2(Y,\tau,\psi) \quad (15)$$

Враховуючи вище приведене та узагальнюючи на випадок *задачі С* результати, отримані в [1], [2] для аналогічної неперервної однопродуктової задачі оптимального розбиття множин з розташуванням центрів підмножин, можна зробити висновок, що має місце теорема Куна-Таккера у двоїстій формі, тобто $I(\lambda_*(\cdot),\tau_*) = G(\psi^*)$, де $(\lambda_*(\cdot),\tau_*)$ - оптимальний розв'язок задачі С, ψ^* - оптимальний розв'язок задачі (15), при чому максимум у двоїстій задачі (15) досягається.

Сформулюємо теорему, яка підсумовує наші розсуди та зумовлює перехід від нескінченновимірної *задачі С* до пошуку сідлової точки функціонала (4).

Теорема 1. *Нехай* $\varphi_i^j(\cdot)$, $i = \overline{1,...,N}$, $j = \overline{1,...,M}$ *– опуклі, двічі неперервно-диференційовані функції свого аргументу та при кожних фіксованих* $\tau \in \Omega^N$ *і* $\psi \in \Lambda$ *має місце умова*

$$mes\left\{x \in \Omega : \left(c^j(x,\tau_i) + \psi_i + \varphi_{iY_i^j}'\left(\int_{\Omega}\rho^j(x)\lambda_i^{*j}(x)dx\right)\rho^j(x)\right) = 0\right\} = 0, \quad i = \overline{1,...,N},$$

тоді сідлова точка $(\lambda_*(\cdot),\tau_*,\psi^*)$ *функціонала (4) для* $i = \overline{1,...,N}$ *та майже всіх* $x \in \Omega$ *визначається наступним чином:*

$$\lambda^j_{*i}(x)=\begin{cases}1, & \text{при } x\in\Omega^j_{*i} \text{ та } x\notin\Omega^j_{*q},\ q\le i,\\ 0, & \text{при } x\notin\Omega^j_{*i},\end{cases}\qquad i=1,...,N,\ j=1,...,M$$

де

$$\Omega^j_{*i}=\left\{\begin{array}{l}x\in\Omega: c^j(x,\tau^*_i)+\psi^*_k+\varphi'^{\,j}_{iY^j_i}(Y^{*j}_i)=\\ =\min\limits_{k=1,N}(c^j(x,\tau^*_k)+\psi^*_k+\varphi'^{\,j}_{kY^j_k}(Y^{*j}_k)),\ i\ne k \text{ м.с. для } x\in\Omega\end{array}\right\},$$

в якості $Y^{*j}_1,...,Y^{*j}_N$, $\tau^{**}_1,...,\tau^{**}_N$, $\psi^{**}_1,...,\psi^{**}_N$ *обирається оптимальний розв'язок двоїстої задачі (8), приведеної до виду*

$$G(\psi)=\min_{\tau\in\Omega^n}\max_{Y\in U}G_2(Y,\tau,\psi)=\min_{\tau\in\Omega^N}\max_{Y\in U}\left\{\sum_{i=1}^{N}-\psi_i b_i+\right.$$

$$\left.+\sum_{j=1}^{M}\sum_{i=1}^{N}\left[\left(\varphi^j_i(Y^j_i)-\varphi'^{\,j}_{iY^j_i}(Y^j_i)\cdot Y^j_i\right)+\int_{\Omega}\min_{k=1,N}\left(c^j(x,\tau_k)+\psi_k+\varphi'^{\,j}_{kY^j_k}(Y^j_k)\right)\rho^j(x)dx\right]\right\}\to\max,$$

при умовах $\psi_i\ge 0,\quad i=p+1,...,N.$

Висновки. Сформульована неперервна багатопродуктова нелінійна задача оптимального розбиття множини Ω з n-вимірного евклідова простору E_n на її неперетинні підмножини з розташуванням центрів цих підмножин при обмеженнях у формі рівностей та нерівностей у випадку опуклого цільового функціоналу. Здійснено перехід від названої нескінченновимірної задачі оптимізації до задачі пошуку сідлової точки функціонала Лагранжа.

Бібліографічні посилання

1. **Киселева Е. М.** Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М. Киселева, Н. З. Шор – К., 2005. – 564с.
2. **Киселёва Е. М.** Решение непрерывной нелинейной задачи оптимального разбиения множеств с размещением центров подмножеств для случая выпуклого целевого функционала / Е. М. Киселёва, М. С. Дунайчук // Кибернетика и системный анализ. – К., 2008. – № 2. – С. 134–152.
3. **Кісельова О. М.** Теорема Куна _Таккера для нелінійної неперервної задачі оптимального розбиття множин / О. М. Кісельова, М. С. Дунайчук // Математичне моделювання., 2007. – №2(17). – С. 41–45.

4. **Киселёва Е. М.** Об алгоритме решения непрерывной нелинейной многопродуктовой задачи оптимального разбиения множеств / Е. М. Киселёва, М. С. Дунайчук // Математика. Компьютер. Образование : XV междунар. конф., г. Дубна, 28 янв. – 2 фев., 2008 г. : тезисы докл. – М. – И. – 2008. – С.445.
5. **Васильев Ф. П.** Методы решения экстремальных задач. / Ф. П. Васильев – М., 1981. – 400 с.

Надійшла до редколегії 26.05.10

УДК 519.8

©**О.М. Кісельова, Л.С. Коряшкіна, О.В. Зайченко**
Дніпропетровський національний університет імені Олеся Гончара

ПРО ОДНУ ДИНАМІЧНУ ЗАДАЧУ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИНИ З НЕДИФЕРЕНЦІЙОВНИМ ФУНКЦІОНАЛОМ

Представлена неперервна задача оптимального розбиття множини, яка характеризується недиференційовним функціоналом та наявністю сімейства звичайних диференціальних рівнянь, що описують в кожній точці множини динаміку деякого параметра задачі залежно від приналежності точки певній підмножині вихідної множини.

Представлена непрерывная задача оптимального разбиения множества, которая характеризуется недифференцируемым функционалом и наличием семейства обыкновенных дифференциальных уравнений, описывающих динамику некоторого параметра задачи в зависимости от принадлежности точки определенному подмножеству исходного множества.

The dynamic continuous set partition problem is performed. This problem is being characterized by non-differential functional and presence of differential equations set, that describe dynamics of some problem's parameters depending on belonging a point to defined subset.

Ключові слова: оптимальне розбиття множини, диференціал Гато, метод лінеаризації.

Вступ. Неперервним задачам оптимального розбиття множин (ОРМ) в різних їх постановках (лінійних, детермінованих, одно- та багатопродуктових, стохастичних, динамічних, нечітких) присвячена велика кількість наукових публікацій. Докладна класифікація задач і методів ОРМ наведена в [1]. Представлена в даній роботі динамічна задача оптимального розбиття множини відрізняється від розглянутих раніше [1–3] по-перше, виглядом цільового функціонала; по-друге, нелінійною динамікою функції попиту, яка, у свою чергу, визначається приналежністю точки (споживача) до зони впливу певного виробника. У досліджуваній задачі розбиття передбачається статичним, за аналогією з [3] і на відміну від [2], де оптима-

льне розбиття множини може бути різним, тобто границі між підмножинами можуть бути рухомими.

Як і в [2;3] сформульовану задачу розбиття можна віднести до класу задач оптимального керування системами із зосередженими параметрами з особливою структурою множини допустимих керувань. Тому для її дослідження і розв'язання будемо поєднувати математичні апарати теорії неперервних задач ОРМ та теорії керування динамічними системами.

Слід ще раз наголосити, що постановка задачі містить функціонал, який є недиференційовним за Фреше, а має лише похідні за напрямами у функціональному просторі. Ця властивість вносить певні особливості до процесу розв'язання задачі. У роботі пропонується застосовувати метод послідовної лінеаризації Р.П. Федоренко, який добре зарекомендував себе при розв'язанні задач оптимального керування з функціоналами, що містять функції $|\cdot|$, $\max(\cdot)$ та інші [4].

Постановка задачі. Нехай $\Omega \subset R^2$ – деяка обмежена вимірна за Лебегом множина; $u_i(x,t) \in L_2(\Omega \times [0,T]), i = \overline{1,N}$ – заданий набір функцій; $T > 0$ – фіксована величина; $P_N(\Omega)$ – клас всіляких розбиттів множини Ω на N підмножин:

$$P_N(\Omega) = \{\bar{\omega} = (\Omega_1,...,\Omega_N): \bigcup_{i=1}^{N} \Omega_i = \Omega, \; mes(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = \overline{1,N}\}.$$

Потрібно

$$F(\bar{\omega}) = \int_{\Omega}\int_{0}^{T} |\rho(x,t;\bar{\omega}) - \xi(x,t)| dt dx \to \min_{\bar{\omega} \in P_N(\Omega)}, \qquad (1)$$

де функція $\rho(x,t;\bar{\omega})$ для кожного розбиття $\bar{\omega} = (\Omega_1, \cdots, \Omega_N) \in P_N(\Omega)$ і для кожного $x \in \Omega_i$ є розв'язком задачі Коші

$$\begin{aligned} &\rho(x,t) = \alpha\rho(x,t) - \beta\rho^2(x,t) + u_i(x,t), \quad t \in [0,T] \\ &\rho(x,0) = \rho_0(x), \end{aligned} \qquad (2)$$

$\alpha > 0, \beta > 0$ – задані величини; $\rho_0(x) \in L_2(\Omega), \xi(x,t) \in L_2(\Omega \times [0,T])$ – відомі функції.

Задачі (1), (2) можна надати таку економічну інтерпретацію. Нехай розглядається співіснування виробників однотипових товарів (подібних за ціною та якістю), споживачами яких є певний прошарок суспільства з приблизно однаковим рівнем накопичувань (чи доходів), які мешкають в однакових умовах та мають однакову шкалу побажань. Виробники товарів іншої якості співіснують в іншій ринковій ніші. Спільним інтересом для фірм-виробників однотипового товару є утримання попиту $\rho(x,t)$ на даний товар на певному рівні $\xi(x,t)$, який можливо лише дещо збільшувати з часом, зберігаючи при цьому головну тенденцію коливання в області оптимального для виробників попиту, за умов врахування граничної межі споживання даного товару та виробничих потужностей підприємства. Оскільки динаміка попиту прямо пропорційно залежить від кількості населення вказаного прошарку суспільства, тому будемо вважати, що зміна попиту прямо пропорційна зміні з часом чисельності населення [5]. Крім того, підвищення попиту на продукцію може здійснюватися за рахунок проведення суб'єктами виробництва певних дій, що пожвавлюють попит, а саме – незначного скорочення вартості товару за рахунок модернізації виробництва, рекламної акції, тощо. Сила впливу таких дій i-го виробника описується функцією $u_i(x,t)$, $1,\overline{N}$. Тому що виробники співіснують, немає сенсу кожному з них «вкладати кошти» в кожного споживача. Потрібно лише розподілити споживачів на зони впливу виробників з урахуванням їх спільних інтересів.

Метод розв'язання. Введення характеристичних функцій $\lambda_i(\cdot)$ підмножин Ω_i, $i=\overline{1,N}$ дозволяє перейти від (1), (2) до еквівалентної задачі

$$I(\lambda(\cdot))=\iint\limits_{\Omega\ 0}^{T}\left|\rho(x,t;\lambda(x))-\xi(x,t)\right|dtdx\to\min\limits_{\lambda\in\Lambda_1},\qquad(3)$$

де через $\rho(x,\cdot;\lambda(x))$ кожному $x\in\Omega$ відповідає розв'язок задачі Коші

$$\rho(x,t)=\alpha\rho(x,t)-\beta\rho^2(x,t)+\sum_{i=1}^{N}u_{ii}(x,t)\lambda(x),\quad t\in[0,T]$$

$$\rho(x,0)=\ _0(\ ),\qquad(4)$$

$$\Lambda_1 = \{\lambda(x) = (\lambda_1(x), \cdots, \lambda_N(x)) : \sum_{i=1}^{N} \lambda_i(x) = 1 \text{ м. в. для } x \in \Omega;$$

$$\lambda_i(x) = 0 \vee 1 \text{ м.в.для } x \in \Omega;\ i = \overline{1, N}\}.$$

Далі, згідно до теорії неперервних задач оптимального розбиття, множина Λ_1 занурюється в симплекс

$$\Lambda_0 = \{\lambda(x) = (\lambda_1(x), \cdots, \lambda_N(x)) : \sum_{i=1}^{N} \lambda_i(x) = 1 \text{ м. в. для } x \in \Omega;$$

$$0 \le \lambda_i(x) \le 1 \text{ м.в.для } x \in \Omega;\ i = \overline{1, N}\}$$

і задача (3), (4) зводиться до задачі

$$I(\lambda(\cdot)) \to \min_{\lambda \in \Lambda_0}, \tag{5}$$

за умов (4).

Задачу (5), (4) будемо розглядати як задачу оптимального керування, в якій керуючою функцією для кожного $x \in \Omega$ виступає функція $u(x,t) = \sum_{i=1}^{N} u_i(x,t)\lambda_i(x), \lambda \in \Lambda_0.$ Застосуємо метод, в основі якого лежить ідея побудови мінімізуючої послідовності функцій $\lambda^k(\cdot) \in \Lambda_0$.

Схема методу.

1. $k = 0$. Фіксується $x \in \Omega$. Задається $\lambda^0(\cdot) \in \Lambda_0$.
2. Для кожного $x \in \Omega$ з функцією $u^{(k)}(x,t) = \sum_{i=1}^{N} u_i(x,t)\lambda_i^{(k)}(x)$ розв'язується задача Коші (4).
3. Обчислюється значення функціонала $I(\lambda(\cdot))$ з (5).
4. В околі незбуреного процесу $\left\{\sum_{i=1}^{N} u_i(x,t)\lambda^k(x), \rho(x,t)\right\}$ задача лінеаризується. Процес лінеаризації включає два этапи: 1) обчислення диференціалу Гато $D(\lambda^k(\cdot), v^k(\cdot)) = D(\lambda^k(\cdot), \overline{\lambda}(\cdot) - \lambda^k(\cdot))$; $\overline{\lambda}(\cdot) \in \Lambda_0$; 2) побудову малого околу $\delta\Lambda_0$ незбуреної функції $\lambda^k(\cdot)$.

Формулюється і розв'язується задача: знайти приріст (збурення) $\nu^k(\cdot) = \bar{\lambda}(\cdot) - \lambda^k(\cdot)$ функції $\lambda^k()$ з умови

$$\min_{\lambda(\cdot)\in\Lambda_0} D(\lambda^{(k)}(\cdot), \lambda(\cdot) - \lambda^{(k)}(\cdot)) = D(\lambda^{(k)}(\cdot), \bar{\lambda}(\cdot) - \lambda^{(k)}(\cdot)). \qquad (6)$$

Ця задача є лінеаризацією задачі, що розв'язується, в околі незбуреного процесу $\left\{\sum_{i=1}^{N} u_i(\cdot,t)\lambda^k(\cdot); \rho(\cdot,t)\right\}$. Розв'язання задачі (6) дозволяє здійснити основний крок процесу – перехід до наступного наближення

$$\lambda^{k+1}(\cdot) = \lambda^k(\cdot) + s(\bar{\lambda}(\cdot) - \lambda^k(\cdot)). \qquad (7)$$

Параметр $s \in [0,1]$ підбирається, задовольняючи умову релаксації:

$$I(\lambda^{k+1}(\cdot)) < I(\lambda^k(\cdot)). \qquad (8)$$

Критерієм закінчення ітераційного процесу може виступити одна з нерівностей

$$\left\|\lambda^{k+1}(\cdot) - \lambda^k(\cdot)\right\| \le \varepsilon; \quad \left|I^{k+1} - I^k\right| \le \varepsilon;$$
$$\left|D(\lambda^k(\cdot), \bar{\lambda}(\cdot) - \lambda^k(\cdot))\right| \le \varepsilon.$$

Диференціал Гато функціоналу задачі (5). Відомо, що $I(\lambda(\cdot))$ буде *диференційовним за Гато*, якщо для будь-якої вектор-функції $\nu(\cdot)$ (з того ж простору, що і $\lambda(\cdot)$) має місце формула

$$I(\lambda(\cdot) + s\nu(\cdot)) = I(\lambda(\cdot)) + sD(\lambda(\cdot), \nu(\cdot)) + o(s), \qquad (9)$$

де s - довільне мале невід'ємне число, $D(\lambda(\cdot), \nu(\cdot))$ – деякий функціонал, що називається похідною функціонала I в точці $\lambda(\cdot)$ за напрямом $\nu(\cdot)$ (диференціалом Гато).

Для кожної точки $x \in \Omega$ введемо до розгляду множини:

$$\Theta^0(x) = \{t \in [0,T]: \rho(x,t;\lambda(x)) - \xi(x,t) = 0\};$$
$$\Theta^+(x) = \{t \in [0,T]: \rho(x,t;\lambda(x)) - \xi(x,t) > 0\};$$
$$\Theta^-(x) = \{t \in [0,T]: \rho(x,t;\lambda(x)) - \xi(x,t) < 0\}.$$

Варіацією функції $\lambda^{(k)}(\cdot) \in \Lambda_0$ будемо вважати функцію $v(\cdot) = \lambda(\cdot) - \lambda^{(k)}(\cdot)$, де $\lambda(\cdot) \in \Lambda_0$.

Дамо приріст $\delta\lambda(\cdot) = sv(\cdot)$ функції $\lambda^{(k)}(\cdot)$, де $s > 0$ - достатньо малий параметр. Цьому приросту відповідає приріст фазової траєкторії $\delta\rho$, що задовольняє лінеаризовану задачу Коші

$$\dot{\delta\rho} = \alpha(\delta\rho) - 2\beta\rho(\delta\rho) + \sum_{i=1}^{N} u_i(x,t)sv_i(x);$$
$$\delta\rho(x,0) = 0 \quad \forall x \in \Omega. \tag{10}$$

Розглянемо значення

$$I(\lambda^{(k)}(\cdot) + \delta\lambda(\cdot)) = \iint_{\Omega\,0}^{T} \left|\rho(x,t;\lambda^{(k)}(x)) - \xi(x,t) + \delta\rho(x,t) + O(\|\delta\lambda\|^2)\right| dtdx =$$

$$= \int_{\Omega} \left\{ \int_{\Theta^0(x)} |\delta\rho(x,t)|dt + \int_{\Theta^+(x)} \left\{\rho(x,t;\lambda^{(k)}(x)) - \xi(x,t) + \delta\rho(x,t)\right\} dt - \right.$$

$$\left. - \int_{\Theta^-(x)} \left\{\rho(x,t;\lambda^{(k)}(x)) - \xi(x,t) + \delta\rho(x,t)\right\} dt \right\} dx + O(s)$$

Порівнюючи цей вираз з (9), заключаємо, що диференціал Гато функціонала $I(\lambda(\cdot))$ обчислюється за формулою

$$D(\lambda^0(\cdot), v(\cdot)) = \int_{\Omega} \left\{ \int_{\Theta^0(x)} |\delta\rho(x,t)|dt + \int_{\Theta^+(x)} \delta\rho(x,t)dt - \int_{\Theta^-(x)} \delta\rho(x,t)dt \right\} dx.$$

Використовуючи рівняння у варіаціях (10) та тотожність Лагранжа, цей вираз можна записати в еквівалентному вигляді відносно приросту $sv(\cdot)$:

$$D(\lambda^0(\cdot), v(\cdot)) =$$

$$= s\int_{\Omega} \left\{ \int_{\Theta^0(x)} \left| \int_{0}^{T} \sum_{i=1}^{N} u_i(x,t) v_i(x) \psi(x,t,t'')dt \right| dt - \int_{\Theta^0(x)} u_i(x,t) v_i(x) \psi(x,t,t)dt \right\} dx$$

,

де для кожного $x \in \Omega$ функція $\psi(x,\cdot)$ є розв'язком задачі Коші

$$-\psi_t(x,t) + (\alpha - 2\beta\rho\delta)\rho\psi(x,t) = \begin{cases} 1, & t \in \Theta^+(x); \\ 0, & t \in \Theta^0(x); \\ -1, & t \in \Theta^-(x); \end{cases} \tag{11}$$

$$\psi(x,T) = 0 \quad x \in \Omega; \quad t \in [0,T], \tag{12}$$

а для кожного $t' \in \Theta^0(x)$ функція $\psi(x,t,t')$ – розв'язок задачі

$$\delta(t-t')\psi(x,t,t') + (\alpha - 2\beta\rho)\psi_t(x,t,t') = \delta(t-t'), \tag{13}$$

$$\psi(x,T,t') = 0 \quad x \in \Omega., \tag{14}$$

$\delta(t-t')$ – функція Дирака. Підкреслимо, що у випадку, коли $mes\,\Theta^0(x) = 0$ майже всюди для $x \in \Omega$, функціонал (5) буде диференційовним за Фреше, і його похідна обчислюється після розв'язання задачі Коші (11)–(12) за формулою

$$\frac{\partial I(\lambda(\cdot))}{\partial \lambda_i(\cdot)} = \int_0^T \psi(\cdot,t)u_i(\cdot,t)dt,$$

що відповідає результатам, отриманим у процесі дослідження задач типу (4) – (5) з лінійним критерієм якості.

Зауваження щодо чисельної реалізації алгоритму розв'язання задачі. Під час реалізації наведеної вище схеми розв'язання задачі (4) – (5) область Ω покривається рівномірною сіткою з кроком h_x по змінній x, h_y - по змінній y:

$$\gamma = \left\{(x_i, y_j): x_i = x_0 + ih_x, y_j = y_0 + ih_y, i = \overline{1,m}; j = \overline{1,m1}\right\}.$$

Функцією $\lambda(\cdot)$, яка є розв'язком вихідної задачі, буде кусково -стала функція

$$\lambda(q) = \lambda_{i+1/2,j+1/2}(q), \quad q = (x,y); \quad x_i \le x \le x_{i+1}; \; y_j \le y \le y_{j+1}.$$

Варіації $\nu(\cdot)$ шукаємо в тому ж самому класі функцій.

Задається початкове наближення $\lambda^0(\cdot)$. Розв'язується задача Коші (4), наприклад, методом Рунге -Кутта тим самим обчислюються значення траєкторії $\rho(\cdot,\cdot)$: $\rho_{ij}^l \quad \rho(x_i, y_j; t_l)$.

Для кожної точки q сітки γ обчислення значення похідної Гато функціонала (5) в точці $\lambda^k(\cdot)$ потребує інтегрування задач Коші (11), (12) та (13), (14). При цьому слід виділити множини індексів $M_\varepsilon^-, M_\varepsilon^+, M_\varepsilon^0$ следующим образом:

$$l \in M_{ij}^-, \text{ якщо } \rho_{ij}^l - \xi(x_i, y_j; t_l) < -\varepsilon;$$
$$l \in M_{ij}^+, \text{ якщо } \rho_{ij}^l - \xi(x_i, y_j; t_l) > \varepsilon;$$
$$l \in M_{ij}^0, \text{ якщо } \left|\rho_{ij}^l - \xi(x_i, y_j; t_l)\right| \le \varepsilon, \ \varepsilon > 0.$$

Величину $\varepsilon > 0$ слід обирати якомога менше, але з урахуванням того, що складність розв'язання задачі пов'язана з кількістю індексів у множині I_{ij}^0. У той же час ε не повинне бути надто малим, щоб уникнути переходу за один крок процесу будь_якого індексу l з I_{ij}^- в I_{ij}^+.

Далі записується скінченновимірна апроксимація задачі (6). Замінивши у виразі похідної Гато інтеграли інтегральними сумами, отримуємо наступну задачу для обчислення функції $\overline{\lambda}(\cdot)$

$$\sum_{p=1}^{N}\sum_{i=1}^{m}\sum_{j=1}^{m1}\sum_{l=1}^{m2} \overline{A}_{ij}^l u_{ijl}^p (\lambda_{ij}^p - \lambda_{ij}^{0p}) + \left| \sum_{(i,j)\in\gamma} \sum_{p=1}^{N} \sum_{l\in M_{ij}^0} A_{ij}^l u_{ijl}^p (\lambda_{ij}^p - \lambda_{ij}^{0p}) \right| \to \min_{\lambda},$$

за умов

$$0 \le \lambda_{ij}^p \le 1, \ p = \overline{1, N}; \qquad \sum_{p=1}^{N} \lambda_{ij}^p = 1; \ i = \overline{1, m}; \ j = \overline{1, m1}.$$

Ця задача зводиться до задачі лінійного програмування [4] і розвязується симплекс_методом.

Як результат методу послідовної лінеаризації отримуємо в кожній точці сітки γ значення вектор_функції $\lambda*(\cdot)$, яка є розв'язком задачі (4), (5). Ця функція може містити або всі цілі компоненти (і тоді вона є також оптимальним розв'язком задачі (3), (4)), або декілька дробових значень, які можна округлити, наприклад за таким правилом: для кожної точки сітки γ більшій компоненті вектора $\lambda*(\cdot)$ відповідає значенняе цієї компоненти вектора $\lambda^{**}(\cdot)$, що дорівнює 1, решта компонент вектора $\lambda^{**}(\cdot)$ буде дорівнювати нулю. Звичайно, при зростанні вимірності задачі, тобто при

зростанні кількості вузлів сітки γ, дробові значення будуть з'являтися частіше, і тоді для отримання розвязку задачі (3), (4) слід використовувати методи цілочисельного лінійного програмування.

Висновки. За межами роботи залишаються відкритими питання: обґрунтування можливості дискретизації області Ω і заміни сімейства задач Коші (2) скінченою їх кількістю. Доцільним також є врахування факторів міграції споживачів та конкуренції між виробниками, які можуть суттєво впливати на вибір споживача.

Бібліографічні посилання

1. **Киселева Е.М.** Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М. Киселева, Н.З. Шор – К., 2005. – 564.

2. **Коряшкіна Л.С.** Особливості розв'язку задачі оптимального розбиття множин з рухомими границями між підмножинами / Л.С. Коряшкіна, Т.О. Шевченко – Питання прикладної математики і математичного моделювання. Збірник наук. праць, 2008. – С.167-178.

3. **Коряшкіна Л.С.** Нові підходи до розв'язання динамічної задачі оптимального розбиття множини / Л.С. Коряшкіна, Т.О. Шевченко – Питання прикладної математики і математичного моделювання. Збірник наук. Праць, 2009. – С.220 – 231.

4. **Федоренко Р.П.** Приближенные методы решения задач оптимального управления. / Р.П. Федоренко – М., 1978.

5. **Капица С.П.** Общая теория роста человечества. / С.П. Капица – М., 1998.

Надійшла до редколегії 05.05.10

УДК 519.85

©**А.И. Косолап**

Днепропетровский национальный университет имени Олеся Гончара

РЕШЕНИЕ ЗАДАЧИ НЕВЫПУКЛОЙ КВАДРАТИЧНОЙ ОПТИМИЗАЦИИ

Розглядається загальна задача квадратичної оптимізації, яка відноситься до класу NP-складних. Для її розв'язку використовується метод напіввизначеної релаксації. Для розв'язку задачі напіввизначеного програмування пропонуеться узагальнений симплекс-метод.

Рассматривается общая задача квадратичной оптимизации, которая принадлежит классу NP-сложных. Для ее решения используется метод полуопределенной релаксации. Для решения задачи полуопределенного программирования предлагается обобщенный симплекс-метод.

We consider the general problem of quadratic optimization which belongs to NP-hard class. For its solution we use the method of a semidefinite relaxation. For the solution of the problem of semidefinite programming we offer the generalised simplex-method.

Ключевые слова: невыпуклая квадратичная оптимизация, полуопределенная релаксация, прямо -двойственный метод внутрешней точки, квадратичная релаксация, симплекс-метод.

Введение. Многие задачи из области экономики, финансов, оптимизации проектов, планирования, компьютерной графики, управления сложными системами преобразуются к задачам квадратичной оптимизации в конечномерном пространстве, когда целевая функция и ограничения содержат общие квадратичные функции. Такие задачи содержат множество локальных минимумов и относятся к классу NP -сложных. Допустимое множество этих задач может быть несвязным и даже дискретным.

Кроме множества приложений, интерес к этим задачам стимулируется унифицированной структурой входных данных при разработке соответствующего программного обеспечения. Входными данными задачи являются симметрические матрицы и векторы.

На сегодняшний день эффективно разрешимой является только задача минимизации общей квадратичной функции при одном выпуклом квадратичном ограничении [1]. Для задачи с двумя квадратичными ограничениями эффективные алгоритмы разработаны только для некоторых частных случаев [2].

Одним из общих подходов при решении этого класса задач является полуопределенная релаксация [3–4]. В этом случае квадратичная функция $x^T Ax$ представляется в виде Axx^T или $A \cdot X$, где X – положительно полуопределенная матрица ранга единица. Такое преобразование позволяет преобразовать общую квадратичную задачу к линейной задаче полуопределенной оптимизации, в которой неизвестной является полуопределенная матрица. Задача полуопределенной оптимизации является эффективно разрешимой. Однако без требования, чтобы ранг искомой матрицы был равен единице, полуопределенная релаксация является приближенным преобразованием. Множество всех полуопределенных матриц образует выпуклый конус, крайними лучами (образующими) которого являются полуопределенные матрицы ранга единицы. Число таких матриц ранга единица бесконечно, поэтому конус полуопределенных матриц не является многогранным. Однако не каждый граничный луч является крайним. Поэтому, если решение преобразованной задачи полуопределенной оптимизации достигается в крайнем луче конуса полуопределенных матриц, то полуопределенная релаксация будет точной. Решение линейной задачи полуопределенной оптимизации не является простым, так как условие положительной определенности матрицы не может быть задано в явном виде. Положительная определенность матрицы равносильна положительности ее минимального собственного значения.

Другие общие подходы к решению общей задачи квадратичной оптимизации используют схемы методов ветвей и границ, которые предусматривают построение дерева подзадач посредством разбиения допустимой области на множество подобластей и сравнение решений на каждой из этих подобластей. Процесс разбиения подобласти завершается, если на ней найдена точка глобального минимума. Очевидно, что такой подход может быть эффективен только для задач малой размерности, которые не представляют практической значимости [5]. Существуют и другие подходы, использующие двойственность [6] или декомпозицию [7]. Однако и в этих случаях гарантируется получение только приближенных оценок. Поэтому проблема эффективного решения общих квадратичных задач до настоящего времени остается открытой.

Постановка задачи и алгоритм ее решения. Рассмотрим общую задачу квадратичной оптимизации

$$\min\{f_0(x) | f_i(x) \le 0,\ i = 1,\ldots,\ m,\ x \in E^n\}, \qquad (1)$$

где все функции $f_i(x) = x^T A_i x + b_i^T + c_i$ – квадратичные, b_i, x – векторы n-мерного евклидового пространства, c_i – константы, а все матрицы A_i – симметричны. Использование полуопределенной релаксации задачи (1) преобразует ее к линейной задаче полуопределенной оптимизации

$$\min\{Q_0 \cdot X | Q_i \cdot X \le 0,\ i = 1,\ldots,m,\ X \ge 0\},$$

где X – положительно полуопределенная матрица

$$X = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix},$$

($X \cdot Y = \sum\sum x_{ij} y_{ij}$), а

$$Q_0 = \begin{pmatrix} 0 & b_0^T / 2 \\ b_0 / 2 & A_0 \end{pmatrix},$$

$$Q_i = \begin{pmatrix} c_i & b_i^T / 2 \\ b_i / 2 & A_i \end{pmatrix}.$$

Для ее решения используют эффективный прямо_двойственный метод внутренней точки [8]. Он используется, когда решение прямой задачи (2) и соответствующей двойственной задачи совпадают. Однако это будут не всегда. Разрыв двойственности может быть даже в случае, когда прямая и двойственная задачи будут иметь решения. Рассмотрим использование симплекс_метода для решения задачи полуопределенной оптимизации. Запишем задачу полуопределенного программирования в каноническом виде

$$\min\{C \cdot X | A_i \cdot X = b_i,\ i = 1,\ldots,\ m,\ X \ge 0\}. \qquad (3)$$

Известно, что любая положительно определенная матрица X ранга k может быть представлена в виде линейной комбинации матриц ранга 1

$$X = \sum_{j=1}^{k} \alpha_j x^j (x^j)^T, \alpha \ge 0. \qquad (4)$$

Подстановка выражения (4) в (3) приводит к задаче линейного программирования

$$\min\{\sum_{j=1}^{k}\alpha_j C\cdot X_i \mid \sum_{j=1}^{k}\alpha_j A_i\cdot X_j = b_i, i=1,...,m, \alpha \geq 0\}. \qquad (5)$$

Первоначально в качестве векторов x^j достаточно взять векторы с компонентами 0, +1, – 1. Обозначим через B базисную матрицу оптимального решения задачи (5). Будем искать новый вектор x^s для которого текущее базисное решение не является оптимальным. Если такого вектора не существует, то будем искать большее число векторов для которых сумма (4) не будет оптимальным решением. Если таких векторов не существует, то текущее базисное решение является оптимальным для задачи (4). Новый столбец матрицы ограничений будет иметь вид

$$B^{-1}A_iX_s,$$

а строка целевой функции

$$C\cdot X_s - \sum_{j=1}^{k} C\cdot X_{B(j)}B^{-1}A_j X_s, \qquad (6)$$

где $B(j)$ – множество индексов базисных переменных. Условие (6) перепишем в виде

$$(C - \sum_{j=1}^{m} C\cdot X_{B(j)}B^{-1}A_j)X_s,$$

или

$$Q\cdot X_s = (x^s)^T Q x^s,$$

где

$$Q = C - \sum_{j=1}^{m} C\cdot X_{B(j)}B^{-1}A_j.$$

Если матрица Q – положительно определенная (непосредственной проверкой убеждаемся, что матрица Q – симметрическая), то $Q\cdot X_s > 0$ и текущее базисное решение оптимально для задачи (3) (не существует матрицы X_s для которой значение целевой функции задачи (5) можно уменьшить).

Для проверки положительной определенности симметрической матрицы Q достаточно решить задачу

$$\min\{x^T Q x \mid \|x\|^2 = 1\}. \qquad (7)$$

Если в точке минимума $x^T Q x > 0$, то матрица Q – положительно определенная. Воспользуемся метод квадратичной релаксации для решения задачи (7). Преобразуем ее к эквивалентной

$$\min\{x_{n+1} | x^T Qx + q \le x_{n+1}, \|x\|^2 = 1\}, \qquad (8)$$

где $q > 0$, такое, что $x^T Qx + q > 0$. Преобразуем пространство переменных $z = Px$, где матрица P равна

$$p = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ z_1 & z_2 & \dots & z_{n+1} \end{pmatrix}.$$

Задача (8) преобразуется к виду

$$\min\{\|z\|^2 | z^T Qz + q \le \|z\|^2, \|z\|^2 = 1\}.$$

Следующее преобразование позволяет преобразовать матрицу Q к положительно определенной

$$\min\{\|z\|^2 | z^T Q^* z + q \le d, r\|z\|^2 = d, \|z\|^2 = 1\}. \qquad (9)$$

где $r > 0$ такое, что матрица $Q^* = Q + (r - 1)I$ – положительно определенная (матрица с преобладающей главной диагональю). Теперь достаточно решить задачу

$$\min\{x^T Q^* x \|x\|^2 = 1\}. \qquad (10)$$

Решения задачи (9) будет найдено из решения последней задачи умножением на положительную константу, которая не будет влиять на положительную определенность матрицы Q. При хорошем начальном приближении, решение задачи (10) может быть получено в явном виде. Решим последовательность задач

$$\max\{e^i x | x^T Q^* x = 1\},\ i = 1, \dots, n + 1,$$

где e^i – единичные вектора. Решения этих задач равны

$$x = \frac{(Q^*)^{-1} e^i}{\sqrt{(e^i)^T (Q^*)^{-1} e^i}}.$$

Определим индекс решения с максимальным по норме значением и выберем это решение в качестве начального приближения. Затем, решим последовательность задач

$$\max\{(x^s)^T x | x^T Q^* x = 1\}.$$

После k итераций будет найдено приближенное решение задачи (10)

$$x^k = \frac{(Q^*)^{-k} e^s}{\sqrt{q_{ss}^{-(2k-1)}}}.$$

Теперь для проверки положительной определенности матрицы Q достаточно вычислить выражение $(x^k)^T Q x^k$, если оно больше нуля, то матрица $Q-$ положительно определенная, в противном случае включение матрицы $X_s = x^k (x^k)^T$ приведет на следующей итерации симплекс -метода к уменьшению целевой функции.

Покажем, что симплекс_метод для задачи полуопределенного программирования будет сходиться к решению за конечное число итераций. Рассмотрим текущий и оптимальный базис. Эти базисы определяют некоторое многогранное множество. Каждый базис также определяет многогранное множество. Множество граней, образованное текущим базисом и одной из вершин оптимального базиса будет разделяющей гранью этих двух базисов. Так как допустимое множество задачи (3) выпуклое и связное, то по крайней мере одна разделяющая грань пересечет это множество в точке с меньшим значением целевой функции. Таким образом, будет получен новый базис с меньшим значением целевой функции. Продолжая этот процесс, мы за конечное число итераций перейдем к оптимальному базису.

Так как оптимальный базис является линейной комбинацией матриц ранга единица, то это решение может быть недопустимым для задачи квадратичной оптимизации. Для получения допустимого базисного решения, на каждой итерации симплекс_метода необходимо решать следующую задачу

$$\min\{x^T Q x | x^T A_i x = b_i,\ i = 1,\ldots,\ m,\ \|x\|^2 = 1\}.$$

Использование метода квадратичной релаксации приводит к сложной задаче поиска максимума квадрата нормы вектора на пересечении эллипсоидов с общим центром [9]. Если для некоторого промежуточного базиса только одно значение $\alpha_i > 0$ (базис существенно вырожденный), то этот базис будет допустимым для задачи квадратичной оптимизации и будет верхней границей решения.

Поиск начального базисного решения зависит от начального выбора матриц X_j. Их линейная комбинация может не содержать допустимых точек задачи (3). Тогда в базис необходимо ввести произвольную матрицу X_j, для которой новая линейная комбинация будет содержать допустимые точки задачи (3). Для исключения этой матрицы из базиса, достаточно заменить целевую функцию задачи (5) на $E \cdot X_j$, где элементы матрицы E

равны e_{ij} =1, если соответствующие элементы матрицы X_i положительны и $e_{ij} = -1$, в противном случае. Очевидно, что если допустимое базисное решение задачи (5) существует, то $E \cdot X_j = 0$ и будет найден допустимый базис этой задачи, после чего возвращаемся к исходной целевой функции. Это есть реализация метода искусственного базиса для задачи (5).

Пример 1. Рассмотрим задачу квадратичной оптимизации

Найти

$$\min \|x\|^2$$

при ограничениях:

$$8x_1^2 - x_2^2 - 2x_1x_2 - 80x_1 + 12x_2 + 100 \le 0,$$

$$8x_1^2 - x_2^2 + 24x_2 + 48 \le 0.$$

Ее допустимая область представлена на рис. 1. Задача имеет два локальных минимума.

Метод полуопределенной релаксации позволяет получить решение $x^0 = (0.879, -2.169)$, которое близко к оптимальному $x^* = (1.029, -2.159)$.

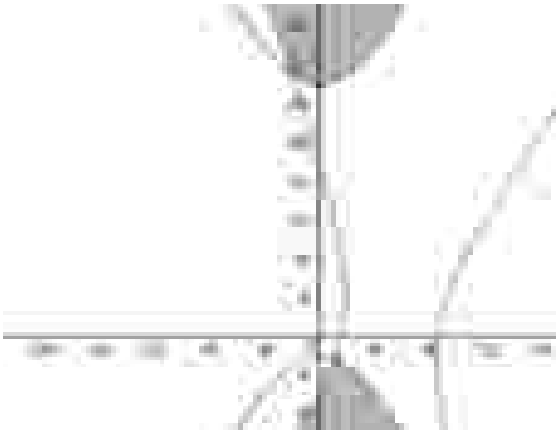


Рис. 1. Два локальных минимума

Пример 2. Найти

$$\min \|x\|^2 \qquad (7)$$

при ограничениях:

$$-10x_1^2 + x_2^2 - 8x_1x_2 - 2x_1 + x_2 + 1 \le 0,$$

$$x_1^2 - 10x_2^2 + 10x_1x_2 + x_1 - 2x_2 + 3 \le 0.$$

В этой задаче 4 локальных минимума, см. рис. 2.

Метод полуопределенной релаксации дает решение $x^0 = (0.435, 0.363)$, которое является приближенным для оптимального $x^* = (0.2174, 0.5802)$.

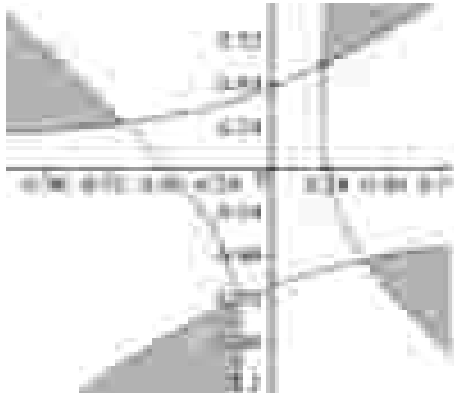


Рис.2. Четыре локальных минимума

Выводы. В работе используется полуопределенная релаксация для решения общей задачи квадратичной оптимизации. Для решения задачи полуопределенного программирования предлагается обобщение симплекс_метода. Доказана его конечная сходимость. Проведены численне эксперименты, свидетельствующие об эффективности выбранного метода решения невыпуклых квадратичных задач.

Библиографические ссылки

1. **Fortin C**. Computing the local minimizers of a large and sparse trust region subproblem / C. Fortin. – Montreal: McGill University, 2004. – 149 p.
2. **Beck A.** Strong duality in nonconvex quadratic optimization with two quadratic constraints / A. Beck, Y. C. Eldar // SIAM J. Optim., 17(3), 2006. – P. 844-860.
3. **Lasserre J.** Global optimization with polynomials and the problem of moments / J. Lasserre // SIAM J.Optim., Vol. 11, N. 3, 2001. – P. 796–817.
4. **Ding Y.** On Efficient Semidefinite Relaxations for Quadratically Constrained Quadratic Programming / Y. Ding. – Waterloo, Ontario, Canada. – 2007. – 68 p.
5. **Horst R.** Global Optimization: Deterministic Approaches, 3rd ed. / R. Horst, H. Tuy. - Springer-Verlag, Berlin, 1996.

6. **Шор Н.З.** Квадратичные экстремальные задачи и недифференцируемая оптимизация / Н. З. Шор, С.И. Стеценко. – К., 1989.– 205с.
7. **Floudas C.A.** Quadratic optimization / C. A. Floudas, V. Visweswaran. – Princeton : Princeton University. – 1995. – 53 p.
8. **Todd M. J.** Semidefinite optimization / M. J. Todd //Acta Numerica, 10, 2001. – P. 515–560.
9. **Nemirovski A.**, Roos C., Terlaky T. On maximization of quadratic form over intersection of ellipsoids with common center / A. Nemirovski, C. Roos, T. Terlaky // Mathematical Programming. – 1999. – Vol. 86. – P. 463 – 473.

Надійшла до редколегії 23.04.10

УДК 519.81

©**В.В. Крючковский, К.Э. Петров**
Херсонский национальный технический университет

СВЯЗЬ И ВЗАИМНАЯ ТРАНСФОРМАЦИЯ ВЕЛИЧИН С РАЗЛИЧНЫМИ ВИДАМИ НЕОПРЕДЕЛЕННОСТИ ПРИ ПРИНЯТИИ РЕШЕНИЙ

При прийнятті рішень можливе взаємне перетворення інтервальних величин з різною формою завдання переваги. При цьому в більшості випадків виникає необхідність евристичного заполнення інформації, якої бракує.

При принятии решений возможно взаимное преобразование интервальных величин с различной формой задания предпочтительности. При этом в большинстве случаев возникает необходимость эвристического восполнения недостающей информации.

While decision making the interconversion of interval values is possible with the different form of preference assignment. At that in most cases there is a necessity of heuristic filling in of failing information.

Ключевые слова: принятие решений, информация, неопределенность.

Введение. Необходимыми условиями эффективности любого решения являются своевременность его принятия, полнота и оптимальность. Однако, неопределенность информации, используемой при моделировании, неясность и противоречивость целей, критериев оценки вариантов развития создают большое количество проблем, затрудняющих создание моделей, адекватных рассматриваемому объекту или процессу. При этом неопределенность исходной информации носит, как правило, интервальный характер. Это означает, что корректное решение проблемы принятия эффективных решений связано с развитием методологии принятия многокритериальных решений в условиях интервальной неопределенности [1]. Очевидно, первая попытка принятия решении в условиях неопределенности была предпринята Яковом Бернулли (1654–1705) в его книге «Искусство предположений» [2]. Именно на принцип недостаточного основания (который гласит, что если распределение вероятностей состояний природы неизвестно, то нет причин считать их различными) Я. Бернулли опирается критерий Лапласа [3]. Важнейшее понятие теории принятия решений в

условиях неопределенности, а именно «оптимальность по Парето», было введено итальянским экономистом и социологом Вильфредо Парето в 1896 г. [4]. Возможно, самый большой вклад в развитие теории принятия решений в условиях неопределенности внес А. Вальд [5]. Отметим также исследования К. Д. Льюиса, Г. Райфа [6].

Постановка задачи. Как известно [7], исходными источниками неопределенности являются : неполнота знаний, невозможность точного и полного учета реакции внешней среды; неточное понимание лицом, принимающим решение (ЛПР) целей системы и другие. Отсюда видно, что понятие «неопределенность» информации является более широким, интегрирующим по отношению к понятию «точность» данных.

В настоящее время отсутствует универсальная, инвариантная виду неопределенности, метрика измерения «неопределенности», поэтому при принятии решений используются специализированные, проблемно_ ориентированные метрики. Именно это обстоятельство предопределило разделение задач принятия решений в условиях неопределенности на три основные класса:

- стохастическую неопределенность, которая характеризуется законом распределения вероятностей и статистическими параметрами – математическим ожиданием, дисперсией, средним квадратичным отклонением и другими;
- нечеткую неопределенность, описываемую видом и количественными характеристиками функции принадлежности некоторому нечеткому множеству;
- интервальную неопределенность, характеризуемую границами интервала, внутри которого находится значение переменной.

Общим для всех перечисленных форм является то, что они обязательно характеризуются интервалами возможных значений переменных, а отличие состоит в способе задания предпочтительности возможных значений переменной внутри интервала.

Проблема возможности взаимной трансформации неопределенных величин, заданных различными способами и приведения их к одной форме является крайне актуальной, так как является необходимым условием формализации процедуры принятия решений в условиях многокритериальности и неопределенности исходной информации.

Изложение основного материала. Рассмотрим возможность взаимной трансформации неопределенных величин, заданных различными способами и приведения их к одной форме.

Проанализируем сначала интервалы возможных значений неопреде-

ленных переменных. Во всех случаях прямо или косвенно заданы границы интервалов. Для нечетких чисел и интервальных величин они заданы непосредственно в виде $\langle X \rangle = [x_1, x_2]$, где x_1, x_2 – минимальное и максимальное значения на числовой оси. Для переменных заданных в вероятностном виде границы интервала возможных значений определяются опосредовано через дисперсию $D(X)$, которую легко и однозначно можно трансформировать в интервал возможных значений с помощью среднего квадратического отклонения $\sigma(X) = \sqrt{D(X)}$. Так, например, для нормального закона распределения вероятностей [8]

$$\langle X \rangle = [x_1, x_2] \cong [M(X) \pm 3\sigma(X)].$$

Для закона равномерного распределения вероятностей с характеристиками

$$M(X) = \frac{x_1 + x_2}{2};\ D(X) = \frac{(x_2 - x_1)^2}{12};\ \sigma(X) = \frac{x_2 - x_1}{2\sqrt{3}}, \qquad (1)$$

границы интервала определяются следующим образом [8]:

$$\langle X \rangle = [x_1, x_2] \cong [M(X) \pm \sqrt{3}\sigma(X)].$$

Сравним правила выполнения бинарных арифметических операций над интервальными величинами с различной формой представления неопределенности. При этом учтем, что многофакторная оценка в форме полинома Колмогорова_Габора в расширенном пространстве переменных может быть представлена как линейная интервальная функция вида [9]

$$\langle P(x) \rangle = \sum_{i=1}^{n} \langle a_i \rangle z_i(x), \qquad (2)$$

где $z_i(x)$ – детерминированные факторы, характеризующие альтернативы $x \in X$; $\langle a_i \rangle$ – интервально заданные параметры модели.

Рассмотрим только две арифметические операции: умножения и сложения.

Умножение интервальных величин на скаляр определяется формулой

$$\langle Y \rangle = \lambda \langle X \rangle = \lambda \langle C_X, V_X \rangle = \langle \lambda C_X, \lambda V_X \rangle,$$

где C_X – в зависимости от формы задания неопределенности является математическим ожиданием $M(X)$, центром интервала c_X или нечеткой числовой точкой (модой) нечеткого множества; V_X – соответственно среднеквадратическое отклонение $\sigma(X)$ или радиус интервала для интервальных и нечетких величин ν_X.

Независимо от вида неопределенности сумму двух интервальных величин можно определить следующим образом

$$\langle X \rangle + \langle Y \rangle = \langle C_X, V_X \rangle + \langle C_Y, V_Y \rangle = \langle (C_X + C_Y), (V_X + V_Y) \rangle. \quad (3)$$

В частном случае, сумму интервальной переменной и некоторой детерминированной величины λ можно определить по аналогии с (3) исходя из формулы

$$\langle X \rangle + \lambda = \langle C_X, V_X \rangle + \lambda = \langle (C_X + \lambda), V_X \rangle.$$

Отметим, что операции умножения интервальных величин на скаляр, а так же их суммирование для всех типов неопределенностей определяются аналогично.

Подчеркнем, что знание значений интервала неопределенности и его характеристик таких, как математическое ожидание, нечеткая числовая точка, а также радиус интервала, дисперсия или границы интервала возможных значений нечеткого числа позволяет определить соответственно функцию распределения и функцию принадлежности нечеткого множества.

Другой важный результат рассмотрения арифметических операций на множестве интервальных переменных заключается в том, что сложение интервальных величин со скаляром и умножение их на скаляр не приводит к изменению вида функции, описывающей предпочтения внутри интервала, т. е. функции распределения и функции принадлежности нечеткому множеству.

При суммировании интервальных случайных величин функция распределения результирующей переменной очень быстро эволюционирует к

нормальному закону распределения независимо от закона распределения слагаемых [8], а функция принадлежности суммы унимодальных треугольных функций принадлежности остается унимодальной треугольной и полностью определяется, как видно из формулы (4), положением нечеткой числовой точки (модой нечеткого числа) и границами интервала возможных значений [10]:

$$\mu_C(z)=\begin{cases}1-\dfrac{c-z}{\alpha}, & z<c;\\ 1, & z=c;\\ 1-\dfrac{z-c}{\beta}, & z>c,\end{cases} \qquad (4)$$

где c – мода, результирующего НЧ C; $z\in[z_1,z_2]$ – интервал его определения; $\alpha>0$, $\beta>0$ – соответственно левый и правый коэффициенты нечеткости.

Эти коэффициенты равны:

$$\alpha=c-z_1;\ \beta=z_2-c,$$

где z_1, z_2 – левая и правая границы интервала возможных значений z.

Таким образом, интервальная многофакторная оценка (2), независимо от вида неопределенности будет иметь вид

$$\left\langle C_{P(x)},V_{P(x)}\right\rangle=\sum_{i=1}^{n}\left\langle C_{a_i},V_{a_i}\right\rangle\cdot z_i(x),$$

где $C_{P(x)}$, $V_{P(x)}$ – соответственно центр и радиус интервала неопределенности.

Вид неопределенности в каждой конкретной ситуации определяется в зависимости от особенностей исходной информации.

В некоторых проблемно -ориентированных ситуациях не только параметры модели многофакторного оценивания, но и все или часть факторов $z_i(x)$, характеризующих альтернативы $x\in X$, являются интервально - неопределенными величинами. В этом случае интервальное значение многофакторной оценки альтернативы $x\in X$ вычисляется по формуле

$$\langle P(x)\rangle = \sum_{i=1}^{n} \langle a_i \rangle \cdot \langle z_i(x)\rangle,$$

где $\langle a_i \rangle$, $\langle z_i(x)\rangle$ – соответственно интервальные значения параметров и переменных модели многофакторного оценивания.

При этом, вид правила умножения зависит от типа неопределенности. Формулы вычисления интервалов совпадают только для интервальных, и нечетких величин, но не совпадают с правилом умножения случайных величин.

Рассмотренные выше арифметические операции над интервальными величинами можно проводить только с переменными, которые имеют одинаковый тип неопределенности. Это означает, что нельзя, например, складывать или умножать нечеткую переменную со случайной или интервальной переменной.

Вместе с этим, в зависимости от ситуации может оказаться, что информация, которая используется при моделировании, например для вычисления многофакторной оценки, поступает из различных источников и имеет различный тип неопределенности. В связи с этим возникает необходимость взаимной трансформации интервальных величин с различными видами неопределенности с целью приведения их к одному типу.

С этой точки зрения самыми толерантными являются величины, заданные в интервальной форме, при отсутствии какой-либо информации о предпочтительности значений переменной внутри интервала. Они относятся к классу «объективной неопределенности». По степени информативности это самый «бедный» класс и именно это обстоятельство дает возможность компенсировать отсутствующую объективную информацию эвристической, трансформируя объективную неопределенность в один из типов субъективной неопределенности – в субъективный риск или субъективную неопределенность, т. е. в форму необходимую для анализа.

Так как внутри интервала отсутствует информация о предпочтительности значений переменной и ни одному из значений нельзя отдать предпочтение, это обстоятельство может быть интерпретировано как равная вероятность появления любого значения внутри интервала. Это означает, что объективная неопределенность может рассматриваться как субъективная случайная величина X, распределенная по закону равной вероятности на интервале $[x_1, x_2]$ со статистическими характеристиками

$M(X)$, $D(X)$ $\sigma(X)$, которые определяются по формулам (1) и плотностью распределения $f(X) = \dfrac{1}{x_2 - x_1}$.

Перечисленных статистических параметров достаточно, чтобы производить с переменной арифметические действия как со случайной величиной.

С другой стороны, при необходимости, объективная интервальная неопределенность может быть интерпретирована как нечеткое число соответствующее лингвистической переменной «лежит в интервале от x_1 до x_2» с функцией принадлежности

$$\mu(x) = \begin{cases} 0, & x < x_1; \\ 1, & x \in [x_1, x_2]; \\ 0, & x > x_2. \end{cases}$$

По определению, этих данных так же достаточно для проведения с этим нечетким числом арифметических операций.

Обратная трансформация, т. е. приведение случайных и нечетких переменных к интервальному виду не представляет затруднений, но при такой трансформации теряется важная информация о статистических характеристиках случайной величины таких, как математическое ожидание, дисперсия, функция распределения, а для нечетких чисел – их функции принадлежности.

Рассмотрим возможность трансформации случайной величины в нечеткое число. Как показано в [8], определение интервала возможных значений переменной на основе среднего квадратического отклонения $\sigma(X)$ не вызывает затруднений. Достаточно корректно оценку математического ожидания случайной величины $M(X)$ можно интерпретировать как нечеткую числовую точку. Вместе с этим трансформировать функцию плотности распределения вероятности случайной величины в функцию принадлежности нечеткому множеству, как это иногда делают, принципиально невозможно, так как они имеют различный смысл – первая характеризует частоту появления конкретного значения случайной величины, а вторая – степень истинности утверждения, что некоторое значение переменной удовлетворяет

нечеткому высказыванию (лингвистической переменной). Поэтому функцию принадлежности нечеткой переменной приходится формировать эвристически, на основе учета специфики предметной области, но в большинстве случаев целесообразно выбирать унимодальную треугольную форму (4).

Трансформация нечетких переменных в вероятностную форму также связана с необходимостью эвристического восполнения недостающей информации о функции распределения случайной величины. Объективно, дисперсию $D(X)$ можно определить на основе интервала возможных значений. Нечеткую числовую точку достаточно адекватно можно интерпретировать как оценку математического ожидания $M(X)$. Однако вид функции плотности распределения, в случае её необходимости, приходится назначать на основе эвристических соображений.

Выводы. Таким образом, в случае необходимости, возможно взаимное преобразование интервальных величин с различной формой задания предпочтительности. При этом в большинстве случаев возникает необходимость эвристического восполнения недостающей информации.

Библиографические ссылки

1. **Петров Э.Г.** Методы и средства принятия решений в социально-экономических системах /Э.Г.Петров, М.В.Новожилова, И.В.Гребенник. – К., 2004. – 256 с.
2. История математики с древнейших времен до начала XIX столетия. Т.2. Математика XVII столетия. – М., 1970. – 300 с.
3. **Кунда Н.Т.** Дослідження операцій у транспортних системах /Н.Т.Кунда. – К., 2008. – 400 с.
4. **Крючковский В.В.** Проблема принятия решений и многокритериальная оптимизация /В.В.Крючковский // Матеріали дванадцятої міжнародної наукової конференції імені академіка М.Кравчука. – К., 2008. – С.238.
5. **Таха Хэмди А.** Введение в исследование операций /Х.А.Таха /Пер. с англ. – М., 2001. – 912 с.
6. **Льюис К.Д.** Методы прогнозирования экономических показателей /К.Д.Льюис. – М., 1986. – 126 с.
7. **Крючковський В.В.** Проблема прийняття рішень: історичні аспекти та сучасні підходи /В.В.Крючковський// Проблеми інформаційних технологій. – 2008. – № 2. – С.76-85.

8. **Гмурман В.Е.** Теория вероятностей и математическая статистика /В.Е.Гмурман. – М., 2000. – 479 с.
9. **Крючковский В.В.** Проблемы формализации принятия решений /В.В.Крючковский, Э.Г.Петров // Проблеми інформаційних технологій. – 2008. - №1 (3). – С.57-64.
10. **Мациевский С.В.** Нечеткие множества: Учебное пособие /С.В.Мациевский. – Калининград, 2004. – 176 с.

Надійшла до редколегії 20.01.10

УДК 519.8/.25

©**О.О. Кузенков, Т.А. Зайцева**
Дніпропетровський національний університет імені Олеся Гончара

НЕЛІНІЙНА ДИФЕРЕНЦІАЛЬНА МОДЕЛЬ ДИНАМІКИ ЯКІСНИХ ПОКАЗНИКІВ СЕРЕД ОДНОРІДНИХ ОБ'ЄКТІВ

Запропонована математична модель динаміки якісних показників серед однорідних об'єктів. Модель застосована для дослідження динаміки індивідуального професійного потенціалу серед співробітників певної установи.

Предложена математическая модель динамики качественных показателей среди однородных объектов. Модель использована для исследования динамики индивидуального профессионального потенциала среди сотрудников определенного учреждения.

In the work the mathematical model of quality factors dynamic from geteroobjects is proposed. The corresponding differential model which is constructed considered on an example of dynamics of individual professional potential among employees of certain establishment.

Ключові слова : диференціальна модель, тривимірний фазовий портрет, біфуркаційна гіперплощина, індивідуальний професійний потенціал.

Вступ. Математичне моделювання динамічних систем зміни в яких мають еволюційний характер повинно ураховувати особливості предметної галузі, що підлягає дослідженню, та концепцію життєвого циклу системи, що будується. Зростаюча складність проблем, що виникають в різних сферах діяльності людей обумовлює появу різноманітних підходів та методів [1]. Широке використання диференціальних рівнянь в процесі моделювання дозволяє дослідити реальні системи, визначити загально динамічні тенденції їх розвитку [2].

Постановка задачі. Розглянемо математичну модель, динаміки індивідуального професійного потенціалу серед співробітників певної установи у такому вигляді [3; 4]:

$$\dot{x}_i = \left(1 - \frac{1}{K}\sum_{p=1}^{n} x_p\right) \cdot \sum_{j=1}^{n} a_j \cdot A_{ij} \cdot x_j \qquad i = \overline{1,n} \tag{1}$$

де x_i – фактична кількість співробітників, що мають i -й рівень індивідуального професійного потенціалу; K – кількість робочих місць в установі; a_j – коефіцієнт перспективи розвитку професійного потенціалу у співробітників, що мають j -й рівень індивідуального професійного потенціалу; коефіцієнти $A_{ij} \in [0;1]$ – частка j -ой групи співробітників, що найімовірніше скорегують свій рівень індивідуального професійного потенціалу до i -го, будемо називати «коефіцієнтами переходу» [4]. Виходячи з предметної інтерпретації коефіцієнтів переходу маємо обмеження

$$\sum_{i=1}^{n} A_{ij} = 1 . \tag{2}$$

За умови коли незалежні змінні моделі не охоплюють всіх можливих складових системи $\sum_{i=1}^{n} A_{ij} \le 1$. Природно, що динаміка у системі відсутня, коли загальна кількість співробітників дорівнює нулю. Елементи матриці Якобі, диференціальної моделі (1) у початку координат набувають вигляду:

$$J_{ij} = A_{ij} \cdot a_j , \tag{3}$$

а на стаціонарній гіперплощині $\sum_{p=1}^{n} x_p = K$ вигляд

$$J_{ij} = -\frac{1}{K}\sum_{k=1}^{n} a_k \cdot A_{ik} \cdot x_k . \tag{4}$$

Розглянемо систему (1) для випадку трьох рівнів індивідуального професійного потенціалу співробітників (низького, середнього та високого). У такому разі система (1) набуде такого вигляду

$$\dot{x}_i = \left(A_{i1} \cdot a_1 \cdot x_1 + A_{i2} \cdot a_2 \cdot x_2 + A_{i3} \cdot a_3 \cdot x_3\right) \cdot \left(1 - \frac{x_1 + x_2 + x_3}{K}\right), \quad i = 1,2,3 . \tag{5}$$

Для системи (5) загальний вигляд елементів матриці Якобі є таким

$$J_{ij} = A_{ij} \cdot a_j \cdot \left(1 - \frac{x_1 + x_2 + x_3}{K}\right) - \frac{\sum_{k=1}^{3} A_{ik} \cdot a_k \cdot x_k}{K},$$

у випадку, коли $x_1 + x_2 + x_3 = K$ - якобіан дорівнює нулю.

Характеристичний рівняння системи (5) має вигляд [5]

$$\lambda^3 + p\lambda^2 + q\lambda + t = 0.$$

Коефіцієнти p, q, t характеристичного рівняння, представлені у вигляді суми головних мінорів матриці Якобі (першого, другого та третього порядку відповідно). Однією з інваріант системи (5) є дискримінант Кардано його характеристичного рівняння, що за позначеннями систем набуває такого вигляду

$$D = \frac{32p^6}{27} + \frac{4p^4 q}{3} - 4p^3 t - 3p^2 q^2 + 18pqt - 27t^2. \qquad (6)$$

При $D > 0$ проекції фазових траєкторій на всі координатні площини утворюватимуть топологію «вузол». При $D < 0$ дві координати будуть взаємопов'язані між собою утворюючи топологію фокус (центр).

Метод розв'язування та аналіз отриманих результатів. Позначимо x_1, x_2, x_3 – кількість співробітників, що мають високий, середній та низький рівень індивідуального професійного потенціалу відповідно. У системі (5) може виникати три типи не вироджених транскритичних біфуркацій [1]: $Det(J) = 0$, $Tr(J) = 0$ та $D(J) = 0$ (6). На рис.1 -а представлено біфуркаційну гіперповерхнющину загального вигляду $Tr(J) = 0$, що у позначеннях системи набуває вигляду : $\sum_{i=1}^{3} A_{ii} \cdot a_i = 0$. На рис.1 -б представлено біфуркаційну гіперповерхню загального вигляду $Det(J) = 0$.

З огляду на динаміку індивідуального професійного потенціалу серед співробітників державної служби біфуркаційні переходи мають предметне тлумачення. Для випадку, коли $Tr(J) = 0$ - має місце зміна знаку коефіцієнта перспективи розвитку при переході через біфуркаційну гіперповерхню.

а) б) в)

Рис.1. Біфуркаційні гіперповерхні моделі (5): «а» $-\ Tr(J)=0$**, «б» –** $Det(J)=0$**, «в» –** $D(J)=0$

При такій структурі системи динаміка двох груп з різним рівнем індивідуального професійного потенціалу має безпосередню взаємозалежність $\sum_{i=1}^{n} A_{ii} < \frac{n}{2}$ (рис.1_а). Коли $Det(J)=0$ – має місце зміна знаку коефіцієнта перспективи розвитку у співробітників з певним рівнем індивідуального професійного потенціалу (Рис.1_б). На Рис.1_б позначено дві точки Φ_1 та Φ_2, що визначають набір певних значень коефіцієнтів системи (5). При переході через біфуркаційну гіперплощину $Det(J)=0$ (Рис.1_б) змінюється знак коефіцієнта перспективи розвитку (a_2). $D(J)=0$ – має місце така трансформація системи при якій встановлюється або порушується взаємозв'язок між динамікою пари груп співробітників з різним рівнем індивідуального професійного потенціалу (рис.1_в).

Розглянемо динаміку індивідуального професійного потенціалу. При цьому кількість співробітників, що корегують за умовний період часу рівень свого індивідуального професійного потенціалу складає більше половини від загальної кількості ($\sum_{i=1}^{n} A_{ii} < \frac{n}{2}, \sum_{i=1}^{n} A_{ij} > \frac{n}{2}, i \neq j$).При визначені складових матриці коефіцієнтів переходу ми вважали, що елементи головної діагоналі мають бути наближеними до 1 таким чином, щоб їх значення було пропорційно коефіцієнту Індивідуального Професійного Потенціала (P_{IIII}), який представлений у такому вигляді:

$$P_{IIII} = El \cdot \left(1 + \frac{St}{4} + \frac{Ol}{18}\right),$$

де El – коефіцієнт рівня освіти робітника. Згідно з загальноприйнятими рекомендаціями [3] коефіцієнт освіти звичайно приймається 0,15 – для осіб, що мають незакінчену середню освіту, 0,60 – для осіб із середньою освітою, 0,75 – для осіб зі середньо -технічною та незакінченою вищою освітою, 1,00 – для осіб з вищою освітою за фахом. St – досвід роботи зі спеціальності. Ol – вік співробітника. Стаж ділиться на 4, а вік на 18 так як встановлено, що стаж в 4 рази менше, а вік у 18 разів менше впливає на результативність роботи ніж освіта. При цьому за середній вік для чоловіків приймається 38 років, а для жінок – 41. Коефіцієнти $1 - A_{ii}$ тобто ті, що не належать головній діагоналі, розподілені між A_{ij}, $i \neq j$ пропорційно коефіцієнту $\frac{1}{|i-j|}$. Слід підкреслити, що такий алгоритм не завжди задовольняє умові (2), тому він вимагає додаткової процедури нормування. Дослідження моделі проведено з використанням статистичних даних динаміки індивідуального професійного потенціалу серед співробітників державної служби Дніпропетровської області з 1998 по 2008 рік [3, 4, 6]. Серед респондентів були виділені підгрупи співробітників, що різняться між собою за професійними обов'язками та ступенем відповідальності. Таким чином коефіцієнти переходу набувають наступних значень

Таблиця 1

Коефіцієнти переходу A_{ij} системи (5)

$i \backslash j$	1	2	3
1	0,63	0,26	0,23
2	0,26	0,48	0,46
3	0,11	0,26	0,31

Побудуємо фазовий портрет системи (5) для коефіцієнтів переходу табл.1:

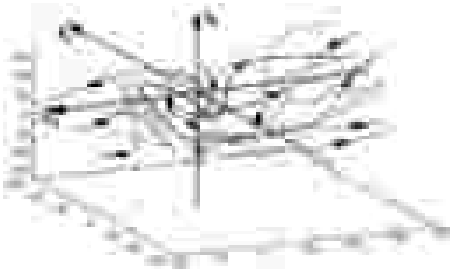


Рис.2. Фазовий портрет системи (5) при $a_1 = -1$, $a_2 = a_3 = 1$,

та коефіцієнтах переходу табл.1

Взаємозалежність x_1 та x_3 обумовлена наявністю пари комплексно спряжених коренів характеристичного рівняння системи (5).

а) **б)**

Рис.3. Проекції фазових траєкторій (рис.2) на площини «а» – $x_1 : 0 : x_3$,

«б» – $x_1 : 0 : x_2$

Поведінка системи в різних площинах є умовно незалежною, так на рис. 3 (б) при проекції фазового портрету на площину $x_1 : 0 : x_2$ утворена топологія «нестійкий дикритичний вузол» [2], та топологія «нестійкий фокус» що утворений при проекції фазових траєкторій на площину $x_1 : 0 : x_3$. За результатами представленими на рис.3(а -б) слід зазначити, що структура динаміки індивідуального професіонального потенціала суттєво залежить від вмінь, навичок та рівня професіональної перспективності співробітників, що є природно і підтверджує адекватність моделі.

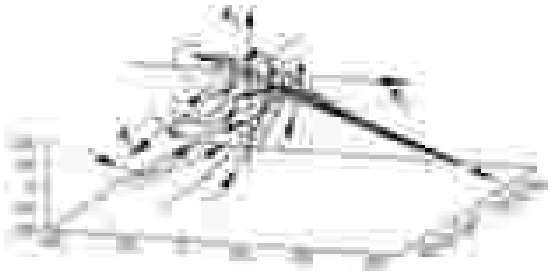


Рис.4. Фазовий портрет системи (5) при $a_1 = a_2 = -1,\ a_3 = 1$**,**

та коефіцієнтах переходу табл.1

На рис.4 представлено фазовий портрет системи (5) з особливою точкою на початку координат. Насамперед це обумовлено зміною знаку інваріанта системи $Tr(J)$ та коефіцієнту перспективності розвитку a_2.

а) б)

Рис.5. Проекції фазових траєкторій (рис.4) на площину «а» – $x_1 : 0 : x_2$**,**

«б» – $x_1 : 0 : x_3$

Розглянемо таку структуру системи коли особлива точка на початку координат є гіперболічною, тобто динаміка всіх груп з відповідним рівнем індивідуального професійного потенціалу не має безпосередньої взаємозалежності між собою. Аналогічна біфуркація $Det(J) = 0$ при умові ($\sum_{i=1}^{n} A_{ii} > \frac{n}{2}, \sum_{i=1}^{n} A_{ij} < \frac{n}{2}, i \neq j$), має іншу біфуркаційну картину.

Таблиця 2

Коефіцієнти переходу A_{ij} системи (5)

$i\backslash\ j$	1	2	3
1	0,70	0,20	0,15
2	0,25	0,70	0,25
3	0,05	0,10	0,60

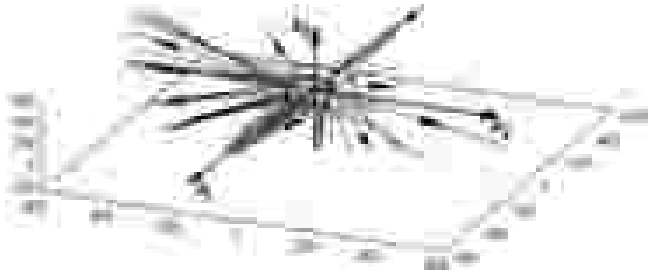


Рис.6. Фазовий портрет системи (5) при $a_2 = -1$, $a_1 = a_3 = 1$, $K = 100$ та коефіцієнтах переходу табл.2

а) б)

Рис.7. Проекція фазового портрету (рис.6) на площину «а» – $x_1 : 0 : x_3$, «б» – $x_1 : 0 : x_2$

Набір коефіцієнтів переходу системи (5) топологія якої зображена на рис.6 відповідає точці M_1 (рис.1_б). Перехід значення визначника матриці Якобі через нуль пов'язан із зміною знаку одного з коефіцієнтів перспективи співробітників з певним рівнем індивідуального професійного потенціалу (такий набір коефіцієнтів відповідає точці M_2). При цьому утворюється картина , що топологічно нееквівалентна топології рис.6.

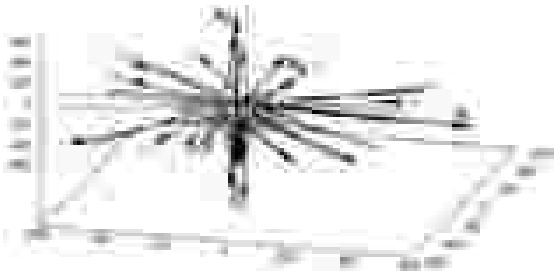


Рис.8. Фазовий портрет системи (5) при $a_1 = a_2 = -1$, $a_3 = 1$, $K = 100$ та коефіцієнтах переходу табл.2

а) б)

Рис.9. Проекція фазового портрету (рис.8) на площину «а» - $x_1 : 0 : x_3$, «б» - $x_1 : 0 : x_2$

Висновки. У роботі розроблена диференціальна модель динаміки груп людей, що однорідні за певною властивістю. Визначені особливі точки та стаціонарні гіперповерхні системи. Досліджені умови виникнення у системі біфуркацій, яким дано прикладне обґрунтування, наведені відповідні тривимірні фазові портрети. Тривимірна модель розглянута на прикладі динаміки індивідуального професійного потенціалу серед співробітників державної служби. Розроблений спеціалізований програмний продукт чисельного розрахунку динаміки індивідуального професійного потенціалу, що дозволяє будувати фазові портрети, біфуркаційні діаграми (біфуркаційні гіперповерхні). Результати дослідження показали адекватність моделі реальним процесам, що мають місце в розглянутій предметній галузі.

Бібліографічні посилання

1. **Магницкий Н.А.** Новые методы хаотической динамики / Н.А. Магницкий С.В. Сидоров – М., 2004. – 320 с.
2. **Эрроусмит Д.** Обыкновенные дифференциальные уравнения// Качественная теория с приложениями./ Д. Эрроусмит, К. Плейс – М., 1986 – 223 с.
3. **Зайцева Т.А.** Модель ефективної діяльності та робота з кадрами у систем і соціального захисту Дніпропетровської області / Т.А. Зайцева, В.В. Васил'єв, К.Ю. Іванова // Соціальна робота в Україні: теорія і практика, №3, – 2003, – С. 82–94.
4. **Кузенков О.О.** Біфуркаційний аналіз диференційної моделі динаміки субпопуляцій / О.О. Кузенков, Т.А. Зайцева // Вісник Запорізького національного університету - З., 2009. С. 25–31
5. **Ван дер Варден Б.Л.** Алгебра./ Б.Л. Ван дер Варден– М., 1976 .648 с.
6. **Зайцева Т.А.** Використання сучасних інформаційних технологій для моделювання соціальних динамічних процесів / Т.А. Зайцева, В.В. Васил'єв, О.С. Футорняк // Матеріали 1 міжнар. науково -практичної конференції «Сучасні проблеми моделювання складних економічних систем» – Кривий Ріг, 2009, – С. 200–201 .

Надійшла до редколегії 15.04.10

УДК 621.3

©А.Я. Кулик, Я.А. Кулик
Вінницький національний технічний університет

МАТЕМАТИЧНА МОДЕЛЬ КАНАЛУ ЗВ'ЯЗКУ

Розроблена математична модель системи передавання як автоматизованої системи управління процесом обміну інформацією у вигляді системи різницевих рівнянь з використанням матриць імовірнісних переходів, які визначають помилки приймання даних, що дозволяє використовувати класичні методи аналізу і синтезу автоматизованих систем управління.

Построена математическая модель системы передачи как автоматизированной системы управления процессом обмена информацией в виде системы разностных уравнений с использованием матриц вероятностных переходов, которые определяют ошибки приема данных, что позволяет использовать классические методы анализа и синтеза автоматизированных систем управления.

The mathematical model of transmission system as an computer-based system of control of process of information interchange in the form of difference equation system using matrixes of probability transitions which define errors of reception of the data that allows to use classical methods of the analysis and synthesis of the automated control systems is constructed.

Ключові слова: автоматизована система, обмін інформацією, матриця, канал зв'язку.

Вступ. Питання управління та побудови алгоритмів оптимізації в різних галузях становлять актуальну проблему, тому постійно розглядаються в наукових роботах [1 ; 2]. Останнім часом з'явилися публікації, які присвячені аналізу можливості використання теорії автоматичного управління для дослідження телекомунікаційних систем [3]. Ряд робіт присвячений розгляду цих питань для телефонних [4], комп'ютерних [5] та радіомереж [6; 7].

Постановка задачі. Фактично основну задачу можна сформулювати як управління процесом передавання інформації каналом зв'язку із вибором параметрів сигналів (потужності, тривалості, структури тощо) з ура-

хуванням впливу завад на процес передавання. Виходячи з цього, для досягнення мети ефективного управління необхідно оцінити відгук на цю дію. Для розв'язання поставленої задачі, систему, в узагальненому вигляді, можна представити, як це подано на рис. 1.

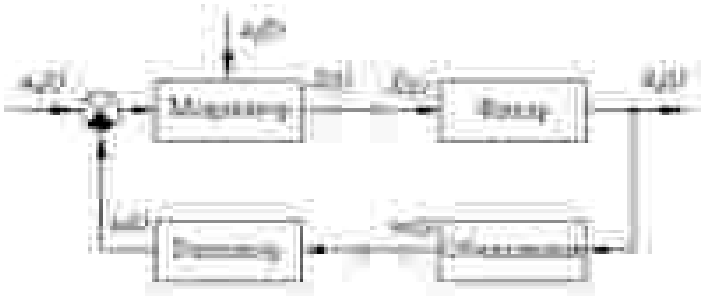


Рис. 1. Узагальнена структура системи передавання інформації

Метод розв'язування. На приймальній частині фільтр виділяє інформаційний сигнал $\hat{a}_i(t)$ і за допомогою обчислювача формує сигнал розузгодження $w(t)$, який, у свою чергу, перетворюється на сигнал управління $u(t)$. Виходячи з цього, задача оптимального управління зводиться до того, щоб сформувати сигнали, які змушували б систему працювати необхідним чином, реагуючи на оцінку якості у відповідності з вибраним критерієм. Задача управління зводиться до оцінки відгуку системи на першому етапі і побудови алгоритму управління з використанням отриманих оцінок – на другому. Такий розподіл задач чітко встановлює пріоритет і показує, що перш ніж розраховувати вплив управління на систему, потрібно оцінити її поведінку. Для цього доцільно використати метод змінних стану.

Оскільки система є дискретною, то її стан визначається лише в дискретні моменти часу, тобто її можна описати системою лінійних рівнянь першого порядку

$$\bar{x}(k+1) = \mathbf{A}(k+1,k)\cdot\bar{x}(k) + \mathbf{\Gamma}(k+1,k)\cdot\bar{\bar{w}}(k) + \Psi(k+1,k)\cdot\bar{u}(k), \qquad (1)$$

$$\bar{z}(k+1) = \mathbf{H}(k+1)\cdot\bar{\bar{x}}(k+1) + \bar{v}(k+1), \qquad (2)$$

де $\bar{x}$ – вектор стану; $\bar{\bar{w}}$ – вектор збурення; $\bar{u}$ – вектор управління; $\bar{z}$ – вектор вимірювання; $\bar{v}$ – вектор похибки вимірювання ; А – перехідна

матриця стану; Γ – перехідна матриця збурення; Ψ – перехідна матриця управління; Η – матриця вимірювання.

З урахуванням рис. 1 легко адаптувати вказані величини до реальних умов. Стан динамічної системи являє собою випадковий процес $x(k)$ з дискретним часом k = 0, 1, … З урахуванням поставлених задач оцінка полягає у визначенні параметра $x(k)$ у момент часу k, використовуючи отримані послідовності результатів вимірювань $z(1), z(2), \ldots, z(j)$. Якщо оцінку значення $x(k)$ позначити через $x(k/j)$, то її можна визначити як вектор-функцію вимірювання $x(k/j) = \varphi_k\left(z(1),..., z(j)\right)$. Зручно розглядати похибку оцінки, яка визначається співвідношенням $\tilde{x}(k/j) = x(k) - x(k/j)$. Якщо оцінка неправильна ($\tilde{x}(k/j) \neq 0$), установлюється штраф, який визначає функцію втрат $L(\tilde{x}(k/j))$. Якщо критерієм якості оцінки J є середнє значення L, тобто $J(\tilde{x}(k/j)) = E\{L(\tilde{x}(k/j))\}$, то оцінка $\tilde{x}(k/j)$ буде оптимальною, якщо вона мінімізує $J(\tilde{x}(k/j))$.

Для двійкових систем помилки накладаються на сигнал, що передається, таким чином, що взаємодія векторів повідомлення і помилки описується за допомогою операції додавання за модулем 2.

Для вирішення задачі управління станом $\bar{x}$ системи вигляду (2) на певному кінцевому інтервалі часу]0, N] необхідно сформувати послідовність сигналів управління $u(k)$, яка мінімізувала б параметр критерію якості. За цей параметр доцільно прийняти математичне сподівання квадрату змінних стану і управління на фіксованому інтервалі часу [8; 9]

$$J_N = E\left(\sum_{k=1}^{N}\left(\bar{x}'(k)\cdot \mathbf{A}(k)\bar{x}(k) - \bar{u}'(k-1)\cdot \mathbf{B}(k-1)\cdot \bar{u}(k-1)\right)\right) \to \min \ . \qquad (3)$$

Для стохастичних систем необхідно розраховувати математичне сподівання E. Параметр J_N інтерпретують як критерій якості вигляду «помилка системи плюс вплив управління». Перша складова передбачає, що бажаний стан системи нульовий, тобто при збільшенні впливу змінних стану x на параметр J_N збільшується відхилення від нуля, а друга – характеризує інтенсивність управління. Таким чином вираз (3) являє собою баланс між похибкою системи та впливом управління. Відносна вага цих двох доданків визначається вибором матриць $\mathrm{A}_{(k)}$ та $\mathrm{B}_{(k-1)}$.

Дані про стан системи мають вигляд послідовності результатів вимірювань $z(1), z(2), \ldots, z(N-1)$ та математичного сподівання $E(x(0))$ початкового стану $x(0)$. Виходячи з цього, вектор управління в кожний момент часу k можна визначити як вектор-функцію

$$\bar{u}(k) = \mu_k\left(\bar{z}(1), \ldots, \bar{z}(k), E(x(0))\right). \qquad (4)$$

Задачі синтезу оптимальних алгоритмів управління таким класом систем достатньо докладно розглянуті в [9 ; 10]. Але їх суттєвим недоліком є те, що для реалізації потрібна повна інформація про всі властивості. Синтез алгоритмів управління в умовах відсутності частини інформації про статистичні властивості параметрів, які впливають на функціонування системи, розглядаються в [10]. Тут особлива увага приділяється байєсовським принципам управління в замкнених стохастичних та адаптивних системах, де результатом є визначення комбінованих стратегій управління динамічною системою. Виходячи з вищевикладеного, виникає необхідність вирішення задачі побудови оптимальних алгоритмів управління параметрами цифрової системи зв'язку. При цьому необхідно враховувати обмеження, які накладають умови функціонування системи та принципи взаємодії об'єктів обміну інформацією. Це необхідно здійснювати, використовуючи сформульований критерій якості, який описується виразом (3).

Перевагою методу змінних стану є можливість побудови рекурентних алгоритмів оцінювання та управління, що дозволяє врахувати нестаціонарність каналу зв'язку і забезпечити необхідне корегування роботи алгоритму. Математична модель комп'ютерної системи передавання інформації у вигляді системи різницевих рівнянь має вигляд

$$\overline{X}_{k+1}(g) = \mathbf{A}(k+1, k) \cdot \overline{X}_k(g), \qquad (5)$$

$$\bar{y}_{k+1}(g) = \bar{x}_{\Sigma, k+1} \oplus \mathbf{B}(k+1, k) \cdot \bar{\xi}_k(g), \qquad (6)$$

де $\overline{x_\Sigma(g)}$ – вектор повідомлення; $\bar{y}(g)$ – вектор спостереження; $\bar{\xi}(g)$ – вектор адитивної похибки; $\mathbf{A}(k+1, k)$ – перехідна матриця джерела повідомлення; $\mathbf{B}(k+1, k)$ – перехідна матриця джерела помилок каналу зв'язку.

При цьому для кожного моменту часу $k = 0, 1, \ldots, N$ повинні бути відомі щільності розподілу імовірностей векторів повідомлення і помилки (ві-

дповідно $p_{k+1}\left(\overline{x_\Sigma}(g)\right) = \mathbf{\Pi}_\mathbf{A}^T \cdot p_k\left(\overline{x_\Sigma}(g)\right)$ та $p_{k+1}\left(\xi(g)\right) = \mathbf{\Pi}_\mathbf{B}^T \cdot p_k\left(\xi(g)\right)$, де $\mathbf{\Pi_A}$ та $\mathbf{\Pi_B}$ – матриці перехідних імовірностей векторів повідомлень та похибок). Оскільки допустимі комбінації векторів утворюють групу над полем Галуа $GF(2)$ розмірністю n, то поелементне додавання векторів повідомлення та похибки, як це показано вище, здійснюється за модулем два.

Матриця перехідних імовірностей $\mathbf{A}_{(k+1,\,k)}$ визначається складом використовуваного кодового алфавіту. З кожним кроком k вигляд матриць **A** і **B** змінюється відповідно до матриць перехідних імовірностей $\mathbf{\Pi}_\mathbf{A}$ та $\mathbf{\Pi_B}$. Таким чином, рівняння (5) та (6) дозволяють описати систему передавання з урахуванням нестаціонарності каналу зв'язку.

Комп'ютерна система управління має фіксовану структуру кодових комбінацій у відповідності з більшістю протоколів передавання [1 1]. Вектор повідомлення має довжину n_Σ розрядів, які вміщують n_i розрядів інформаційного повідомлення та n_u розрядів слова управління. Таким чином формуються вектори:

інформаційний $\overline{x_i} : x_{i.1}...x_{i,n_i}\,0_{u.1}...0_{u.n_u}$; (7)

управління $\overline{x_u} : 0_{i.1}...0_{i,n_i}\,x_{u.1}...x_{u.n_u}$; (8)

сумарний $\overline{x_\Sigma} : x_{i.1}...x_{i,n_i}\,x_{u.1}...x_{u.n_u}$. (9)

Вектор управління $\overline{x}_{u.k+1}(g)$ і, відповідно, перехідна матриця управління $\mathbf{\Gamma}_{(k+1,\,k)}$ формуються згідно алгоритмів управління. З урахуванням того, що у системі використовується декілька параметрів управління, формат вектора управління модифікується з використанням двох шаблонів: ідентифікатора параметра регулювання $\overline{x}_{p.j}$ та його значення $\overline{x}_{z.j}$ з кількістю розрядів n_p та n_z відповідно:

$$\overline{x_u} : 0_{i.1}...0_{i,n_i}\,x_{p.1}...x_{p.n_p}\,x_{z.p1.1}...x_{z.p1.n_z}...x_{z.n_p1}...x_{z.n_p.n_z}. \quad (10)$$

$$\overline{x_\Sigma} : x_{i.1}...x_{i,n_i}\,x_{p.1}...x_{p.n_p}\,x_{z.p1.1}...x_{z.p1.n_z}...x_{z.n_p1}...x_{z.n_p.n_z}. \quad (11)$$

Тоді рівняння (5) можна привести до вигляду

$$\overline{x}_{k+1}(g) = \mathbf{A}(k+1,\,k) \cdot \overline{x}_k(g) \oplus \mathbf{\Gamma}_p(k+1,\,k) \cdot \overline{x}_p(g) \oplus \mathbf{\Gamma}_{z.p1}(k+1,\,k) \times$$

$$\times \bar{x}_{z.p1}(g) \oplus \ldots \oplus \mathbf{\Gamma}_{z.pn}(k{+}1,\, k) \cdot \bar{x}_{z.pn}(g), \qquad (12)$$

де $\mathbf{\Gamma}_{p}(k{+}1,\, k)$ – перехідна матриця вектора ідентифікаторів j-того параметра управління; $\mathbf{\Gamma}_{z.pj}(k{+}1,\, k)$ – перехідна матриця вектора значень j-того параметра управління.

Розподіл імовірностей вектора управління $\bar{x}_u(g)$ необхідно знати в кожний момент часу $k = 0, 1, 2, \ldots$

$$p_{k+1}(\bar{x}_u(g)) = \mathbf{\Pi}_u^{T} \cdot p_k(\bar{x}_u(g)). \qquad (13)$$

Структурна схема моделі каналу зв'язку з управлінням подана на рис. 2.

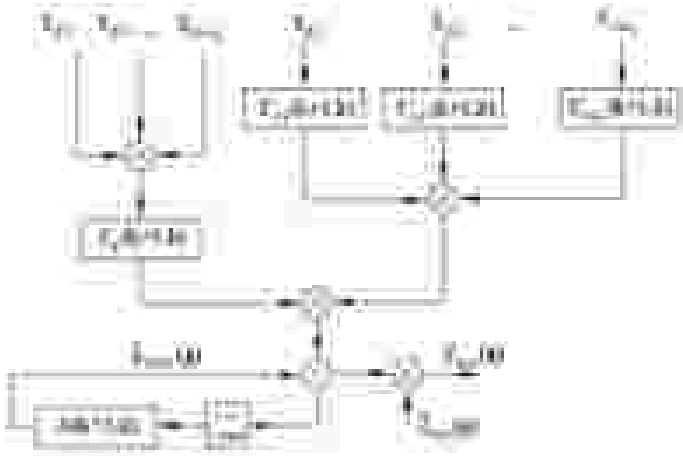


Рис. 2. Структура, яка реалізує модель каналу зв'язку з управлінням

Разом з тим, реальна система передавання інформації працює в напівдуплексному або в дуплексному режимі, що передбачає наявність двох каналів зв'язку – прямого та зворотного, причому приймачі -передавачі здебільшого ієрархічно підпорядковуються один одному (за принципом «головний – другорядний»). При цьому кожен з них буде поміщувати власну інформацію до блока управління. Залежно від особливостей протоколу передавання окремі розряди слова управління можуть не використовуватися, але для подальшого аналізу на даному етапі це враховувати недоцільно. Тоді для прямого (a) і зворотного (b) каналів можна записати системи рівнянь

$$\bar{x}^{a}_{k+1}(g) = \mathbf{A}^{a}(k+1,\,k)\cdot \bar{x}^{a}_{k}(g) \oplus \mathbf{\Gamma}_{p}{}^{b}(k+1,\,k)\cdot \bar{x}^{b}_{p}(g) \oplus \mathbf{\Gamma}^{b}{}_{z.p1}(k+1,\,k)\times$$
$$\times\, \bar{x}^{b}_{z.p1}(g) \oplus \ldots \oplus \mathbf{\Gamma}^{b}{}_{z.np}(k+1,\,k)\cdot \bar{x}^{b}_{z.np}(g) = \mathbf{A}^{a}(k+1,\,k)\cdot \bar{x}^{a}_{k}(g) \oplus$$
$$\oplus\, {}_{u}{}^{b}(k+1,\,k)\times \bar{x}^{b}_{u}(g), \qquad (14)$$

$$\bar{y}^{a}_{k+1}(g) = \bar{x}^{a}_{\Sigma} \oplus \mathbf{B}^{a}(k+1,k)\cdot \bar{\xi}^{a}_{k}(g), \qquad (15)$$

$$\bar{x}^{b}_{k+1}(g) = \mathbf{A}^{b}(k+1,\,k)\cdot \bar{x}^{b}_{k}(g) \oplus \mathbf{\Gamma}_{p}{}^{a}(k+1,\,k)\cdot \bar{x}^{a}_{p}(g) \oplus \mathbf{\Gamma}^{a}{}_{z.p1}(k+1,\,k)\times$$
$$\times\, \bar{x}^{a}_{z.p1}(g) \oplus \ldots \oplus \mathbf{\Gamma}^{a}{}_{z.np}(k+1,\,k)\cdot \bar{x}^{a}_{z.np}(g) = \mathbf{A}^{b}(k+1,\,k)\times$$
$$\times\, \bar{x}^{b}_{k}(g) \oplus \mathbf{\Gamma}_{u}{}^{a}(k+1,\,k)\cdot \bar{x}^{a}_{u}(g), \qquad (16)$$

$$\bar{y}^{b}_{k+1}(g) = \bar{x}^{b}_{\Sigma} \oplus \mathbf{B}^{b}(k+1,k)\cdot \bar{\xi}^{b}_{k}(g), \qquad (17)$$

Структура моделі, що описується вказаними рівняннями, наведена на рис. 3.

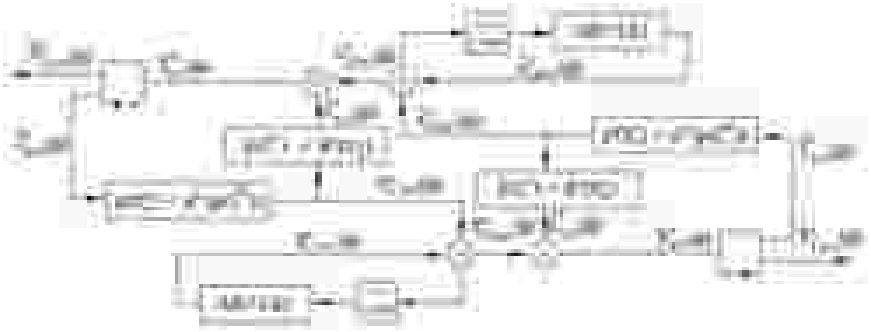


Рис. 3. Структура, яка реалізує модель системи зв'язку з управлінням для напівдуплексного або дуплексного режиму роботи

Аналіз одержаних результатів. Приймач здійснює ідентифікацію сигналів і формує відгук на вплив управління. У відповідності з цим, встановлюються необхідні вектори управління за активованими параметрами регулювання. Відгук на вектори управління описується функцією φ (x_u),

яка являє собою залежність розподілу імовірності вектора помилок від вектора управління.

Указана модель не конкретизує параметри управління та їх кількість, але враховує нестаціонарність каналу та помилки, які виникають внаслідок впливу зовнішніх завад.

Незалежно від природи помилок (адитивні, мультиплікативні, граничні), процес їх виникнення в кодових комбінаціях двійкових систем як для біполярного, так і для уніполярного режимів передавання можна описати у вигляді порозрядного додавання за модулем два

$$\bar{y}_n = \bar{x}_\Sigma \oplus \bar{\xi}_n \,, \tag{18}$$

$$(y_1 y_2 \ldots y_n) = (x_1 x_2 \ldots x_n) \oplus (\xi_1 \xi_2 \ldots \xi_n) \,. \tag{19}$$

Розподіл імовірностей для групи векторів можна визначити за допомогою формули Бернуллі для незалежних помилок, що виникають в каналі зв'язку

$$p_{\xi.B}(j) = \sum_{w_j=1}^{n} C_{n_j}^{w_j} \cdot p_0^{w_j} \cdot (1 - p_0)^{n_j - w_j} \,, \tag{20}$$

де w_j – вага j-тої кодової комбінації; n_j – довжина j-тої кодової комбінації; p_0 – імовірність помилки в елементарному символі.

Для помилок із синдромом групування імовірність виникнення помилок можна описати формулою, запропонованою Л.П. Пуртовим:

$$p_{\xi.\Pi}(j) = \left(\frac{n_j}{w_j} \right)^{1-\alpha_k} p_0 \,, \tag{21}$$

де α_k – показник групування помилок для різних типів каналів.

У формулі (2 1) показник групування помилок α_k можна визначити для конкретного типу каналу лише експериментально. У [12] наведені значення коефіцієнта α_k = (0,32 ... 0,6) для деяких типів каналів. На рис. 4 для порівняння наведені характеристики помилок, розрахованих за вказаними методиками для однакових параметрів передавання. Необхідно відзначити, що байтовий формат передавання даних в асинхронному режимі зменшує вплив синдрому групування.

Залежно від апріорних відомостей щодо умов передавання інформації можна використовувати ту чи іншу методику. Це дозволяє визначити імовірності виникнення помилок для всіх дозволених кодових комбінацій.

Імовірність вектора спостереження $\bar{y}(j)$ для цих кодових комбінацій визначається операцією згортання

$$p_k(\bar{y}(j)) = \sum_{i=1}^{2^n} p_k(\bar{x}_\Sigma(i)) \cdot p_k(\xi(i \oplus j)) \qquad (22)$$

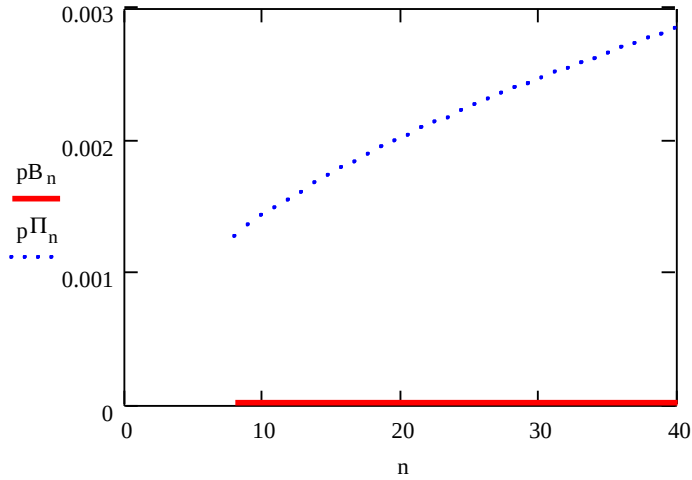


Рис. 4. Розрахунки імовірностей виникнення помилок у кодових комбінаціях для $p_0 = 10^{-3}$ за формулами Бернуллі (pB) та Пуртова (pП)

Умовна імовірність ідентифікації кодової комбінації $\bar{y}(j)$ за умови передавання кодової комбінації $\bar{x}_\Sigma(i)$ визначається імовірністю появи кодової комбінації (i_n) $\oplus$ (j_n). Тоді умовну імовірність вектора спостереження можна визначити як

$$p(\bar{y}(j)/\bar{x}_\Sigma(i)) = p(\bar{\xi}(i \oplus j)). \qquad (23)$$

Повна імовірність появи вектора ідентифікації $\bar{y}(j)$ визначається як згортка розподілів імовірностей кодових комбінацій повідомлення і помилок

$$p(\bar{y}(j)) = p\langle \bar{x}_\Sigma(g), \bar{\xi}(g) \rangle , \qquad (24)$$

яка визначається виразом (21). Кінцевий вираз для визначення умовної імовірності згідно формули Байєса

$$p(\bar{x}_\Sigma(i)/\bar{y}(j))=\frac{p(\bar{x}_\Sigma(j))\cdot p(\bar{\xi}(j\oplus i))}{\sum_{m=1}^{2^n} p(\bar{x}_\Sigma(m))\cdot p(\bar{\xi}(m\oplus j))}, \qquad (25)$$

або з урахуванням структури повідомлення

$$p(\bar{x}_i(j)\oplus\bar{x}_p(i)\oplus\bar{x}_{z.1}(i)\oplus\ldots\oplus\bar{x}_{z.n_p}(i)/\bar{y}(j))=$$
$$=\frac{p(\bar{x}_i(j)\oplus\bar{x}_p(i)\oplus\bar{x}_{z.1}(i)\oplus\ldots\oplus\bar{x}_{z.n_p}(i))\,p(\bar{\xi}(j\oplus i))}{\sum_{m=0}^{2^n-1}\left(p(\bar{x}_i(j)\oplus\bar{x}_p(i)\oplus\bar{x}_{z.1}(i)\oplus\ldots\oplus\bar{x}_{z.n_p}(i))\,p(\bar{\xi}(m\oplus j))\right)}. \qquad (26)$$

Формула (25) може бути використана як для прямого, так і для зворотного каналу, тому ідентифікатор каналу не використовується.

Вектор помилок у каналі зв'язку можна подати у вигляді $\bar{\xi}_{k+1}(g)=(\xi_1\xi_2\ldots\xi_n)$, де $\xi_j = 0$, якщо помилки немає, $\xi_j = 1$, якщо помилка наявна. Тоді вага вектора помилок дорівнює сумі його елементів. Показник якості передавання

$$E_k=\frac{1}{\sum_{k=1}^{K}\sum_{j=1}^{n}\xi_j} \qquad (27)$$

де K – об'єм вибірки для оцінювання стану каналу зв'язку,

характеризує якість передавання даних каналом зв'язку і може змінюватись у діапазоні]0, 1[. Граничні значення цього інтервалу характеризують потенційні параметри каналу – абсолютно зашумлений канал, де всі біти являють собою помилки ($k\to\infty$) та ідеальний канал без помилок.

Повідомлення, що має передаватися до каналу зв'язку $\bar{x}_k$, складається з інформаційної $\bar{x}_i$ та службової $\bar{x}_u$ частин

$$\bar{x}_k : \bar{x}_i \oplus \bar{x}_u \ . \qquad (28)$$

Після виконання завадозахищеного кодування, до кодової комбінації $\bar{x}_k$ додається ще й m контрольних розрядів

$$\bar{x}_n : \bar{x}_k \oplus \bar{x}_m = \bar{x}_i \oplus \bar{x}_u \oplus \bar{x}_m \ . \qquad (29)$$

Такий формат представлення даних дозволяє вносити необхідні корективи до вхідних параметрів розробленої моделі каналу.

Висновки. Розроблена математична модель дозволяє використовувати потужний математичний апарат теорії автоматичного управління для аналізу і синтезу систем передавання дискретної інформації. Залучаючи стандартні алгоритми регулювання, можна будувати адаптивні системи, використовуючи для цього цільову функцію показника якості передавання інформації каналом зв'язку E_k, який залежить від складу вектора сигналу $\bar{x}_n$, що передається, імовірності спотворення елементарного сигналу p_0, довжини блока, що передається, N_b та кількості помилок, що виправляються, t. Таким чином, вказану задачу можна сформулювати у вигляді

$$\begin{cases} E_k = \max\limits_{\bar{x}_n, p_0, N_b, t} E_k(\bar{x}_n, p_0, N_b, t) \\ \upsilon_k \to \upsilon_{пор} \\ p_k \to p_{пор} \end{cases}, \qquad (30)$$

де $\upsilon_k = N_b / T = (8k\upsilon \cdot N_b)/\tau$ – швидкість передавання інформації; p_k – імовірність правильного приймання блоку з N_b байт даних; $p_{пор}$ – порогова імовірність правильного приймання блоку з N_b байт даних; υ_k – швидкість передавання даних каналом зв'язку; $\upsilon_{пор}$ – порогова швидкість передавання даних каналом зв'язку; T – час передавання блоку з N_b байт даних, з урахуванням граничного значення об'єму сигналу $V_k \le V_{пор}$.

Бібліографічні посилання

1. **Калашников А.Е.** Диалоговая система многокритериальной оптимизации технологических процессов: дис. ... канд. техн. наук: 05.13.01 / Калашников Александр Евгеньевич. – М., 2004. – 136 с.

2. **Щипин К.С.** Система прогнозирования на основе многокритериального анализа временных рядов: дис. ... канд. техн. наук: 05.13.10 / Щипин Константин Сергеевич. – М.; 2004. – 134 с.

3. Анализ возможности применения теории автоматического управления для исследования телекоммуникационных систем: тр. 2 междунар. форума «Информационные технологии в XX_ веке» [Електронный ресурс] / Батыр С.С., Суков С.Ф. – Днепропетровск, 27 – 28 апреля 2004. – Режим доступу: http://www.masters.donntu.edu.ua/2004/kita/batyr/library/statya1.htm

4. **Абилов А.В.** Разработка и исследование алгоритмов оценки цифровых сигналов и оптимального использования частотного ресурса в радиотелефонной системе: дис. ... канд. техн. наук: 05.13.16 / Абилов Альберт Винерович. – Ижевск, 2000. – 160 с.

5. **Сысоев С.С.** Рандомизированные алгоритмы стохастической оптимизации и их применение для повышения эффективности работы вычислительных комплексов и сетей: дис. ... канд. физ. -мат. наук: 05.13.11 / Сысоев Сергей Сергеевич. – С.-Пб., 2005. – 80 с.

6. **Елисеев С.Н.** Исследование алгоритмов и устройств цифровой обработки сигналов в системах связи и радиовещания: дис. ... канд. техн. наук: 05.12.13 / Елисеев Сергей Николаевич. – Самара, 2002. – 202 с.

7. **Палей Д.Э.** Исследование динамики дискретных систем фазовой синхронизации второго порядка с нелинейным фильтром: дис. ... канд. техн. наук: 05.12.01 / Палей Дмитрий Эзрович. – Ярославль, 1998. – 188 с.

8. **Медич Дж.** Статистические оптимальные линейные оценки и управление / Медич Дж.; пер. с англ. – М., 1973. – 440 с.

9. **Квакернак Х.** Линейные оптимальные системы управления / Х. Квакернак, Р. Сиван; пер с англ.. – М., 1977. – 650 с.

10. **Кулик А.Я.** Інформаційна оцінка ефективності адаптивної системи передавання / А.Я. Кулик // Інформаційні технології та комп'ютерна інженерія. – 2006. – № 3. – С. 41 – 43.

11. Телекоммуникационные технологии [Електронний ресурс] / Ю.А. Семёнов. – Режим доступу: http://book.itep.ru/1/intro1.htm

12. **Кузьмин И.В.** Кодирование и декодирование в информационных системах / И.В. Кузьмин, В.И. Ключко, В.А. Литвин. – К., 1985. – 190 с.

Надійшла до редколегії 13.10.10

УДК 004.415(045)

©**О.І. Марченко, В.А. Хоменко**
Національний авіаційний університет

КРИТЕРІЇ ВИБОРУ МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вибір моделі життєвого циклу програмного забезпечення є багатокритеріальною задачею, якість вирішення якої зараз визначається досвідом та інтуїцією керівників проекту. Досліджено один з етапів формалізації такого вибору – визначення критеріїв прийняття рішення.

Выбор модели жизненного цикла програмного обеспечения является многокритериальной задачей, качество решения которой сейчас определяется опытом и интуицией руководителей проекта. Исследован один из этапов формализации такого выбора – определение критериев принятия решения.

The choice of model of іщаецфку life cycle is a multicriterion problem which quality of the solution is now defined by experience and intuition of project managers. Article is devoted to investigation of one of phases on a way of formalisation of such choice – to definition of criteria of decision-making.

Ключові слова: модель життєвого циклу, програмне забезпечення, інженерія програмного забезпечення.

Вступ. Процеси життєвого циклу програмного забезпечення визначаються відповідно до визначеної моделі – схеми, що відображає узагальнені процеси, дії і завдання, які залучаються до розробки, експлуатації і супроводу програмного забезпечення, починаючи з визначення вимог і закінчуючи виведенням з експлуатації [1]. На теперішній час відома певна множина моделей життєвого циклу програмного забезпечення, як чисто теоретичних, так і вживаних на практиці [2]. Головні цілі моделювання життєвого циклу полягають у тому, щоб, абстрагуючись від деталей, визначити склад і порядок виконання фаз, їх взаємозв'язок та принципи переходу від однієї фази до іншої.

Модель життєвого циклу, визначаючи загальну схему виконання процесів, розподіл ресурсів і створення продуктів під час виконання проекту розробки програмного забезпечення, впливає, таким чином, на ключові

характеристики процесів. Метою вибору моделі життєвого циклу програмного забезпечення є мінімізація ризиків, витрат і термінів проекту при забезпеченні заданого рівня якості програмного забезпечення.

Задача вибору моделі життєвого циклу програмного забезпечення в даний час вирішується менеджерами проекту на основі особистого досвіду і елементарних міркувань, що визначає високу суб'єктивність і низьку достовірність такого рішення. Формалізовані підходи і методики для вибору моделей практично відсутні, тому їх розробка і дослідження є актуальним завданням.

Постановка задачі. Для вирішення задачі вибору моделі життєвого циклу програмного забезпечення можна використовувати адаптовані відомі методи прийняття рішень [3], які включають наступні етапи:

- постановку мети;
- вивчення проблеми (отримання завдання, усвідомлення проблеми, оцінка можливості виконання відповідно з наявними ресурсами та умовами);
- вибір і обґрунтування критеріїв результативності і наслідків прийнятого рішення;
- обговорення з фахівцями різних варіантів вирішення проблеми;
- вибір і формулювання оптимального рішення;
- прийняття рішення;
- конкретизацію рішення для його виконавців.

Для формування критеріїв вибору моделей життєвого циклу програмного забезпечення проведемо огляд підмножини основних існуючих моделей, їх групування та дослідження характеристик моделей, які пропонуються доступними джерелами.

Моделі та їх групи. На цей час пропонується декілька класифікацій моделей життєвого циклу, жодна з яких не може вважатися досконалою з точки зору охоплення та деталізації. Відповідно до [4] можна виділити три групи моделей життєвого циклу:

- послідовні – модифікації каскадної моделі орієнтовані на розробку «з нуля». До цієї групи входять класична каскадна, спіральна, інкрементна, покрокова моделі; модель швидкої розробки; модель прототипію-

вання і еволюційна. Ця група представляє найбільш використовувані моделі життєвого циклу;

- засновані на багатократному і повторному використанні. Сюди входять моделі компонентної розробки і моделі, засновані на повторному використанні. Вони використовуються при реінженерії успадкованого програмного забезпечення;

- автоматичного синтезу програмного забезпечення. Моделі цієї групи засновані на усуненні фаз життєвого циклу шляхом автоматизації відповідних процесів та використовуються для створення програмного забезпечення в обмежених класах добре формалізованих задач.

Таке групування дає одну з важливих характеристик проекту – спосіб розробки програмного забезпечення. У статті досліджуються характеристики тільки найбільш використовуваних моделей, які відносяться до групи послідовних.

Показники проекту. Для формування системи критеріїв реалізується аналіз множини умов проекту, експертна оцінка їх впливу на досягнення цілей, ранжирування, квантифікація і вибір.

Умови і обмеження проекту можливо розділити на зовнішні (обумовлюються замовником, законодавством і тому подібне) і внутрішні (визначаються можливостями розробника з виконання конкретних задач проекту). До зовнішніх умов відносяться вартість, терміни виконання, способи спілкування із замовником, ступінь визначеності вимог, рівень ризиків, рівень вимог до якості проекту; до внутрішніх – рівень кваліфікації персоналу, рівень засобів та інструментів розробки, оперативні і резервні ресурси розробника, рівень досконалості процесів розробки і тому подібне [5].

Для визначення умов проекту, які можуть бути критеріями вибору моделей життєвого циклу програмного забезпечення, проаналізовано джерела [5 – 10] та виділені характеристики проектів за множиною певних показників.

Критеріями обрано такі показники проекту, як розмір, складність, визначення вимог, оцінка ризиків, контроль за якістю розробки, контроль за виконанням, участь замовника (таблиця).

Характеристики моделей. Історично першими моделями були «каскадна» (характерна для періоду 1970–1985 років) та «спіральна» (характерна для періоду після 1986 р.).

Каскадна модель [6] використовується при достатньо точно і повно сформульованих на самому початку вимогах до програмного забезпечення, що проектується. Також вона може використовуватися коли вимоги до якості домінують над вимогами до затрат та графіку виконання. Основним недоліком цього підходу є те, що реальний процес створення системи ніколи повністю не укладається в таку жорстку схему, постійно виникає потреба у поверненні до попередніх етапів і уточненні або перегляді раніше ухвалених рішень. До інших недоліків цієї моделі відносять те, що вона не враховує ітерації між фазами, а також дії, направлені на аналіз ризиків; важко розробляються паралельні дії.

V-образна модель [7], засновуючись на каскадній та успадковуючи всі її недоліки, приділяє особливе значення плануванню, спрямованому на верифікацію та атестацію продукту на ранніх стадіях його розробки; спрощення (у порівнянні з каскадною моделлю) відстежування ходу процесу розробки, можливість більш реального використання графіка проекту.

Спіральна модель, відображаючи ітеративний характер створення програмного забезпечення дозволяє користувачам бачити систему на ранніх етапах, забезпечує гнучке проектування, дозволяє розбивати проект на невеликі частини [8]. На етапах аналізу і проектування реалізованість технічних рішень і ступінь задоволення потреб замовника перевіряється шляхом створення прототипів. Кожен виток спіралі відповідає створенню працездатного фрагмента або версії системи. Це дозволяє уточнити вимоги, цілі і характеристики проекту, визначити якість розробки, спланувати роботи наступного витка спіралі. Таким чином заглиблюються і послідовно конкретизуються деталі проекту і в результаті вибирається обґрунтований варіант, який задовольняє дійсним вимогам замовника і доводиться до реалізації. Особлива увага приділяється ризикам. До недоліків моделі відносять різкий ріст вартості використання моделі при розробці невеликого проекту; необхідність проведення серйозної оцінки ризиків; зростання документації при великії кількості стадій; необхідність визначення моменту переходу до наступного етапу.

В інкрементній моделі [9] результатом виконання кожного інкременту розробки є функціональний продукт. Кожна подальша версія системи додає до попередньої певні функціональні можливості, до тих пір, доки не будуть реалізовані всі вимоги до системи. Це дає зниження ризиків та витрат на первинне постачання продукту, прискорення початкового

Таблиця

Характеристики моделей за показниками

№	Показник проекту	Характеристики моделей				
		Каскадна	**V-образна**	**Спіральна**	**Інкрементна**	**Еволюційна**
1.	Розмір	великій	від невеликого до великого	великій	від невеликого до великого	великій
2.	Складність	складний з дуже високими вимогами	складний	складний з дуже високими вимогами	складний з дуже високими вимогами	складний
3.	Визначеність вимог	добре визначені перед розробкою	добре визначені перед розробкою	можуть змінюватися у процесі розробки	можуть бути частково визначені	часткова специфікація, уточнюються впродовж усього проекту
4.	Оцінка ризиків	не передбачена	не передбачена	включені аналіз і управління ризиками	зниження ризиків за рахунок контролю над дрібними інкрементами	включені аналіз і управління ризиками

Закінчення таблиці

№	Показник проекту	Характеристики моделей				
		Каскадна	**V-образна**	**Спіральна**	**Інкрементна**	**Еволюційна**
5.	Контроль за якістю розробки	тільки на етапі тестування	Забезпечується атестацією та верифікацією всіх проміжних результатів	Забезпечується валідацією та верифікацією прототипів та тестуванням кінцевого продукту	в кінці кожного інкремента	при випуску кожної версії (еволюції)
6.	Контроль за виконанням	жорсткий впродовж всього проекту	на протязі всього проекту, завершення кожної фази є контрольною точкою	в кінці кожної ітерації	жорсткий	слабкий, велика вірогідність затягування процесу розробки
7.	Участь замовника	не бере участь	не бере участь	бере участь, включаючи аналіз ризиків	ознайомлюється з кожною версією	бере участь протягом всього процесу розробки

графіка поставки. До недоліків моделі відносять відсутність ітерацій у рамках кожного інкремента та необхідність дуже складного та чіткого планування та проектування.

Еволюційна модель [4] застосовується, коли проміжні версії програмного забезпечення можуть експлуатуватися замовником. Після дороблення продукту розробник передає замовнику більш досконалі версії, доки не буде побудована кінцева версія продукту. Ця модель є розвитком інкрементної моделі та дозволяє наблизити терміни початку експлуатації продукту. Очевидно, що передача версій продукту для використання потребує додаткових зусиль на розгортання та впровадження.

У таблиці надано характеристики обраних моделей життєвого циклу за показниками.

Висновки. Визначена множина показників пропонується як основа для дослідження та розробки формалізованих методів прийняття рішень щодо вибору моделі життєвого циклу. Наступним кроком для вирішення цієї задачі є декомпозиція, квантифікація та ранжирування критеріїв шляхом збору та аналізу емпіричних даних щодо впливу тих чи інших показників на характеристики проектів.

Бібліографічні посилання

1. ISO/IEC 12207:1995 Standard for Information Technology – Software Lifecycle Processes.

2. **Соммервилл И.** Инженерия программного обеспечения. / И. Соммервилл – М., 2002.

3. **Блюмин С.Л.** Модели и методы принятия решений в условиях неопределённости. / С.Л. Блюмин, И.А. Шуйкова– Липецк, 2001.

4. **Sidorov N.** Software Engineering. – K., 2007.

5. **Boehm B.W.** et al. Software cost estimation with COCOMO II. Prentice Hall PTR. – New Jersey, 2000.

6. **Winston W. Royce.** Managing the development of large software systems: Concepts and techniques. In WESCON Technical Papers, Vol. 14. – Los Angeles, 1970.

7. **Lewin, Marsha D.** Better Software Project Management – A Primer for Success. – New York, 2002.

8. **Barry W. Boehm.** Spiral Development: Experience, Principles, and Refinements. Spiral Experience Workshop, February 9, 2000 / Special Report CMU/SEI-2000-SR-008, July, 2000.

9. **Whitten Neal.** Managing Software Development Project. – New York, 1995.

10. **Belanger, Thomas C.** Choosing a Project Life Cycle / Field Guide to Project Management, edited by David I.Cleland. – New York, 1998. P. 61-73

Надійшла до редколегії 21.01.10

УДК 621.372

©**А.А. Омельченко, А.Н. Полетайкин**
Донецкая академия автомобильного транспорта

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ СОСТАВЛЕНИЯ ОПТИМАЛЬНОГО ПЛАНА РЕАЛИЗАЦИИ КОМПЬЮТЕРНОЙ ТЕХНИКИ

Розглядається задача складання оптимального плану реалізації комп'ютерної техніки та її пристроїв на регіональному ринку. Використовується імітаційне моделювання для генетичного кодування й декодування істот, що враховує особливості психології масового споживача.

Рассматривается задача составления оптимального плана реализации компьютерной техники и ее устройств на региональном рынке. Используется имитационное моделирование для генетического кодирования и кодирования существ, которое учитывает особенности психологии массового потребителя.

The problem of optimal planning of realization of computer equipment and its devices is considered at the regional market. A simulation technique is used for the genetic coding and coding of creatures, which takes into account the features of psychology of mass consumer.

Ключевые слова: генетический алгоритм, компьютерная техника, оптимальный план.

Постановка задач исследования. В условиях современной экономики объем реализации любого товара уже не определяется привычным соотношением количественных оценок спроса и предложения. В настоящее время реализация компьютерной техники (КТ) представляет собой сложный социально_экономический процесс, в основе которого заложено множество факторов неэкономического характера. Системный анализ влияния этих факторов реализации с использованием методов инженерии знаний и экспертных оценок [6] позволил выявить существенные и получить дополнительную информацию для их системного представления, а также для управления и оптимизации указанного процесса в структуре экспертной системы (ЭС). Учитывая слабую структурированность предметной области, совершенно очевидно, что представление факторов должно носить

вероятностный характер. При этом, как известно из теории вероятностей, оценка случайной величины имеет более выразительный смысл в системе хотя бы двух координат xy, где отражен характер неопределенности функциональной зависимости $F=y(x)$. В [6] разработана модель экспертной оценки факторов реализации КТ в формате нечетких множеств, в связи с чем введено понятие нечеткого фактора (НФ). Здесь под зависимостью F понимается проекция НФ на характеристические признаки (ХП) дифференцирования товара. Множество рассмотренных зависимостей используются как входные данные для прогнозирования спроса КТ с использованием методов структурно -параметрической идентификации и композиционного правила вывода.

Следующим этапом является решение задачи определения варианта картины реализации, при фактическом выполнении которой потребность населения в КТ удовлетворялась бы в наибольшей степени. В [3] обосновано применение для этой цели технологии генетических алгоритмов, которая в последнее время все чаще используется для решения задач функциональной оптимизации. Преимущества ГА состоят в нахождении приблизительно оптимальных решений многопараметрических задач за относительно короткое время, и в возможности поиска всего множества экстремумов для многоэкстремальных функций [5].

Актуальной областью в технологии генетических алгоритмов является задача определения фитнес -функции, являющейся неотъемлемой компонентой отбора – важнейшего алгоритма ГА. Фактически, от успешной реализации фитнес-функции зависит эффективность работы ГА. По аналогии с современной эволюционной теорией в ГА могут быть реализованы принципы естественного и искусственного отбора. Критерий естественного отбора задается непрерывной, определенной на некотором промежутке, целевой функцией (ЦФ). Искусственный отбор задается дискретной функцией годности (ФГ) со значениями «годен» или «негоден», и отсеивает заведомо непригодные решения. При этом искусственный отбор может не иметь места, что замедлит сходимость поиска решения, но обеспечит планомерность развития популяции.

Проблемой применения ГА является слабая способность получения принципиально новых решений, лежащих вне области поиска алгоритма. В качестве средства для получения ОДР, в которой также формируется начальная популяция, используется имитационная модель удовлетворения потребности в компьютерной технике [3]. Модель учитывает особенности психики массового потребителя (человеческий фактор) и которая позволя-

ет получать множественную картину реализации КТ. Инструментом фиксирования моделируемых параметров в области допустимых решений (ОДР), определяемой, главным образом, интеллектуальным механизмом параметризации системы, основанным на знаниях экспертов, является аппарат вероятностной математики: используются вероятностные распределения (моделирование непрерывно действующих факторов), марковские цепи (дискретная параметризация объектов и отношений) и матрицы вероятностей переходов (реализация отношений между объектами).

Разработка генетического алгоритма. Рассмотрим классический ГА, работу которого структурно демонстрирует рисунок 1. На первой итерации начальная популяция автоматически переходит в новое поколение, к которому применяется процедура естественного отбора — первая стадия оптимизации, — в результате которой из нового поколения отсеиваются менее приспособленные особи. Лучшие особи, прошедшие естественный отбор, получают право скрещивания и рекомбинации. Прежде всего, происходят турнирные розыгрыши пар для рекомбинации — вторая стадия оптимизации. Далее полученные пары подвергаются кроссинговеру, в результате которого образуются два потомка, с которыми далее происходит изменчивость. В результате получается множество потомственных особей, равное по величине множеству предков. Оба эти множества в совокупности подвергаются третьей стадии оптимизации — процедуре формирования нового поколения, которая на основании фитнес-функции производит отбор особей в новое поколение. Так происходит одна реализация ГА, в котором, как видно из его описания, каждый процесс выполняется также итеративно (на рисунке 1 итеративные процессы обведены двойной рамкой), где количество итераций определяется размерностью обрабатываемого множества особей. Алгоритм включает в себя три этапа оптимизации, каждый из которых опирается на вычисление значения ЦФ и ФГ. Функционирование ГА прекращается по истечении заданного количества реализаций, либо когда среди нового поколения будет получена особь, максимально точно соответствующая реальной картине, чему соответствует минимальное значение ЦФ при истинном значении функции годности.

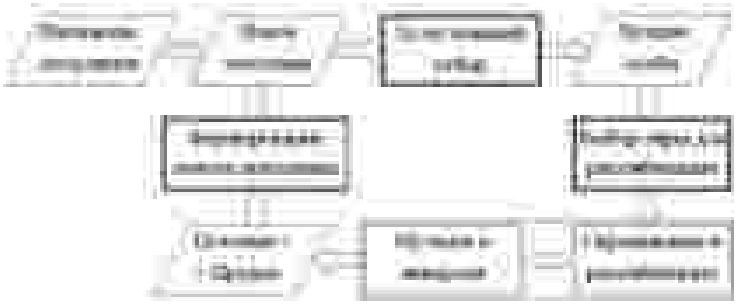


Рис. 1. Итерационный процесс функционирования ГА

Структура хромосомы. В данной задаче оптимального планирования хромосома кодирует развернутый план реализации, структура которого представлена в таблице 1. Кодирование производится в терминах объектно_структурного анализа [1], согласно которому предметная область решаемой задачи оптимизации дезагрегируется на несколько областей знаний – страт (вертикальная сегментация), которые, в свою очередь, иерархически детализируются по горизонтали от уровня проблемы до уровня подзадачи. Данный подход позволяет создавать структурные модели баз знаний любой сложности и степени детализации.

Развернутый план реализации представляет собой в общем виде множество кортежей $R_1\{S_1,...,S_N\}$ … $R_M\{S_1,...,S_N\}$, атрибуты S_i которого отражают параметры процесса реализации. Минимальный набор таких атрибутов: S_1: Код_товара и S_2: Количество_реализации. С целью определения качественных характеристик в множество $\{S\}$ включены дополнительные параметры, отражающие пространственный S_3, временной S_4 и функциональный S_5 аспекты процесса реализации (табл. 1). Параметр S_3: Канал_реализации представляет собой указатель локализации операции в моделируемом процессе. Параметр S_4: Дата_реализации определяет дату выполнения операции в данном временном периоде. Функциональная особенность реализации товара отражает способ, каким был отпущен товар: отдельным устройством или в составе компьютера. Этот аспект процесса задается параметром S_5: Способ_реализации.

Таблица 1

Структура хромосомной матрицы

Ген	S_1	S_2	S_3	S_4	S_5
Слой знаний	ЧТО	СКОЛЬКО	ГДЕ	КОГДА	КАК
Кортеж R_i	**{T}**	**Q(T)**	**{C}**	**{D}**	**{H}**

Алфавит всех параметров, за исключением S_2, есть конечное множество значений, определяемое в случае параметра S_1 из номенклатуры реализуемого товара {T}, для S_3 – из структуры подсистемы реализации товара {C}, а для S_4 – из планируемого периода, определяющего множество {D} возможных дат на шкале времени. Параметр S_5 характеризует способ реализации товара из множества {H} мощностью 2. Параметр S_2, не выражаемый конечным множеством, описывается непрерывным вероятностным распределением Q(T) по экспертно определенному закону.

Таким образом, хромосома включает набор кортежей – подхромосом, состоящих из пяти генов, и имеет матричную структуру. Особи должны иметь одинаковую структуру, в которую включены кортежи для каждого товара, упорядоченные по артикулу, то есть признак $\mathbf{S_1}$ в хромосоме является обязательным. Остальные параметры заданы только для наборов реализуемых товаров в соответствии с кодируемым планом. Популяция таких особей подвергается воздействию разрабатываемого ГА, который ее анализирует и преобразует на основе механизма натуральной эволюции. Ввиду значительной временной сложности оптимизационного алгоритма, $R_{ПОП}$ следует принять равным 100, а количество поколений $N_{ПОК}$ =20.

Естественный отбор. Проведенные исследования показали, что наиболее эффективным методом отбора особей в новую популяцию для данной задачи является метод турнирного отбора с высокой размерностью турнирной группы ($k = 4$) , который предполагает в итоге формирование набора $M(n)$ лучших особей, где n можно положить чётным $0{,}7N$. Такая часть хорошо сочетается с методикой формирования нового поколения, которая, как будет показано далее, приводит к появлению двойной популяции.

Выбор пары для рекомбинации . Существует множество способов образования родительских пар [5], один из которых представляет для задачи оптимального планирования наибольший интерес – это аутбридинг. Сравнение генотипов осуществляется посредством функции аутбридинга $A(X,Y)$, построенной по принципу ЦФ (1). Выбор наиболее различающейся пары можно произвести методом турнирного отбора с $k = 4$.

$$A(X,Y) = \sum_N K_E \left(\frac{par_X - par_Y}{par_T} \right)^2 . \qquad (1)$$

Здесь par_X и par_Y – параметры сравниваемых особей; par_T –заданные отклонения параметров; K_E – коэффициент, определяющийся точностью

кодирования параметра. Уставки отклонений параметров par_T при вычислении функции аутбридинга бально оценены экспертами и представлены в таблице 2. Параметры 2 и 3 нормированы в соответствии с линейным преобразованием в единичный отрезок [0, 1] исходя из статистики, полученной из хромосомы, принимая минимальное значение величин равное нулю (как для идеальной системы реализации товара).

Таблица 2

Уставки отклонений параметров для функции аутбридинга

Код	Параметр	Задание
1	Вероятность отказа	0,15
2	Нормированная длительность обслуживания заявок	0,5
3	Нормированная длительность пребывания товара на складе	0,5
4	Точность предложения товара	0,3

Формирование нового поколения. В [2] для задач, использующих аутбридинг, предлагается идея элитного отбора, основанная на построении новой популяции только из лучших особей репродукционной группы G с наилучшими показателями приспособленности. Надо отметить, что размерность группы G составляет $2n = 1{,}4N$, из чего следует, что $0{,}4N$ особей следует отсеять как заведомо лишние, так как новое поколение должно иметь размерность популяции N, принятую на подготовительном этапе ГА. Таким образом, массив G в первую очередь проверяется на функции годности. Особи, получившие отказ, исключаются из множества G. Если после этого массив G имеет размерность больше, чем N, запускается процедура элитного отбора.

Кроссинговер выполняется с вероятностью $P_{КР.} = 0{,}9$. В [2] показано, что использование случайного чередования одно- и двухточечного кроссинговеров с равномерным распределением по хромосомной матрице понижает вероятность преждевременной сходимости ГА. Предлагаются такие варианты перестановок, опирающиеся на структуру поля знаний: междутоварные (ЧТО-страта), междуканальные (ГДЕ-страта), междудневные (КОГДА-страта), а также произвольный разрыв. Наследование родительских частей для одноточечных операторов выполняется посредством обмена родительскими частями, как показано на рисунке 2, а.

Специализированный оператор рекомбинации используется при скрещивании родителей и работает следующим образом. Рассматриваются гены S_1 каждого из родителей. Если у каждого из них этот ген является пустым, у потомков данный ген также не задействуется. Если же хотя бы один из особей -предков имеет обозначенный ген, то потомки, зависимо друг от друга, будут иметь задействованный ген с N-вероятностью (2), равной корню N-й степени вероятности его включения в хромосомах родителей в количестве N пар, скорректированной на интегральную оценку, полученную на основе приспособленности скрещиваемых особей.

$$P_N = \sqrt[N]{\frac{\sum_{i=1}^{N^S} C_i^S + \sum_{j=1}^{N^O} \left(1 - C_j^O\right)}{N_{CR}}}, \qquad (2)$$

где C_i^S – значение ЦФ i-й особи, включающей данный товар; C_j^O – значение ЦФ j-й особи, не включающей данный товар; N^S – количество особей, включающих данный товар в план; N^O – количество особей, не включающих данный товар в план; $N_{CR} = N^S + N^O = \text{even}(0.7N)$ – количество особей, подвергаемых рекомбинации.

Мутация. Полученные потомки подвергаются мутации с вероятностью $P_{МУТ.} = 0{,}005 \ldots 0{,}01$. В классическом испонении данная операция также одно- или двухточечная по аналогии с операцией кроссинговера и такими же параметрами распределения точек в хромосоме (рис. 2, б). Кроме того, при рассмотрении кроссинговера были предложены два специализированных оператора мутации: 1) с дополнителями и 2) с заменителями.

Инверсия происходит с вероятностью $P_{ИНВ.} = 0{,}01 \ldots 0{,}015$. Механизм такой одно- или двухточечной изменчивости сводится к обмену частями хромосомы, включающей гены $S_2 - S_5$ при фиксированном положении гена S_1, в результате чего образуется качественно иная особь.

а) одноточечный кроссинговер **б) изменчивость**

Рис. 2. Схематическая демонстрация операторов рекомбинации ГА

Параметрическая фитнес-функция. Рассматриваемая задача оптимизации является многопараметрической, где параметры, определяющие план реализации, формируют состав ЦФ:

$$F^* = \sum_N K_E \left(\frac{par_{зад.} - par_{рез.}}{par_{зад.}} \right)^2 \to \min_{par_{рез.}}, \qquad (3)$$

где $par_{рез.}$ – параметры, выделенные из решения ГА; $par_{зад.}$ – заданная величина, определяющая оптимум. Как и для функции аутбридинга, параметры 2 и 3 нормированы в единичный отрезок.

Таблица 3

Уставки параметров оптимизации целевой функции

Код	Параметр	Задание
1	Вероятность отказа	0,001
2	Нормированная длительность обслуживания заявок	0,01
3	Нормированная длительность пребывания товара на складе	0,01
4	Точность предложения товара	0,01

Дискретная функция годности отбраковывает неподходящие особи. Выполняется проверка выборки по параметру S_2 в хромосоме на соответствие заданному закону распределения по критерию согласия Пирсона. Если гипотеза не подтверждается, план оказывается вне области адекватности. Также выборка анализируется по частотному распределению каналов реализации (S_3), по равномерности реализации (S_4) и соответствию прошлым временным рядам, пропорционально относительно способа ре - ализации (S_5). Интегральная оценка адекватности модели образуется посредством алгоритма «И» четырех критериев $C_{S_i}^{\alpha_i}$ типа $\chi^2 \le \chi^2_{n,\alpha}$ в соответствии с экспертными оценками уровня значимости α_i:

$$AD = \sum C_{S_i}^{\alpha_i} = C_{S_2}^{\alpha_2} \cap C_{S_3}^{\alpha_3} \cap C_{S_4}^{\alpha_4} \cap C_{S_5}^{\alpha_5} \qquad (4)$$

Рассмотренные в данном разделе процедуры ГА выполняются последовательно в рамках схемы, показанной на рисунке 1 с циклическим проигрыванием ряда поколений, количество которых определяется фиксировано, либо достижением определенного уровня оптимальности, определяемого исходя из значения параметрической целевой функции. Укрупненный алгоритм работы ГА схематически представлен на рисунке 3.

Исследования и результаты работы генетического алгоритма . Поскольку разработанный ГА встроен в экспертную систему, эффективность его работы целесообразно оценивать в структуре ЭС. С этой целью в [4] предложена концепция подсистемы самооценки системы по 4 независимым оценкам: обобщенной полезности, вероятности достижения цели по интегральному критерию оптимизации, интегральному показателю эффективности и коэффициенту упорядоченности. Результаты оценивания представлены на рис. 4, где под циклом управления подразумевается полный цикл бизнес_процессов планирования реализации КТ.

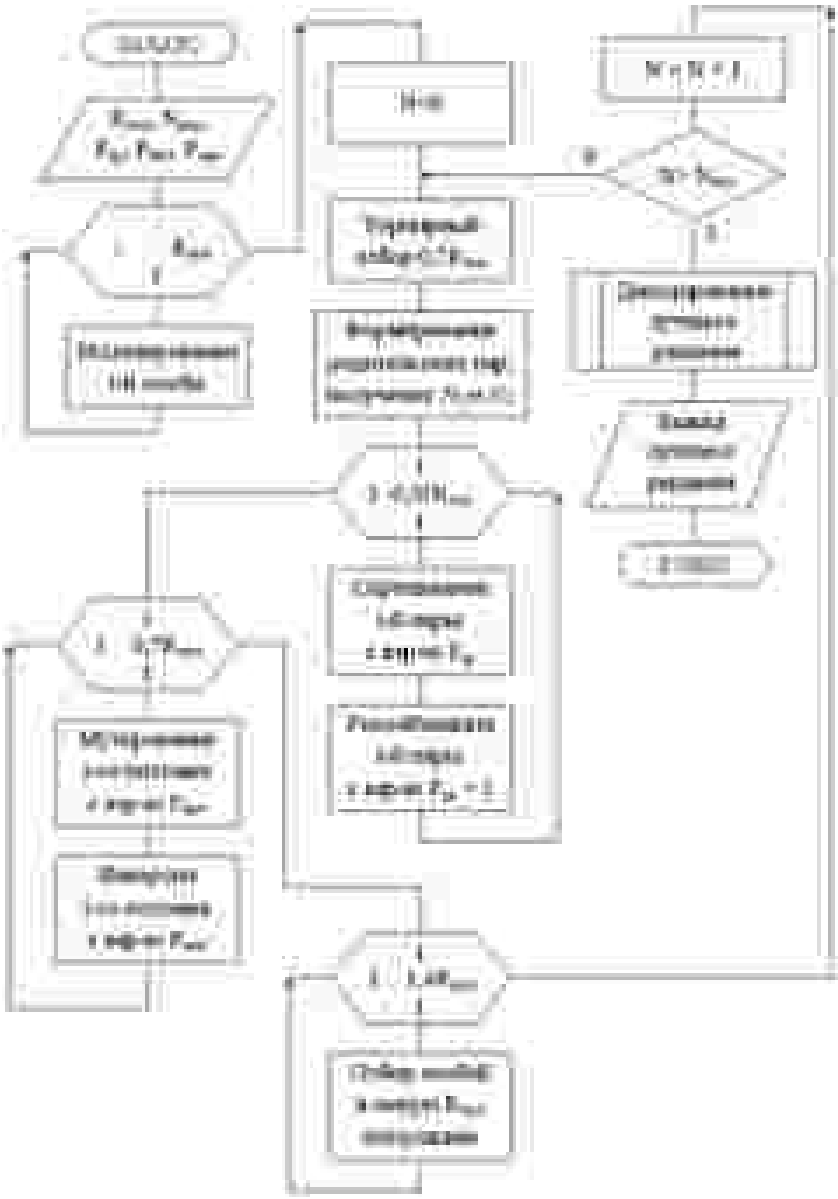


Рис. 3. Схема укрупненного алгоритма работы ГА

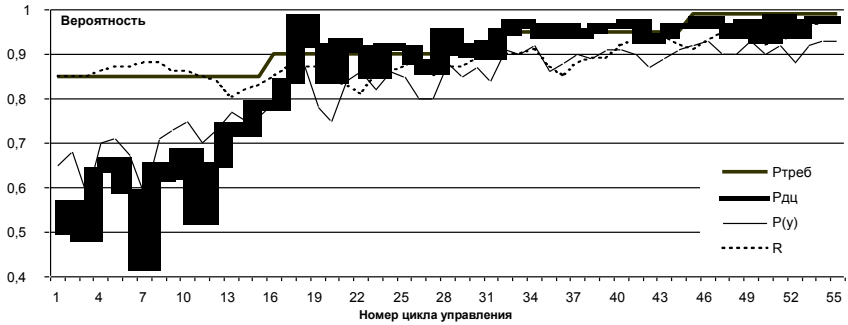


Рис. 4. График изменения эффективности ЭС

Выводы. 1. Предложен новый подход к оптимизации многопараметрических систем, который заключается в совместном использовании аппарата имитационного моделирования и генетических алгоритмов. 2. Разработан генетический алгоритм для получения оптимально составленного плана реализации, который встраивается в структуру экспертной системы поддержки принятия решений при реализации компьютерной техники.

Библиографические ссылки

1. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб., 2001. – 384 с.
2. **Гладков, Л.А.** Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик; под ред. В.М. Курейчика. – [2-е изд., испр. и доп.] – М., 2006. – 320 с.
3. **Полетайкин, А.Н.** Имитационная модель процесса удовлетворения потребности в компьютерной технике / А.Н.Полетайкин // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація, випуск 12 (118). – Донецьк, 2007. – с. 92 – 101.
4. **Полетайкин, А.Н.** Модель оценивания эффективности автоматизированных информационных систем. / А.Н.Полетайкин // Сборник научных трудов НГТУ. – 2009 №3 (53) С.58 – 65.
5. **Скобцов, Ю.А.** Основы эволюционных вычислений. / Ю.А.Скобцов – Учебное пособие. – Донецк, 2008. – 326 с.
6. **Спорыхин, В.Я.** Концепция экспертной системы оценки факторов реализации компьютерной техники. / В.Я.Спорыхин, А.Н.Полетайкин // Вестник Херсонского национального технического университета.− Херсон, 2007. − №4(27). − С.286−292.

Надійшла до редколегії 12.01.10

УДК 539.3

©**А. М. Пасічник** [*], **В. А. Пасічник** [**], **А. М. Світлична** [**]

[*]*Академія митної служби України*

[**]*Дніпропетровський національний університет імені Олеся Гончара*

МЕТОД ПОБУДОВИ РОЗВ'ЯЗКІВ НЕЛІНІЙНИХ КРАЙОВИХ ЗАДАЧ ДИНАМІКИ РОЗПОДІЛЕНИХ СИСТЕМ

Запропоновано метод побудови розв'язку нелінійних задач динаміки розподілених систем із застосуванням методу збурення та апроксимації Паде.

Предложен метод построения решения нелинейных задач динамики распределенных систем с применением метода возмущения и аппроксимации Паде.

The method of constructing the solution of nonlinear dynamics tasks of distributed systems using the perturbation method and Pade approximation is shown.

Ключові слова: метод збурення, нелінійна динаміка, апроксимація Паде.

Вступ. Стрімкий розвиток різних галузей машинобудування передбачає широке застосування в якості конструктивних рішень пластин та стрижневих систем і ставить нові вимоги до міцності та надійності таких конструкцій. Особливо важливий клас задач пов'язаний з урахуванням нелінійного характеру крайових умов. Тому удосконалення методів розрахунку динамічних характеристик пластин та стрижневих систем є достатньо актуальною проблемою.

Проблемі врахування нелінійного характеру крайових умов при дослідженні коливань механічних систем присвячено значну кількість робіт, в яких використовуються різні варіанти методу малого параметра [1; 3; 5] та варіаційні підходи [2 ; 4]. При цьому для малих значень коефіцієнта нелінійності ($\alpha<<1$) задача ефективно розв'язується методом збурень. У випадку суттєво нелінійного характеру крайових умов ($\alpha\sim1$) побудова розв'язку значно ускладнюється і зводиться до розв'язання нескінченої

системи алгебраїчних рівнянь. Для побудови розв'язків якої можливо застосувати метод урізання, що призводить до значних ускладнень навіть при невеликій кількості рівнянь.

У даній роботі для побудови асимптотичного розв'язку задачі подовжніх коливань стрижня використовується «метод малих δ» [8], у відповідності з яким малий параметр вводиться до показника степені нелінійності а потім реалізується процедура асимптотичного інтегрування.

Постановка задачі. Розглянемо застосування «методу малих δ» для побудови періодичного розв'язку модельної нелінійної крайової задачі для рівняння другого порядку

$$\ddot{x} + x^3 = 0, \qquad (1)$$

$$x(0) = 1, \quad \dot{x}(0) = 0. \qquad (2)$$

У відповідності з «методом малих δ» введемо у показник степеня нелінійного члена в рівнянні (1) малий параметр δ таким чином:

$$\ddot{x} + x^{1+2\delta} = 0. \qquad (3)$$

Оскільки параметр $\delta << 1$, то можемо розкласти величину $x^{2\delta}$ в такий ряд

$$x^{2\delta} = 1 + \delta \ln(x)^2 + 0.5\delta^2 \ln(x^2)^2 + \dots . \qquad (4)$$

Розв'язок рівняння (1) подамо у вигляді

$$x = \sum_{k=0}^{\infty} \delta^k x_k \qquad (5)$$

та проведемо заміну змінної часу

$$t = \tau/\omega, \qquad (6)$$

і подамо частоту ω у вигляді розвинення за степенями параметра δ:

$$\omega^2 = 1 + \sum_{i=1}^{\infty} \alpha_i \delta^i. \qquad (7)$$

Тут α_i – невідомі коефіцієнти розкладення, які необхідно визначити в процесі побудови розв'язку.

Підставляючи співвідношення (4) – (7) до рівняння (1) та виконуючи асимптотичне розщеплення за параметром δ отримуємо таку рекурентну послідовність крайових задач:

$$\ddot{x}_0 + x_0 = 0, \qquad (8)$$

$$x_0(0) = 1, \quad \dot{x}_0(0) = 0; \qquad (9)$$

$$\ddot{x}_1 + x_1 = -x_0 \ln(x_0)^2 - x_1 \ddot{x}_0, \qquad (10)$$

$$x_1(0) = \dot{x}_1(0) = 0; \qquad (11)$$

$$\ddot{x}_2 + x_2 = -x_1 \ln(x_0)^2 - 2x_1 - x_0(\ln x_0^2)^2 - x_2\ddot{x}_0 - x_1\ddot{x}_1, \qquad (12)$$

$$x_2(0) = \dot{x}_2(0) = 0. \qquad (13)$$

.....

Розв'язок крайової задачі нульового наближення (8), (9) має вигляд

$$x_0 = \cos t. \qquad (14)$$

З урахуванням розв'язку (14) крайова задача першого наближення запишеться так

$$\ddot{x}_1 + x_1 = -\cos t \ln(\cos^2 t) + x_1 \cos t = L_1. \qquad (15)$$

Для того, щоб розв'язок крайової задачі (15), (11) не мав секулярних членів, необхідно виконання умови

$$\int_0^{\pi/2} L_1 \cos t \, dt = 0. \qquad (16)$$

Із умови отримуємо значення x_1

$$x_1 = 1 - 2\ln 2. \qquad (17)$$

Період коливань при цьому визначається так

$$T = 2\pi[1 + \delta(\ln 2 - 0.5)]. \qquad (18)$$

При $\delta = 1$ значення періоду T=6.807. Оскільки точне значення T′=7.4164, то похибка першого наближення становить 8.2 %.

При цьому, побудова розв'язку крайової задачі (12), (13) дає значення періоду T=7.4111, що практично співпадає з точним значенням.

Розглянемо тепер задачу подовжніх коливань стрижня з урахуванням нелінійності умов його закріплення. Крайова задача в даному випадку запишеться так [6]

$$u_{tt} = u_{xx}, \qquad (19)$$

$$u(0, t) = 0, \qquad (20)$$

$$u_x(l, t) + u(l, t) + u^3(l, t) = 0. \qquad (21)$$

У відповідності з «методом малих δ» введемо до показника степеня нелінійності члена в крайових умовах (21) параметр δ

$$u^3(l, t) = u^{1+2\delta}. \qquad (22)$$

Розв'язок крайової задачі (19) –(21) подамо у вигляді асимптотичного ряду за степенями параметра δ

$$u = \sum_{k=0}^{\infty} \delta^k u_k. \qquad (23)$$

Уведемо також заміну змінної часу

$$t = \tau/\omega, \qquad (24)$$

та подамо частоту ω у вигляді розвинення за степенями параметра δ

$$\omega^2 = \omega_0^2 \,(1 + \sum_{i=1}^{\infty} \alpha_i \,\delta^{\,i}) \cdot \qquad (25)$$

Тут α_i – невідомі коефіцієнти розкладення, які необхідно визначити за процесом побудови розв'язку.

Нелінійний член (22) також подамо у вигляді розкладення:

$$u^3 = u^{1+2\,\delta} = u\,[1 + \delta\,(\ln(u^2)) + \delta/2\,(\ln(u^2))^2 + \ldots\,]. \qquad (26)$$

Згідно з методологією асимптотичного підходу підставимо співвідношення (22) – (26) у вихідну крайову задачу (19) – (21) і виконуючи розщеплення за параметром δ отримаємо таку рекурентну послідовність крайових задач:

$$u_{0tt} = u_{0xx}\,, \qquad (27)$$

$$u_i(0,\,t) = 0, \quad i=1,\,2,\,3,\cdots\,, \qquad (28)$$

$$u_{itt} = u_{ixx} - \sum_{i=0}^{\infty} \alpha_{i(n-p)}\, u_{ptt}\,, \quad i=1,\,2,\,3,\,\cdots\,, \quad \alpha_0=0, \qquad (29)$$

$$\text{при } x = l \quad u_{0x} + 2u_0 = 0, \qquad (30)$$

$$u_{1x} + 2u_1 = -u_0 \ln(u_0^2), \qquad (31)$$

$$u_{2x} + 2u_2 = -u_1 \ln(u_0^2) - 2u_1 - 0.5\,u_0\,[\ln(u_0^2)]^2. \qquad (32)$$

……

Результати дослідження. Побудова розв'язку нульового наближення. Розв'язок крайової задачі (27), (28), (30) в нульовому наближенні покладаємо в такому вигляді

$$u_0 = A \sin(\omega_0 x) \sin(\omega_0 t), \qquad (33)$$

де частота ω_0 визначається із наступного трансцендентного рівняння

$$-\omega_0/(2\pi) = \operatorname{tg} \omega_0. \qquad (34)$$

Розрахунок перших десяти ненульових значень ω_0 наведено в табл. 1.

Таблиця 1

Числові значення параметра ω_0

$\omega_0^{(1)}$	$\omega_0^{(2)}$	$\omega_0^{(3)}$	$\omega_0^{(4)}$	$\omega_0^{(5)}$
2.289	5.087	8.096	11.173	14.276
$\omega_0^{(6)}$	$\omega_0^{(7)}$	$\omega_0^{(8)}$	$\omega_0^{(9)}$	$\omega_0^{(10)}$
17.393	20.518	23.646	26.778	29.912

Розв'язок трансцендентного рівняння (34) при $k \to \infty$ має вигляд

$$\omega_0^{(k)} \to \pi/2\,(2k+1). \qquad (35)$$

Побудова розв'язку першого наближення. З урахуванням отриманого розв'язку нульового наближення крайова задача першого наближення запишеться так

$$u_{1xx} - u_{1tt} = \alpha_1 A\omega_0^2 \sin(\omega_0 x)\sin(\omega_0 t), \qquad (35)$$

$$u_1(0, t) = 0, \qquad (37)$$

при $x = l$

$$u_{1x} + 2u_1 = A\sin\omega_0 \sin(\omega_0 t)\ln(A^2\sin^2\omega_0) -$$

$$- A\sin\omega_0 \sin(\omega_0 t)\ln(\sin^2\omega_0). \qquad (38)$$

Загальний розв'язок крайової задачі першого наближення складається із частинного розв'язку рівняння (36) та розв'язку відповідного однорідного рівняння.

Частинний розв'язок рівняння (36), що задовольняє крайовій умові (37) має такий вигляд

$$u_1^{(1)} = \alpha_1 x A\sin(\omega_0 x)\sin(\omega_0 t). \qquad (39)$$

Із умови компенсації резонансної складової визначаємо

$$\alpha_1 = \frac{\ln(A^2\sin^2\omega_0) + 0.5 - \ln 2}{(6 + \omega_0^2)}. \qquad (40)$$

Праву частину крайової умови (38) представимо так:

при $x = l$

$$u_{1x} + 2u_1 = A_1 \sum_{k=2}^{\infty} T_k \sin(k\omega_o t), \qquad (41)$$

де $A_1 = -A\sin\omega_0$; $T_k = \dfrac{\omega_0}{2\pi}\displaystyle\int_0^{\frac{2\pi}{\omega_0}} \sin(k\omega_0 t)\sin(\omega_0 t)\ln\left(\sin^2(\omega_0 t)\right)dt$.

Розрахунок перших десяти ненульових значень періоду коливань T_k наведено в табл. 2.

Таблиця 2

Числові значення параметра T_k

T_1	T_2	T_3	T_4	T_5
−0.193	0	−0.25	0	−0.083
T_6	T_7	T_8	T_9	T_{10}
0	−0.042	0	−0.025	0

Розв'язок однорідного рівняння запишеться так

$$u_1^{(2)} = A_1 \sum_{k=2}^{\infty} T_k^{(1)} \sin(k\omega_o x)\sin(k\omega_o t) , \qquad (42)$$

де $T_k^{(1)} = \dfrac{T_k}{k\omega_0 \cos(k\omega_0) + 2\sin(k\omega_0)}$.

Остаточно повний розв'язок задачі у першому наближенні матиме такий вигляд

$$u_1 = u_1^{(1)} + u_1^{(2)}. \qquad (43)$$

Побудова розв'язку другого наближення. Для побудови розв'язку другого наближення праву частину крайової умови (32) представимо в такому вигляді:

при $x = l$ $\quad u_{2x} + 2u_2 = 2u_1 - 0.5\ln(u_0^2) \times$

$$\times [2u_1 + u_0 \ln(u_0^2)] = -2u_1 - 0.5\ln(u_0^2)\, u_{1x}. \qquad 44)$$

Виокремимо у правій частині крайової умови (44) резонансну гармоніку:

при $x = l$ $\quad u_{2x} + 2u_2 = -R_2 A \sin\omega_0 \sin(\omega_0 t) + \varphi(t), \qquad (45)$

де $R_2 = 0.5\{x_1[4 + \omega_0(\omega_0 + 2)\ln(A^2 \sin^2\omega_0) + 0.5 - \ln 2] + \beta\}$,

$$\beta = \frac{\omega_0}{2\pi} \int_0^{\frac{2\pi}{\omega_0}} \left[\sin(\omega_0 t) \ln\left(\sin^2(\omega_0 t)\right) \sum_{k=2}^{\infty} T_k^{(1)} \cos(\omega_0 t)\sin(k\omega_0 t) \right] dt ,$$

через $\varphi(t)$ позначена сума резонансних гармонік.

Побудуємо частинний розв'язок рівняння (45), який задовольняє крайовій умові (29) для резонансної правої частини

$$u_2^{(1)} = \alpha_2 A\omega_0^2 x \cos(\omega_0 x)\sin(\omega_0 t) +$$

$$+\alpha_1^2 A\omega_0^3 x^2 \cos(\omega_0 x)\sin(\omega_0 t) - \alpha_1^2 A\omega_0^2 x \sin(\omega_0 x)\sin(\omega_0 t) =$$

$$= A\sin(\omega_0 t)\ \omega_0^2 [\alpha_2 x \cos(\omega_0 x) + \alpha_1^2 x(x\omega_0 \cos(\omega_0 x) - \sin(\omega_0 x))]. \qquad (46)$$

Стала величина α_2 визначається із умови виключення резонансних членів у правій частині крайової умови (46). У результаті отримуємо такий вираз

$$\alpha_2 = \frac{R_2 - \alpha_1^2 \omega_0 (9 + \omega_0^2)}{\omega_0 (6 + \omega_0^2)} . \qquad (47)$$

Таким чином, з точністю до членів третього порядку малості за параметром δ розшукувану частоту коливань можна подати

$$\omega \cong \omega_0 \sqrt{1+\alpha_1\delta+\alpha_2\delta^2}\,. \qquad (48)$$

Застосовуючи апроксимацію Паде до розкладення (48) отримаємо уточнений розв'язок задачі

$$\omega \cong \omega_0 \sqrt{\frac{\alpha_1+(\alpha_1^2-\alpha_2)\delta}{\alpha_1-\alpha_2\delta}}\,. \qquad (49)$$

Висновок. Побудований розв'язок дає задовільні результати для низьких значень частоти коливань. Для випадку високих частот коливань необхідно застосовувати асимптотичне розвинення за величинами оберненими частоті ω. Запропонований метод допускає узагальнення для побудови розв'язків та дослідження більш складних двох - та трьохвимірних систем .

Бібліографічні посилання

1. **Андріанов І.В.** Асимптотичні методи дослідження математичних моделей прикладних задач / І.В. Андріанов, А.М. Пасічник. – Д., 2004.
2. **Голоскоков Е.Г.** Нестационарные колебания деформируемых систем / Е.Г. Голоскоков, А.П. Филиппов. – К., 1977.
3. **Милосердова И.В.** Импульсные волны в одномерной системе с нелинейными границами / И.В. Милосердова, А.И. Потапов, А.А. Новиков // Волны и дифракция. – 1987. –Т.27, №2. – С. 296-301.
4. **Муницын А.И.** Собственные колебания балки с нелинейными опорами / А.И. Муницын //Пробл. машиностроения и надежности машин. – 1998. – №2. – С. 36-39.
5. **Найфэ А.** Введение в методы возмущений. /А. Найфэ – М., 1984.
6. **Каудерер Г.** Нелинейная механика. / Г. Каудерер – М., 1961.
7. **Andrianov I.V.** Asymptotic investigation of the nonlinear dynamics boundary value problem for rods / I.V. Andrianov, V.V. Danishevsky // Technische Mechanik. – 1995. – Vol.15. – №1. – P. 53-55.
8. **Bender C.M.** A new perturbative approach to nonlinear partial differential equations / Bender C.M., Boettcher S., Milton K.M. //J. Math. Phys. – 1991. – Vol.32. – №11. – P. 3031-3038.

Надійшла до редколегії 16.03.10

УДК 519+61:681.3

©**Ю.А. Прокопчук**
Украинский государственный химико-технологический университет,
Институт технической механики НАН Украины и НКА Украины

МОДЕЛЬ ДИНАМИЧЕСКОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СППР

Розглянуто задачу розробки формалізму для опису довільних об'єктів та процесів у природних предметних областях, і задачу визначення основних операторів динамічної інтелектуальної СППР.

Рассмотрена задача разработки формализма для описания произвольных объектов и процессов в естественных предметных областях, и задача определения основных операторов динамической интеллектуальной СППР.

The problem of development of a formalism for the description of any objects and processes in natural subject domains, and a problem of determination of the basic operators of dynamic intellectual DSS is considered.

Ключевые слова: модель, система поддержки принятия решений, формализм.

Для решения интеллектуально сложных задач в таких естественных предметных областях как социология, медицина, экология, криминалистика, образование, МЧС и т.д. разрабатываются специализированные СППР. Как показала практика, они наиболее выигрышны при описании слабоструктурированных систем, характеризующихся многоаспектностью происходящих в них процессов, отсутствием достаточной количественной информации об их динамике, изменчивостью структуры системы и характера процессов во времени. Приходится иметь дело даже с автоматизацией таких когнитивных процессов как интуиция, опыт, ассоциативность мышления, догадка, предвидение [1-7].

Постановка задач исследования. Несмотря на большое количество работ по тематике СППР, остаются актуальными задача разработки прозрачного и эффективного в прикладном плане формализма для описания и моделирования произвольной ситуации действительности, включая ее динамику, и задача описания базовых когнитивных процессов, лежащих в основе динамической интеллектуальной СППР (ДИ СППР).

Описание ситуаций действительности. В [4] произвольная ситуация действительности α определяется следующим образов: $\alpha = \alpha(\{<J_\tau \tau/T, J_t t/\Lambda>\})$, где τ/T – значение элементарного теста τ, выбранное из домена T; t/Λ – значение времени t, выбранное из домена Λ; J – оператор оценки истинности значения теста.

Конструкции $<\tau/T, t/\Lambda>$ и $<\neg\tau/T, t/\Lambda>$ определяют *элементарные события*. Конструкция $<\{\tau/T\}, t/\Lambda>$ определяет *составное событие*, а конструкция $<\{\tau/T\}, \delta/\Lambda>$ – *протяженное событие*, где δ – временной интервал.

Примеры доменов Λ: $\Lambda 1$= ‘Дата: время’; $\Lambda 2$= ‘Дата: {утро, день, вечер, ночь}’; $\Lambda 3$= ‘Дата’; $\Lambda 4$= ‘Месяц, Год’; $\Lambda 5$= ‘Год’. Вполне очевиден способ задания пересчета из одного домена в другой (с большим номером), т. е. определена вычислительная цепочка (ориентированный граф доменов): $\Lambda 1 \to \Lambda 2 \to \Lambda 3 \to \Lambda 4 \to \Lambda 5$. Для любых тестов домены образуют подобные ориентированные графы (в виде дерева). Оценка истинности J также как любой тест может выражаться с помощью шкал (доменов) разного уровня общности – от числовой шкалы до лингвистической шкалы.

Примеры событий:

<Кашель/{Имеется; Отсутствует}? Имеется, $t/\Lambda 1$? 23.02.08: 12.45>,

<¬’Le крови’/{Снижено; Норма; Повышено}? Норма, $t/\Lambda 3$? 23.02.08>;

АД = <АДс/$D1$? 145, АДд/$D1$? 93, $t/\Lambda 1$? 13.05.09: 12.45>;

<Ток солнечных батарей/{Низкий; Рабочий}?Низкий, $t/\Lambda 2$? [9.00;12.00]>;

<Напряжение нагрузки/{Низкое; …; Высокое}? Низкое, $t/\Lambda 1$? 23.02.08: 12.45>

Любая ситуация действительности (образ) может храниться в упакованном виде, взаимодействуя с другими ситуациями при помощи ключей – минимального набора тестов. Распаковка ситуации может осуществляться под управлением заданного аспекта (множества значений некоторых тестов) на основе фрактального алгоритма, т. е. приводить к бесконечно развертывающемуся ряду событий. Смена аспекта может привести к смене ряда. В силу потенциальной бесконечности ситуация может просматриваться фрагментами, с помощью локальной фокусировки. В дальнейшем, при описании произвольной ситуации, будем понимать именно локальный фокус, который интересует исследователя в данный промежуток времени.

Множество ситуаций действительности обозначим через $\Omega = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Конфигураторы тестов. Конфигураторы тестов являются частью онтологии предметной области (ПрО) [6] и являются процедурным уровнем представления ориентированных графов доменов произвольных тестов $\{T \to T'\}_\tau$.

Базовым доменом в конфигураторе теста называется домен, с помощью которого описываются максимально точные значения теста. Дискретный домен максимальной общности называется N-*арным конструктом теста*, где N – число элементов домена. Примеры бинарных конструктов: {Истина; Ложь}, {Норма: Отклонение}, {Эффективно; Неэффективно}, {Жалобы есть; Жалоб нет}, {Благоприятный; Неблагоприятный} и т. д. Для одного и того же теста может быть задано потенциально сколь угодно много N-арных конструктов. Между базовым доменом теста и N-арным конструктом также может быть задано сколь угодно много промежуточных доменов. Внутри иерархии доменов задаются правила пересчета элементов одного домена в элементы домена более высокого уровня общности. Правила пересчета могут быть самыми разными: детерминированными, нечеткими, нейросетевыми и т. д.

Конкретную иерархию доменов, основанием которой служит базовый домен, вершинами – N-арные конструкты, с фиксированными правилами пересчета назовем *конфигуратором теста*.

Доменная структура в рамках конфигуратора представляет собой дерево. Общую схему конфигураторов с использованием синтаксиса лексических деревьев [3] можно представить следующим образом:

Тест [^ Тест...] [# Тест$_X$...] {
Dom_1 [^Dom_1...] [#DomX] { ; ; }
Dom_2 [^Dom_2...] [#DomY] { ; ; }
...
Dom_N [^Dom_N...] { ; ; }},

где ‘Тест’ – название теста; ‘^ Тест...’ – список условных обозначений теста; ‘# Тест$_X$...’ – список ссылок на более общие тесты; ‘ Dom_K’ – название K-го домена; ‘^Dom_K...’ – список условных обозначений K-го домена; ‘#DomX’ – ссылка на домен предок; {; ;} – список элементов домена. Каждый элемент домена может иметь собственный список обозначений, которые также играют роль символов групп обобщения. Элементы доменов могут содержать параметры, которые обеспечивают однозначность вычислительных схем в зависимости от тех или иных факторов, например, пола.

Порядок размещения доменов в конфигураторе – сверху вниз и слева направо – означает рост точности значений теста за счет большей детали-

зации (увеличения числа элементов). В упорядоченной последовательности доменов метки элементов домена явно задают однозначные правила перерасчета значений из текущего домена в другой, размещенный выше или слева.

Реализовать конфигураторы для конкретной задачи можно, например, с помощью электронной таблицы. Примеры конфигураторов:

Уровень гемоглобина ^Hb {D1 {Нормальный уровень Hb ^N; Ненормальный уровень Hb ^a b} D2 {Анемия ^a; Нормальный уровень Hb ^N; Повышенный уровень Hb ^b} D3 {Резко выраженная анемия ^a [20; 50]; Выраженная анемия ^a (50; 90]; Умеренная анемия ^a (90;120]; Нормальный уровень Hb ^N (120; 160]; Уровень Hb умеренно повышен ^b (160;170]; Повышенный уровень Hb ^b (170;180]; Резко повышенный уровень Hb ^b (180; 200] } D4 {[20; 200]}}

Индекс массы миокарда ЛЖ ^18 R ИММЛЖ {
3 { Норма ^0;
Увеличение ИММЛЖ ^1 2 3}
2 { Норма ^0 М [30; 125] Ж [30; 110];
Умеренное увеличение ИММЛЖ ^1 М (125; 150] Ж (110; 135];
Значительное увеличение ИММЛЖ ^2 М (150; 200] Ж (135; 185];
Резкое увеличение ИММЛЖ ^3 М (200; 400] Ж (185; 400]}
1 { [30; 400]}}

Возраст {
5 #2 {молодой ^ a; не молодой ^ b c}
4 #2 {средних лет ^ b; не средних лет ^ a c}
3 {пожилой ^ c; не пожилой ^ a b}
2 {молодой ^ a [18;35]; средних лет ^ b [36;55]; пожилой ^ c [56;65]}
1 { [18; 65] }}

Температура блока химических батарей ^ТБХБ {
D5 # D2 {Средняя и ниже ^C H; Выше среднего ^B}
D4 # D2 {Ниже среднего ^H; Средняя и выше ^C B}
D3 {Средняя ^C; Ниже или выше среднего ^H B}
D2 {Низкая ^H [0; 20]; Средняя ^C (20; 30]; Высокая ^B (30; 50]}
D1 {[0; 50]}}

Каждый конфигуратор является, по сути, моделью индуктивного обобщения результатов теста. Для числовых тестов ключевым моментом является переход от непрерывного интервала к дискретному разбиению (фа-

зовый переход от бесконечности к конечности). Подобный переход может быть выполнен разными способами, что, безусловно, отражается на результатах моделирования. Важно отметить, что конфигураторы стирают границу между непрерывным и дискретным – любой непрерывный тест всегда имеет множественное дискретное представление (интервальное, символическое).

Для числовых тестов конфигуратор кроме лингвистической иерархии может включать иерархию в виде вложенных разбиений или физического фрактала типа «Канторовская пыль». Иерархия строится между базовым доменом и первым дискретным разбиением (в конфигураторе задается тип разбиения/фрактала и его параметры – скейлинг). Другими словами, конфигураторы числовых тестов не имеют в основании неделимых элементов или «атомов простоты», демонстрируя бесконечность свойства и отсутствие предела процессов делимости материи и информации. Так между доменами D4 и D3 теста $\wedge Hb$ (или D1 и D2 теста ^ТБХБ) могут быть автоматически сгенерированы дополнительные домены по типу вложенных разбиений/фрактала. Данный факт несет большую философско-методологическую нагрузку, так как позволяет наглядно проиллюстрировать взаимодействие на уровне «сознание - подсознание». Действительно, символическая иерархия (в примере $\wedge Hb$: D3 $\rightarrow$ D2 $\rightarrow$ D1) формирует макроуровень в описании теста, а иерархия вложенного интервального разбиения/фрактала (D4 $\rightarrow \ldots \rightarrow$ D3) – микроуровень. Так как макроуровень выразим в языке, он контролируется «сознанием», т. е. логикой, а микроуровень в силу своей потенциальной бесконечности принципиально не может контролироваться «сознанием», хотя именно на этом уровне выполняется основная вычислительная работа. Любое «число» в подобной вычислительной среде автоматизмами среды представляется бесконечным рядом элементов возрастающей общности (согласно конфигуратору), следовательно, любые операции выполняются не над числовыми значениями тестов, а над бесконечными рядами значений (элементы ряда отличаются уровнем общности), реализуя модель «информационного взрыва». Разные комбинации доменов тестов в таких операциях могут использовать разные механизмы реализации с разным временем выполнения (применяется интервальная, нечеткая, фрактальная, нейросетевая математика).

Пусть заданы конфигураторы $\{fr_\tau\}$ тестов $\{\tau\}$. *Замыканием* произвольного множества значений $\{\mathfrak{T}/T\}$ над конфигураторами $\{fr_\tau\}$ назовем множество $\{\mathfrak{T}/T\}^+$, содержащее все вычисляемые на основе конфигураторов значения тестов из исходного множества $\{\mathfrak{T}/T\}$.

Верхней гранью множества $\{\mathfrak{T}/T\}$ над конфигураторами $\{fr\tau\}$ назовем множество $\{\mathfrak{T}/T\}^{\wedge}$, являющееся подмножеством замыкания $\{\mathfrak{T}/T\}^{+}$ и состоящее из результатов тестов максимального уровня общности.

Нижней гранью множества $\{\mathfrak{T}/T\}$ над конфигураторами $\{fr\tau\}$ назовем множество $\{\underline{\mathfrak{T}/T}\}$, являющееся минимальным подмножеством $\{\mathfrak{T}/T\}$, замыкание которого содержит $\{\mathfrak{T}/T\}$, т.е. $\{\underline{\mathfrak{T}/T}\} \subset (\{\mathfrak{T}/T\})^{+}$.

Согласно определениям $\{\mathfrak{T}/T\}$, $\{\mathfrak{T}/\underline{T}\}$, $\{\mathfrak{T}/T\}^{\wedge} \subset \{\mathfrak{T}/T\}^{+}$. Очевидно, выполняется соотношение: $\{\mathfrak{T}/T\}^{+} = (\{\mathfrak{T}/T\})^{+}$. Во многих случаях для принятия решения достаточно информации $\{\mathfrak{T}/T\}^{\wedge}$.

Операции вычисления замыкания, нижней и верхней граней следует отнести к автоматизмам вычислительной среды.

При хранении ситуаций действительности в базе прецедентов достаточно хранить нижние грани множеств результатов тестов в каждый момент времени, а именно: $\alpha(\{<\mathfrak{T}/T, t/\Lambda>\}) \equiv \alpha(\{<\{\underline{\mathfrak{T}/T}\}, t/\Lambda>\})$. При извлечении ситуации α из памяти с помощью автоматизмов среды возможно генерирование полного описания ситуации: $\alpha(\{<\{\mathfrak{T}/T\}^{+}, t/\Lambda>\})$.

Модель вычислительной среды ДИ СППР и Модель ПрО. Модель ПрО представим в виде кортежа $< O, k >$, где O – модель онтологии этой предметной области, а k - модель адекватной системы знаний. Онтология O включает в себя, в частности, Банк тестов вместе с конфигураторами каждого теста [3; 4; 7].

Модель знаний k включает в себя следующие компоненты [7]: $k = k_C \cup k_M \cup k_A \cup k_O$, где k_C – вычислительные знания; k_M – описания моделей динамических процессов данной ПрО; k_A – множество аксиом, фактов, высказываний с оценкой их истинности; k_O – прочие знания. Модель знаний k формируется путем системной реконструкции множества ситуаций действительности $\Omega = \{\alpha\}$.

В развернутом виде модель вычислительных знаний k_C представима в виде [7]

$$k_C = \{f/\mu: k^1 \rightarrow k^2 \mid \mu \in \{\mu\}_f\} \cup P_k,$$

где f/μ – отображения, реализующие те или иные математические модели; $\{\mu\}_f$ – разные механизмы реализации отображений (со своей энергетикой и ресурсами) ; k^1 – входные данные задачи (описание информационной среды и задание); k^2 – выходные данные задачи; P_k – правила композиции схем задач, т. е. правила, описывающие способы объединения локальных задач.

Обозначим элементарные тесты через $a, b, c, d, \{a\}$ и т. д. Пусть $W(\{c/C\})$ – некоторое многообразие на множестве результатов тестов

$\{c/C\}$. С учетом принятой модели эмпирических данных любое отображение модели знаний k_C можно представить следующим образом:

$$f/\mu\colon \{J_\beta\ b/B\} \to \{J_\gamma\ a/A\},\quad \text{для } \{c/C\} \in W_f(\{c/C\}),\ \mu \in \{\mu\}_f.$$

В [1; 6; 7] описан процедурный уровень реализации представленных моделей с использованием концепции Многоцелевых банков знаний. Формат произвольного отображения с использованием Банка тестов (ФО – функциональное отображение):

<Полное имя ФО> – f {ФО_<Аббревиатура для ФО>
[Условие {cond_
<тесты - условие> – $\{c/C\}$}]
Вход {inp_
<входные тесты> – $\{b/B\}$}
Выход {out_
<выходные тесты> – $\{a/A\}$}
Методы {met_
<методы обработки> – $\{\mu\}_f$ }}

Отображения объединяются в семантические группы, разного уровня вложенности:

<Имя предметной области> {ПРО_<СемантИмя>
[Определения {def_
<Полное имя теста> ^Алиасы [{Ссылка}]
….}]
[<ПРО_Имя>]
<ФО_1>
…
<ФО_N>
} [ПрО]

Информационные множества и множества обобщения. Под *информационным множеством* $I_\gamma(\alpha,\{\tau/T\})$ для ситуации α и аспекта $\{\tau/T\}$ будем понимать совокупность всех гипотетических ситуаций действительности, удовлетворяющих заданным критериям истинности γ и совместимых с ситуацией α на уровне общности $\{\tau/T\}$. Можно записать

$$I_\gamma(\alpha,\{\tau/T\}) = \{\beta \mid \{\tau/T\}_\beta = \{\tau/T\}_\alpha\ \&\ \gamma(\beta) = true\},$$

где $\gamma(\ )$ – оператор оценки истинности информации, который реализует одну из когнитивных функций базы знаний.

Пример аспекта для ситуаций действительности (из медицинской практики): {τ/T} = {Пол?Ж; Ds? Бронхит; Возраст/ B2? 40-55; Жалобы/ D? Отсутствуют; Об_но {…}; Биохимия {…}}.

В простейшем случае информационное множество может быть получено путем замены в ситуации α точечных событий на совместимые с ними протяженные события, а именно:

$$\alpha = \alpha(\{<J_\tau\ \tau/T,\ J_t\ t/\Lambda>\}) \rightarrow I(\alpha) = \{\beta = \beta\ (\{<J_\tau\ \tau/T,\ J_\delta\ \delta/\Lambda>\})\}.$$

В самом общем случае ситуации β из информационного множества охватывают прошлое, настоящее и будущее развитие ситуации α. Для подобного моделирования необходимо определить понятия орбиты для ситуации действительности с учетом возможности многоуровневого описания, а также эмпирический оператор эволюции произвольной ситуации действительности.

Символический образ фрагмента орбиты ситуации α длительностью δ/Λ в пространстве тестов $\{a/A\}$ определим следующим образом:

$Orb_\alpha(\{a/A\}, \delta/\Lambda) = \cup_\nu \{<\omega/W, \delta/\Lambda, \{p/P\}>_{i=1,N}\}_\nu$,
$\forall\nu \quad \Sigma_{i=1,N}(\delta_i/\Lambda)_\nu = \delta/\Lambda$,

где ν – индекс схемы разбиения пространства на ячейки (схемы обобщения); N_ν – общее число ячеек, через которые проходит траектория за время δ/Λ при ν-ом разбиении; $(\omega_i/W)_\nu$ – имена (или адреса) ячеек ν-го пространства разбиения (схемы обобщения); $(\delta_i/\Lambda)_\nu$ – длительность непрерывного пребывания траектории в ячейке $(\omega_i/W)_\nu$; $(\{p/P\}_i)_\nu$ – характеристики траектории в ячейке $(\omega_i/W)_\nu$, например, такие как плотность, информационное отражение и т. д. Как и ранее все тесты определяются своими конфигураторами.

Обозначим через U({ σ/C}) – окрестность множества { σ/C}. Тот факт, что некоторая ситуация α удовлетворяет условиям U({ σ/C}) будем записывать следующим образом: $\alpha\ \nabla$ U({σ/C}).

Эмпирический оператор эволюции произвольной ситуации действительности $\phi^t()$ определяет значения заданных тестов { a/A} в момент времени t/Λ, используя для этого базу прецедентов Ω

$$\phi^{t/\Lambda}(\{a/A\}/U(\{\sigma/C\}) = \cup_{\alpha\in\Omega}\{f/\mu: t/\Lambda, \{<J_\tau\ \tau/T,\ J_t\ t'/\Lambda>\}_\alpha \rightarrow \{J_a\ a/A\}_\alpha \mid \alpha\ \nabla U(\{\sigma/C\})\},$$

где $f/\mu \in k_C$; $t'/\Lambda = t/\Lambda$ или $t'/\Lambda \in [0, t/\Lambda]$ или $t'/\Lambda \le t/\Lambda$ (выбор варианта зависит от $\{a/A\}$). Результат представляет собой мультимножество. Кроме результата оператор эволюции позволяет дать развернутое пояснение каждому результату, что порой, не менее важно, чем сам результат. Отметим

также, что множество прецедентов Ω может содержать как реальные, так и модельные ситуации.

Начальные значения и параметры подобия разных ситуаций действительности содержатся во множестве $U(\{a/C\})$. Из определений оператора эволюции следует, что область $U(\{a/C\})$ необходимо задавать таким образом, чтобы обеспечить большую устойчивость результата.

Свертка частных результатов позволяет во многих случаях повысить устойчивость общего результата. Если правую часть оператора эволюции обозначить $\cup_{\alpha\in\Omega}\{a/A\}_\alpha$, то *оператор эволюции со сверткой значений* запишется следующим образом:

$$\phi^{t/\Lambda}(\{a/A\}/U(\{a/C\}) \equiv g/\mu_g : t/\Lambda,\ \cup_{\alpha\in\Omega}\{J_a\ a/A\}_\alpha \rightarrow \{J_a\ a/A\}_{t/\Lambda},$$

где $g/\mu_g \in k_C$.

Эмпирический оператор эволюции реализует схему *когнитивного вычислительного моделирования* в рамках ДИ СППР.

Пусть запись $\beta:\alpha$ означает, что ситуация β является обобщением ситуации α, а α рассматривается как конкретизация ситуации β. Обобщение будем связывать с заменой тестов $\{a/A\}_\alpha$ на более общие тесты $\{b/B\}_\beta$. Для такой замены в базе знаний k, являющейся основой оператора γ, должно существовать отображение вида $f/\mu: \{a/A\} \rightarrow \{b/B\}$, с помощью которого осуществляется преобразование значений одной группы тестов в значения другой группы тестов. Главным свойством операции обобщения является ее необратимость, т. е. из α можно получить β, но из β нельзя получить α.

Множеством обобщения ситуации α назовем множество $G_\gamma(\alpha)$ следующего вида:

$$G_\gamma(\alpha) = \{\beta \mid \beta:\alpha\ \&\ \gamma(\beta) = true\}.$$

Пусть, например, ситуация α описывается с помощью базовых доменов N тестов, графы доменов которых содержат M_i доменов (без потери общности примем, что тесты не повторяются), тогда общее число различных описаний действительности, получаемых из α и обобщающих α, составляет $M_1{}^{\times}M_2{}^{\times}\ldots{}^{\times}M_N - 1$ [5]. Положим $\gamma(\beta) = true$, если β является одним из таких описаний. Таким образом, получили точную оценку мощности множества $G_\gamma(\alpha)$ для данного примера, а именно: $|G_\gamma(\alpha)| = M_1{}^{\times}M_2{}^{\times}\ldots{}^{\times}M_N - 1$. Пример с повторяющимися тестами – любой временной ряд. Более сложные примеры: медицинская карта, карта учащегося и т. д.

Эффективным средством формирования множества обобщений для произвольной ситуации действительности является «Метод предельных обобщений» [5].

Приведем пример обобщения из области анализа электроэнцефалог-

рамм (ЭЭГ). Ставится задача найти отличия в ЭЭГ биологических объектов, подвергавшихся определенным видам информационного воздействия, от ЭЭГ этих же биологических объектов в обычной информационной обстановке (фоновых ЭЭГ). Множество всех ЭЭГ образуют ситуацию действительности α. На первом этапе анализа ЭЭГ подвергались предобработке. Сначала производилось сглаживание ЭЭГ по методу скользящего среднего. Затем участки сигнала с положительной первой производной заменялись на 1, остальные на 0. Таким образом, вместо исходного сигнала обработке подвергалась последовательность, состоящая из 0 и 1. Множество всех преобразованных ЭЭГ образуют ситуацию действительности β, при этом $\beta{:}\alpha$.

Множеством обобщения для группы ситуаций $\{\alpha\}$ назовем множество $G_\gamma(\{\alpha\})$ следующего вида:

$$G_\gamma(\{\alpha\}) = \{\beta \mid \beta{:}\{\alpha\} \equiv (\beta{:}\alpha \ \forall\alpha \in \{\alpha\}) \ \&\ \gamma(\beta) = true\}.$$

Следует иметь ввиду, что, как правило, $G_\gamma(\{\alpha\}) \neq \cup_{\alpha \in \{\alpha\}} G_\gamma(\alpha)$.

Задачи построения информационных множеств и множеств обобщения относятся к числу базовых когнитивных процессов, лежащих в основе динамической интеллектуальной СППР. С помощью данных множеств реализуется *информационный взрыв* в задачах принятия решений.

Макромодель ДИ СППР. В наиболее общем виде процедуру принятия любого решения можно отобразить комбинацией трех операторов - оператора расширения λ, оператора локализации решения ϑ и оператора оценки истинности информации γ [4, 7], а именно:

$$X^* = \vartheta(\lambda(X)), \text{ при условии: } E(\lambda) > E_I^*;\ \gamma(\lambda(X)) = \textit{истина}, \qquad (1)$$

где X – первичная (априорная) информация; E – энергия – движущая сила когнитивного процесса; I – цель; X^* – результирующее решение.

Задачей оператора расширения λ является конструирование универсума, адекватного решаемой задаче, т. е. получение максимума с точки зрения цели I достоверной информации (с учетом оператора γ) о прошлом, настоящем и будущем поведении системы, которую можно получить, используя ресурсы R и априорную информацию X. В результате действия оператора расширения происходит резкое возрастание информативности всей системы зачастую сравнимое с «информационным взрывом».

Для успешной реализации λ необходимо создать *критическое усилие*, чтобы найти и заполнить пустоты в семиотическом пространстве (понятие «пустоты», трактуется в восточной философии как ресурс и возможность). Данное требование выражается условием: $E(\lambda) > E_I^*$. Если данное условие не выполняется, то качественное решение (или просто решение) не может

быть получено. Решение задачи прекращается, как только исчезает движущая сила – энергия. Таким образом, локализацию можно рассматривать как синергетический эффект взаимодействия всех информационных процессов, который остается после исчезновения энергии, выделенной на выработку решения. Следовательно, решение всегда является феноменом семантического синтеза.

Параметр E_I^* существенно зависит от физики системы, осуществляющей выработку решения, в частности, от наличия у системы «врожденных» и/или приобретенных автоматизмов (синергий), связанных с получением и преобразованием информации. Чем больше автоматизмов задействовано в процессе выработки решения, тем меньше требуется энергии (меньше значение параметра E_I^*), следовательно, тем больше шансов получить качественное решение. В первом приближении справедлива следующая оценка: $|\lambda(X)| \sim E(\lambda)$.

В терминах модели ситуации действительности задача (1) может быть представлена следующим образом:

$$\{u/U\}^* = \vartheta(\lambda(\{I/T\})), \qquad (2)$$

при условии: $\{e/E\}_\lambda > \{e/E\}_I^*$; $\gamma(\lambda(\{I/T\}))$ = *истина,*

где $\{e/E\}_\lambda$ – параметры, описывающие критическое усилие.

Обозначим σ^t оператор *оценки свойств (характеристик) информации*. Он формирует множество значений $\{p/P\}_t$ – количественных и качественных параметров всесторонней оценки некоторой первичной информации (например, принятого в прошлом решения) в момент времени t. Зависимость от времени определяется тем, что многие характеристики, например, такие как ценность информации (конъюнктурная ценность, прогностическая ценность, асимптотическая ценность), эффективность и обоснованность решения, изменяются со временем (они как бы созревают). Применительно к решению (1) можно записать:

$$\{p/P\}_t = \sigma^t(X^*, \alpha) = \sigma(X^*, t, \alpha) = \sigma(\vartheta(\lambda(X)), t, \alpha), \qquad (3)$$

где α – ситуация действительности (в динамике). В развернутом виде, с учетом (2), можно записать

$$\{p/P\}_{t/\Lambda} = \sigma(\{u/U\}^*, t/\Lambda, \{<J_\tau I/T, J_t t'/\Lambda'>\}\alpha). \qquad (4)$$

Каждый тест в (4), включая время, определяется своим конфигуратором.

В точке бифуркации процесса в результате решения задачи (2) определяется стратегия управления (принимается стратегическое решение). Стратегия управления во времени представляется серией этапов $\pi = <f_1/\delta_1, f_2/\delta_2, \ldots, f_n/\delta_n>$, где f_j – требующий конкретизацию закон управления на j-ом этапе, а δ_j – предполагаемая длительность этапа. Конкретизацию закона

управления на j-ом этапе, т. е. локализацию решения обозначим δ_{j-1}/δ_j. Без потери общности будем считать, что задача (2) выполняется на старте программы управления и в моменты смены этапов. Моменты смены программ управления π, т. е. точки бифуркации обозначим через t^*.

Обозначим через Ψ оператор изменения действительности или оператор *перехода* [4; 7]. Оператор перехода зависит от решения X^* и действующих возмущений υ. Он отвечает на главный вопрос «что произойдет, в конце концов, в результате реализации данной стратегии (или текущего этапа управления)». Математическими образами таких устойчивых, предельных режимов являются притягивающие множества в фазовом пространстве или аттракторы.

Если через t обозначить время смены этапов δ_{j-1}/δ_j (j=1,…,n; $\delta_0 = t^*$), а через (t+1) время δ_j/δ_{j+1}, то работу ДИ СППР в рамках реализации выбранной стратегии π можно описать следующей схемой [4; 7]:

$$X(t+1) = \lambda(\Psi(\vartheta(X(t)), \upsilon)), \qquad (5)$$

при условии: π *фиксировано,* $new\ t^* \notin [t, t+1]$, $\gamma(X) =$ *истина*; $E(\lambda) > E_I^*$.

Подчеркнем, что все операторы в (5) зависят от цели управления (принятия решений) и имеющихся ресурсов. Ресурсы могут изменяться скачкообразно.

В терминах модели ситуации действительности схему (5) можно представить следующим образом:

$$\{\mathfrak{T}/T\}_{t/\Lambda+1} = \lambda(\Psi(\vartheta(\{\mathfrak{T}/T\}_{t/\Lambda}), \{<\mathfrak{V}/V, \mathfrak{t}'/\Lambda'>\}_{[t/\Lambda,\,(t/\Lambda)+1]})), \qquad (6)$$

При условии: π *фиксировано,* $new\ t^*/\Lambda \notin [t/\Lambda, (t/\Lambda)+1]$,
$\gamma(\{\mathfrak{T}/T\}) =$ *истина*; $\{\mathfrak{e}/E\}_\lambda > \{\mathfrak{e}/E\}_I^*$.

Типичные цели управления I на языке нелинейной динамики могут быть сформулированы следующим образом:

- «Завести траекторию системы в заданное поглощающее множество $W = \{\mathfrak{T}/T\}_W$»;
- «Обеспечить асимптотическое движение системы к притягивающему или инвариантному множеству $W = \{\mathfrak{T}/T\}_W$»;
- «Обеспечить максимум функции жизнеспособности системы (при заданных ресурсах)».

В [4] рассматриваются варианты процедурной реализации оператора расширения λ. Оператор перехода Ψ может быть смоделирован на основе одноименного фрагмента вычислительных знаний

$$\Psi_k = \{f/\mu\colon \{<J_\tau\ \mathfrak{T}/T,\ J_t\ \mathfrak{t}/\Lambda>\} \rightarrow \{<J_\tau\ \mathfrak{T}/T,\ J_t\ \mathfrak{t}/\Lambda>\}' \mid \mu \in \{\mu\}_f\} \cup P_k, \qquad (7)$$

События в модели (7) учитывают как управляющие воздействия вида (2), так и возмущения вида $< J_{\upsilon}\, \mathfrak{V}/V, J_t\ t/\Lambda >$, которые имели место в период $[t/\Lambda, (t/\Lambda)']$.

Выводы. Построена модель ДИ СППР, которая обладает рядом полезных свойств, среди которых – нежесткие требования к формализму, учет имеющихся ресурсов и изменчивости критериев истинности, невысокая чувствительность модели к априорной полноте описания состояний, высокая способность к адаптации в изменяющейся среде, учет реальной компетентности управляющего агента. Эти свойства становятся решающими при управлении поведением саморазвивающихся систем.

Библиографические ссылки

1. **Алпатов А.П.** Информационные технологии в образовании и здравоохранении / А.П. Алпатов, Ю.А. Прокопчук, О. В. Юденко, С. В. Хорошилов– Д., 2008. - 287 с.
2. **Колесов Ю.Б.** Моделирование систем. Динамические и гибридные системы / Ю.Б. Колесов, Ю.Б. Сениченков – СПб., 2006. - 224 с.
3. **Прокопчук Ю.А.** Интеллектуальные медицинские системы: формально - логический уровень./ Ю.А. Прокопчук– Д., 2007.- 259 с.
4. **Прокопчук Ю.А.** Интеллектуальное синергетическое управление динамическими системами / Ю.А. Прокопчук // Искусственный интеллект, 2009. - №4.- С. 12 – 21.
5. **Прокопчук Ю.А.** Метод предельных обобщений для решения слабо формализованных задач / Ю.А. Прокопчук // Управляющие системы и машины. – 2009. - №1. – С.31 – 39.
6. **Прокопчук Ю.А.** Архитектура многоцелевых банков знаний в области клинической медицины / Ю.А. Прокопчук // Труды Всероссийской конференции «Знания -Онтологии-Теории»-2009 в 2 -х томах (Новосибирск, 22 - 24 октября 2009 г.). – Новосибирск, Т.1. - С. 173-177.
7. **Прокопчук Ю.А.** Управление эколого -социальными системами на основе интегративной логики и модели предметной области / Ю.А. Прокопчук // Збірн. наук. праць Індуктивне моделювання складних систем. – Київ, 2009. – С. 165 – 174.

Надійшла до редколегії 12.01.10

УДК 004.942

М. Є. Сердюк
Дніпропетровський національний університет імені Олеся Гончара

ПРО АНАЛІТИЧНЕ КОНСТРУЮВАННЯ ІНТЕРПОЛЯЦІЙНОГО СПЛАЙНУ МІНІМАЛЬНОЇ ЛОКАЛЬНОЇ КРИВИНИ ТА ЙОГО ЗАСТОСУВАННЯ ДЛЯ КАРКАСНОЇ ІНТЕРПОЛЯЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ

Запропонована конструкція шеститочкового інтерполяційного сплайну з покращеними характеристиками кривини на середній частині відрізку його завдання. Отримано аналітичне подання такого сплайну та проведено порівняльний аналіз з відомими сплайнами.

Предложена конструкция шеститочечного интерполяционного сплайна с улучшенными характеристиками кривизны в средней части отрезка его задания. Получено аналитическое представление такого сплайна и проведен сравнительный анализ с известными сплайнами.

The structure of six-point interpolation spline with improved characteristics of curvature is proposed. The analytic representation of such spline is received. The comparative analysis with known splines is performed.

Ключові слова: інтерполяційний сплайн, каркасна інтерполяція, цифрове зображення.

Вступ. Інтерполяція сплайнами широко застосовується в багатьох областях обчислювальної математики. Зокрема, сплайни використовуються для побудови функції зображення в задачах просторової інтерполяції статичних растрових зображень. Шукана функція повинна будуватися з урахуванням вимог, які висуваються до інтерпольованого зображення. До таких вимог відноситься збереження достатньої гладкості контурних ліній (відсутність aliazing_ефекту), різкості та чіткості результуючого зображення. Звідси виникає задача побудови функції за відомими значеннями її в декількох точках так, щоб кривина отриманої лінії була найменшою. Для розв'язання цієї задачі доцільно застосувати саме інтерполяцію поліноміальними сплайнами. У [1] використання інтерполяційного сплайну міні-

льної локальної кривини запропоновано для відновлення функції зображення на каркасі, який є наближенням топографічної карти зображення і визначається за певним алгоритмом. При цьому, сплайн будується на шести точках обраного фрагмента каркасу, а конструкція сплайна визначена, виходячи із умови мінімуму кривини отриманої лінії на всьому фрагменті. Але для прогнозування значень функції зображення використовується лише середня частина обраного фрагмента. Отже саме вона має найбільш вагоме значення при відновленні функції зображення. Тому варто розглянути задачу побудови сплайну з покращеними характеристиками кривини у середині відрізку. У даній роботі запропонована конструкція сплайну, який будується за значеннями функції в шести опорних точках за умови мінімуму кривини у середній частині відрізку.

Попередні теоретичні положення. Нехай $\{t_i = (i-1)h\}_{i=1}^{N}$ є рівномірною сіткою на відрізку $[0,W]$. Позначимо через $\tilde{Z}$ простір числових послідовностей довжини N. Покладемо $Z = H^2(0,W)$. Нехай $T \in L(Z,\tilde{Z})$ – оператор звуження Z на $\tilde{Z}$, який означимо як $Tx = \{x(t_1), x(t_2),..., x(t_N)\}$. Нехай Y – деякий гільбертів простір і $B \in L(Z,Y)$ – заданий оператор.

Означення 1. Функцію $f \in Z$ будемо називати (T,B)-сплайном, що відповідає вектору $\tilde{x} = \{x(t_1), x(t_2),..., x(t_N)\} \in \tilde{Z}$, якщо виконується наступна умова

$$\|Bf\| = \inf_{x \in T^{-1}(\tilde{x})} \|Bx\|. \qquad (1)$$

Тут через $T^{-1}(\tilde{x})$ позначено множину розв'язків наступного операторного рівняння

$$Tx = \tilde{x}.$$

Позначимо через $N(T)$ і $N(B)$ підпростори нулів для операторів T та B відповідно. Ясно, що кожна з означених множин лежить у просторі Z.

У [3] показано, що якщо лінійна багатоздатність $BN(T)$ є замкненою і при цьому $N(B) \cap N(T) = \{0\}$, то для кожного вектора $\tilde{x} = \{x(t_1), x(t_2),..., x(t_N)\} \in \tilde{Z}$ знайдеться єдиний (T,B)-інтерполяційний сплайн $f \in Z$. Крім того, показано, що множина $BN(T)$ є замкненою, якщо оператор $B \in L(Z,Y)$ є нормально розв'язним, тобто множина йо-

го значень $R(B)$ є замкненою в Y, його підпростір нулів $N(B)$ є скінченно вимірним і при цьому $N(B)\bigcap N(T)=\{0\}$. Отже, задача про (T,B)-інтерполяційний сплайн має єдиний розв'язок, якщо виконані наступні умови: 1) $R(B)=Y$; 2) $N(B)$ – скінченно вимірна множина; 3) $\tilde{Z}$ – скінченно вимірний простір; 4) $\dim N(B)=k<l=\dim\tilde{Z}$; 5) $N(B)\bigcap N(T)=\{0\}$.

Постановка задачі. Нехай $\{c_{-3},c_{-2},c_{-1},c_1,c_2,c_3\}$ – заданий масив значень, за якими необхідно побудувати функцію. Для задачі каркасної інтерполяції – це значення функції зображення у пикселах, які обрані за певним алгоритмом як ті, що належать фрагменту лінії рівня зображення. Припускаємо, що відстань між двома будь -якими сусідніми пікселами дорівнює одиниці. Це можливо, якщо метрику означити за наступним правилом:

$$d(x,y)=\max\{|x_1-y_1|,|x_2-y_2|\}\quad\forall x,y\in\overline{\Delta},$$

де $\overline{\Delta}=[0;W]\times[0;H]$ – задана підмножина в R^2. Отже, можна вважати, що $c=\{c_i\}$, $i=-3,-2,...3$ – значення функції зображення в точках $[-5/2,-3/2,-1/2,1/2,3/2,5/2]$. За цими значеннями необхідно спрогнозувати значення функції на інтервалі $\left(-\frac{1}{2},\frac{1}{2}\right)$.

Нехай $Z=H^2\left(-\frac{5}{2},\frac{5}{2}\right)$, $Y=L^2\left(-\frac{5}{2},\frac{5}{2}\right)$, $\{t_i=-\frac{7}{2}+i, i=1,...,6\}$ – система опорних точок на відрізку $\left[-\frac{5}{2},\frac{5}{2}\right]$, $x=\{x_i\}_{i=1}^6$ - вектор вхідних даних, для якого виконуються умови

$$x_1=c_{-3},\ x_2=c_{-2},\ x_3=c_{-1},\ x_4=c_1,\ x_5=c_2,\ x_6=c_3,$$

лінійний неперервний оператор $B\in L(D(B),Y)$ задається правилом

$$Bx(t)=x''(t),\ x(t)\in D(B).$$

Уведемо оператор B наступним чином. Будемо казати, що функція $s(t)$ належить області визначення $D(B)$ оператора B, якщо

$$s(t) = \begin{cases} s_1(t) &= a_1t^5 + b_1t^4 + g_1t^3 + d_1t^2 + e_1t + f_1, \quad t \in [-5/2,-1/2], \\ s_2(t) &= a_2t^3 + b_2t^2 + g_2t + d_2, \quad t \in [-1/2,1/2], \\ s_3(t) &= a_3t^5 + b_3t^4 + g_3t^3 + d_3t^2 + e_3t + f_3, \quad t \in [1/2,5/2], \end{cases} \quad (2)$$

$$\begin{gathered} s_1'(-1/2) = s_2'(-1/2), \quad s_2'(1/2) = s_3'(1/2), \\ s_1''(-1/2) = s_2''(-1/2), \quad s_2''(1/2) = s_3''(1/2), \end{gathered} \quad (3)$$

$$s(-5/2+i) = c_{-3+i}, \quad s(5/2-i) = c_{3-i}, \forall i = 0,1,2 \quad (4)$$

Нехай оператор проектування $T \in L(Z, \tilde{Z})$ задається правилом

$$Tx(t) = \{x(t_i)\}_{i=1}^{6}.$$

Формула (1) у відповідності до означення (1) може бути переписаною у вигляді

$$\|Bf(t)\|^2 = \inf_{s(t)\in\Omega} \|Bs(t)\|^2 = \inf_{s(t)\in\Omega} \int_{-1/2}^{1/2} [s'']^2 dt,$$

де $\Omega = \{s \in D(B) : Ts = \{x(t_i)\}_{i=1}^{6}\}$. Отже, задача побудови (T,B)-сплайну складається у пошуку на множині Ω функції $s \in Z$, яка реалізує мінімум наступного функціонала

$$I(s) = \int_{-1/2}^{1/2} [s''(t)]^2 dt. \quad (5)$$

Оскільки кривина кривої пропорційна другій похідній функції, що описує цю криву, то (T,B)-сплайн $s(t)$, який є розв'язком сформульованої задачі умовної мінімізації є функцією мінімальної на $\left[-\frac{1}{2},\frac{1}{2}\right]$ локальної кривини.

Метод розв'язування та аналіз отриманих результатів. Перевіримо умови розв'язності даної задачі, використовуючи наведені вище умови 1) –5) задачі про (T,B)-інтерполяційний сплайн. Областю значень оператора $B = \frac{d^2}{dt^2}$ є гільбертів простір $Y = L^2\left(-\frac{5}{2},\frac{5}{2}\right)$, множина його нулів $N(B)$ є чотиривимірним підпростором многочленів, отже $\dim N(B) = 4$. Разом з тим, апроксимуючий простір $\tilde{Z}$ є 6-вимірним евклідовим простором R^6. Це означає, що умови 1) -4) виконуються. Залишається перевірити умову

5), тобто показати, що $N(B)\bigcap N(T)=\{0\}$. Нехай s є довільною функцією з множини $N(B)\bigcap N(T)$. Тоді, з однієї сторони, s повинна бути поліномом степеня не вище третього, а з іншої, приймати нульові значення на системі точок $\{x(t_i)\}_{i=1}^{6}$. Ясно, що це можливо лише за єдиної умови $s(t)\equiv 0$. Таким чином, у відповідності до наведених вище результатів задача умовної мінімізації

$$\int_{-1/2}^{1/2}[s''(t)]^2\,dt\to\inf_{s\in\Omega} \tag{6}$$

має єдиний розв'язок $s(t)\in H^2\left(-\frac{5}{2},\frac{5}{2}\right)$.

Функцію $s(t)\in H^2\left(-\frac{5}{2},\frac{5}{2}\right)$ будемо називати шеститочковим (T,B)-сплайном, побудованим на ланцюгу вхідних даних $\{c_{-3},c_{-2},c_{-1},c_1,c_2,c_3\}$. Ця функція має мінімальну локальну кривину на відрізку $\left[-\frac{1}{2},\frac{1}{2}\right]$ і відтворює функцію зображення на цьому відрізку за її значеннями у шести точках фрагмента лінії рівня. Очевидно, що мінімізація функціонала (6) приводить до такої функції $s(t)$, де $s_2(t)$ (див. формулу (2)) є лінійною. Тобто на відрізку $\left[-\frac{1}{2},\frac{1}{2}\right]$ отримаємо пряму лінію. Але такий результат не є задовільним, оскільки лінійна інтерполяція сприяє появі «сходинкового» ефекту в результуючому зображенні. Тому пропонується модифікувати функціонал (6) та розв'язувати ту ж саму задачу умовної мінімізації з функціоналом

$$\int_{-1}^{1}[s''(t)]^2\,dt\to\inf_{s\in\Omega}. \tag{7}$$

Таким чином, щоб визначити аналітичне подання функції $s(t)$, необхідно знайти невідомі коефіцієнти $a_1,...,f_1,a_2,...,d_2,a_3,...,f_3$ у формулах (2) за умови (3)-(4) та за умови мінімуму функціонала (7).

Для визначення шістнадцяти невідомих коефіцієнтів маємо дванадцять рівнянь, які отримуємо із умов (2)-(4). Ця система рівнянь є лінійною і має безліч розв'язків. Отже, можемо знайти загальний її розв'язок, обравши вільними чотири невідомі. Тоді отримаємо вирази для дванадцяти коефі-

цієнтів через чотири вільні. Далі необхідно визначити той набір коефіцієнтів, на якому функція (2) реалізує мінімум функціоналу (7). Підставимо до функціонала (7) функцію $s(t)$, яка визначається формулою (2). Після відповідних перетворень отримаємо вираз, до якого входять десять із шістнадцяти невідомих коефіцієнтів:

$$I = \frac{3175}{56}(a_1^2 + a_3^2) + \frac{279}{10}(b_1^2 + b_3^2) + \frac{21}{2}(g_1^2 + g_3^2) + 2(d_1^2 + d_3^2) + \\ + \frac{315}{4}(a_3 b_3 - a_1 b_1) + \frac{93}{2}(a_1 g_1 + a_3 g_3) + \frac{75}{4}(a_3 d_3 - a_1 d_1) + \\ + \frac{135}{4}(b_3 g_3 - b_1 g_1) + 14(b_1 d_1 + b_3 d_3) + 9(d_3 g_3 - d_1 g_1) + \\ + 3a_2^2 + 4b_2^2 \qquad (8)$$

Цей вираз будемо розглядати як функцію десяти змінних. Після підстановки коефіцієнтів, знайдених за результатом розв'язання системи рівнянь, і відповідних перетворень отримаємо із (8) функцію чотирьох змінних. Далі розв'язуємо задачу пошуку мінімуму функції чотирьох змінних. Для цього прирівнюємо до нуля частинні похідні цієї функції за кожною із змінних, дістаємо систему з чотирьох рівнянь, розв'язуючи яку отримаємо останні чотири коефіцієнти.

Описана процедура здійснена у системі Maple, в результаті був отриманий вигляд (T, B)-сплайну (2) на відрізку $\left[-\frac{5}{2}, \frac{5}{2}\right]$. Так, на центральній частині $\left[-\frac{1}{2}, \frac{1}{2}\right]$ цей сплайн має наступне подання

$$s(t) = \left[\frac{127}{45118}t^3 - \frac{245}{93172}t^2 - \frac{127}{180472}t + \frac{245}{372688}\right]c_{-3} + \\ + \left[-\frac{4490}{22559}t^3 + \frac{4459}{23293}t^2 + \frac{2245}{45118}t - \frac{4459}{93172}\right]c_{-2} + \\ + \left[\frac{26305}{45118}t^3 - \frac{17591}{93172}t^2 - \frac{206777}{180472}t + \frac{203935}{372688}\right]c_{-1} + \\ + \left[-\frac{26305}{45118}t^3 - \frac{17591}{93172}t^2 + \frac{206777}{180472}t + \frac{203935}{372688}\right]c_1 + \\ + \left[\frac{4490}{22559}t^3 + \frac{4459}{23293}t^2 - \frac{2245}{45118}t - \frac{4459}{93172}\right]c_2 + \\ + \left[-\frac{127}{45118}t^3 - \frac{245}{93172}t^2 + \frac{127}{180472}t + \frac{245}{372688}\right]c_3. \qquad (8)$$

Для отриманого сплайну був проведений порівняльний аналіз із кубічним сплайном мінімального дефекту [2] та сплайном, запропонованим в [1]. На малюнку 1 наведені три приклади наборів значень функції у шести точках, за якими здійснювалась інтерполяція, та відповідні результати інтерполяції на відрізку $\left[-\frac{1}{2},\frac{1}{2}\right]$ за допомогою кубічного сплайну (пунктирна лінія), сплайну, запропонованого в [1] (тонка безперервна лінія), сплайну (8) (товста лінія). Характерним для обраних прикладів є те, що інтерполяція за аналогічними множинами значень функції приводить до появи так званих «хибних» екстремумів. А це, у свою чергу, стає причиною «розмиття» результуючого зображення, недостатньої його чіткості. Як видно із рисунків, використання саме сплайну (8) дозволяє отримати найбільш прийнятний результат.

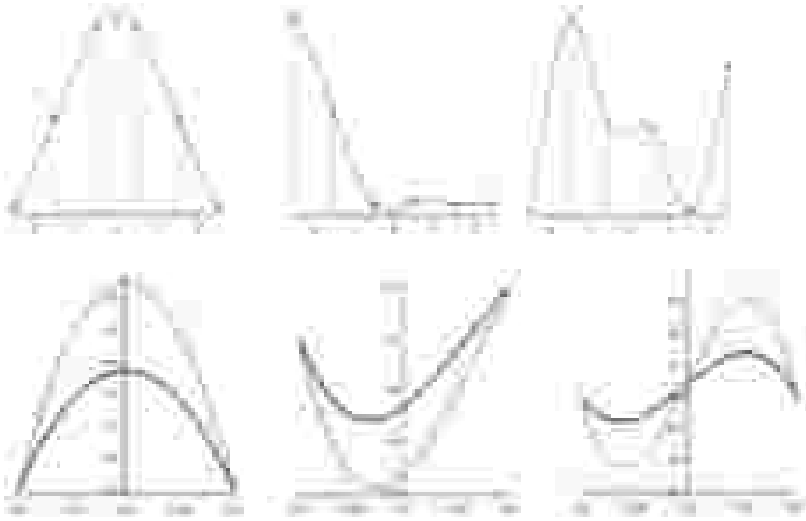


Рис.1. Результати інтерполяції на середній частині відрізку за обраними наборами значень функції у шести точках

Висновки. Запропонована процедура будування шеститочкового (T,B)-сплайну дозволила отримати подання сплайну, який на центральній частині відрізку інтерполяції має кращі характеристики кривини, ніж кубічний сплайн мінімального дефекту або сплайн, наведений в [1]. Це робить доречним залучення його для відновлення функції зображення в

задачах просторової інтерполяції зображень, оскільки це сприяє збереженню гладкої неперервності контурів та чіткості зображення, що у свою чергу дозволяє зберегти якість інтерпольованих зображень.

Бібліографічні посилання

1. **Когут П.І.** Задачі каркасної інтерполяції статичних зображень / П. І. Когут, М. Є. Сердюк // Вестник Херсонского национального технического университета. – 2007. – № 2(28). – С.153–157.

2. **Корнейчук Н.П.** Сплайны в теории приближения / Н.П. Корнейчук. – М., 1984. – 452 с.

3. **Треногин В.А.** Функциональный анализ / В.А.Треногин. – М., 2002. – 496 с.

Надійшла до редколегії 03.11.09

УДК 519.5

©**Е. А. Татаринов**
Институт прикладной математики и механики НАНУ, г. Донецк

M-НУМЕРАЦИЯ, КАК МЕТОД РАСПОЗНАВАНИЯ ГРАФОВ

Розглядається задача розпізнавання графа мобільним агентом. Агент рухається графом, змінює та зчитує помітки на елементах графа, оперує з внутрішньою пам'яттю, в якій він будує представлення досліджуваного графа. Запропоновано новий метод, що базується на побудові на вершинах графу деякої нумерації. При цьому спочатку досліджуваний граф невідомий агенту. Нумерація реалізується помітками на елементах графа. Запропоновано метод розпізнавання графа на основі побудови M-нумерації на його вершинах. Знайдені верхні границі часової та ємнісної складності метода. Наведено приклади алгоритмів, що реалізують цей метод.

Рассматривается задача распознавания неизвестного графа мобильным агентом. Агент передвигается по графу, изменяет и считывает метки на элементах графа, оперирует с внутренней областью памяти, в которой он строит представление исследуемого графа. Предложен новый метод, основанный на построении на вершинах графа некоторой нумерации. При этом учитывается, что изначально исследуемый граф агенту не известен. Нумерация реализуется метками на элементах графа. Предложен метод распознавания графа на основании построения M-нумерации на его вершинах. Найдены верхние оценки временной и емкостной сложности метода. Приведены примеры алгоритмов, реализующих данный метод.

Graphs recognition problem by means an agent is considered. The agent moves on unknown graph, changes and reads marks of graph elements, operates with it's internal memory, in which it builds explored graph representations. The method based on some numeration of graphs vertices. Numeration realized by marks on graphs elements. Method graphs recognition based on M-numeration is proposed. Upper bounds of time complexity of the method is given.

Ключевые слова: распознавание графов, окрестность вершины, агент, метка.

Введение. Рассматривается задача распознавания конечного , неориентированного, связного графа, без петель и кратных ребер при помощи

агента, который перемещается по нему и изменяет метки на вершинах и ребрах графа [1]. Известен ряд подходов к решению данной задачи [2 – 5]. Данная работа посвящается анализу существующих алгоритмов и выявлению единого подхода к решению задачи распознавания графа и реализации «меток» на вершинах и инцидентора исследуемого графа. Особое внимание автор уделяет рассмотрению общего алгоритма распознавания графа, из которого небольшими модификациями могут быть получены уже известные.

Основные определения. В статье под графом G понимается конечный, неориентированный, связный граф, без петель и кратных ребер. $V(G)$ - множество вершин графа G, а $E(G)$ – множество его ребер. Мощности соответствующих множеств будем обозначать как $n = |V(G)|$, а $m = |E(G)|$, где $n \geq 1$, а $m \geq 0$. Инцидентором будем называть точку соприкосновения ребра и вершины. Если $e = (uv)$ – ребро графа, то пары $\{u,e\}$, $\{v,e\}$ – называются инцидентора ребра e. При этом говорят, что инцидентор $\{u,e\}$ примыкает к вершине u, а $\{v,e\}$ – примыкает к вершине v. I_v обозначим множество всех инцидентора $\{ve\}$, где $e = (xy)$, для которых выполняется либо $x = v$, либо $y = v$. Будем называть 1 – окрестностью вершины v множество, которое состоитиз множества I_v и множество всех вершин смежных с вершиной v. Это множество будем обозначать O_v. Элементом графа будем называть ту часть, которую агент может воспринимать, используя свои сенсорные возможности, и с которой он может проводить дополнительные манипуляции для изменения ее цвета. Примером элемента графа может служить вершина, ребро, инцидентор. Краской будем называть специальный ресурс, при помощи которого агент может изменить цвет элемента графа. Меткой вершины v будем называть совокупность красок на самой вершине и на элементах графа, которые входят в I_v. Через $\mu(v)$ и $\mu(ve)$ обозначим метку вершины v и метку инцидентора (ve) соответственно. В каждой конкретной задаче элемент графа, краска и метка будут определяться сенсорными возможностями агента и/или свойствами графа G. Будем называть s вершину, в которой в данный момент находится агент, а $\delta(se)$ – вершину u смежную с s через ребро $e = (su)$. Будем говорить, что граф G изоморфен графу H тогда и только тогда, когда существует сюръектив-

ное отображение $\varphi : V(G) \to V(H)$, для которого $(v,u) \in E(G)$ тогда и только тогда, когда $(\varphi(v), \varphi(u)) \in E(H)$. Нумерацией $F: V(G) \to N$ вершин графа G называется инъективное отображение множества его вершин во множество N натуральных чисел. Номером вершины v в нумерации F обозначается $F(v)$. M-нумерацией вершин графа называется нумерация вершин графа, соответствующая порядку их обхода при поиске в глубину. Древесными ребрами назовем ребра графа G порождающие дерево поиска при обходе графа в глубину. Все остальные ребра назовем обратными.

Все не определяемые понятия общеизвестны и их можно найти в [6].

Постановка задачи. Задан граф G неориентированный, конечный, связный, без петель и кратных ребер, вершины и инциденторы которого можно метить специальными красками. Задан агент, который может передвигаться из вершины в вершину неизвестного ему графа G по ребру, соединяющему их, воспринимать и анализировать некоторую локальную информацию об 1 – окрестности вершины, в которой он находится. Агент обладает ресурсом для изменения цветов элементов графа, для реализации меток вершин, конечной, но бесконечно наращиваемой памятью, в которую он будет записывать список ребер графа H. Изначально предполагается, что все элементы графа G не помечены. Агент помещается в произвольную вершину графа G. Необходимо разработать такой метод обхода графа G, раскраски его элементов, сбора информации о нем, чтобы можно было построить граф H изоморфный графу G с точностью до меток на вершинах графов.

Метод распознавания графов при помощи построения на его вершинах нумерации. Для того, чтобы построить граф H изоморфный графу G, с точностью до меток на вершинах графов, необходимо поставить каждой вершине v графа G в однозначное соответствие вершину v' из $V(H)$. После чего необходимо посетить все ребра графа G, так чтобы для каждого посещенного ребра определить каким вершинам оно инцидентно в графе G. Поскольку у нас установлено однозначное соответствие между множествами вершин графов H и G, то во множество ребер $E(H)$ можно будет добавить ребро, которое будет взаимно однозначно соответствовать посещенному ребру во множестве $E(G)$. Для этого пе-

ренумеруем, в произвольном прядке все вершины графа G, целыми числами $1,..,n$, которые будут соответствовать номерам вершин в графе H. Таким образом, каждой вершине графа G будет поставлена вершина графа H, и каждая вершина графа G будет иметь уникальную метку. Следовательно, при обходе ребер графа G агент для каждого ребра $e = (uv) \in E(G)$ сможет добавить в $E(H)$ ребро $e' = (\mu(u), \mu(v))$, которое будет однозначно соответствовать ребру $e \in E(G)$. Таким образом, между множествами вершин и ребер графов G и H будет взаимно однозначное соответствие, и будет построен граф H изоморфный графу G с точностью до меток на вершинах графов. Следовательно, для распознавания графа G необходимо построить на его вершинах некоторую нумерацию F, а затем обойти все его ребра.

«Алгоритм 1 ». Распознавание графа при помощи построения на нем некоторой F - нумерации.

Идея: Построить на вершинах графа G некоторую нумерацию F, которая будет соответствовать номерам вершин в графе H, затем пройти по всем ребрам e графа G и добавить в $E(H)$ ребра соответствующие им, используя для этого F -номера вершин графа G.

Вход: Граф G вершины, которого не помечены. Агент помещается в произвольную вершину графа G. $E(H) := \varnothing$.

Выход: Граф H изоморфный графу G с точностью до меток на вершинах графа; на вершинах графа G построена некоторая F -нумерация.

«Алгоритм 1»

Для простоты описания предположим, что все вершины графа G мы пронумеруем по порядку их вхождения во множество $V(G)$.

$i := 1;$
For each $v \in V(G)$ do
begin
$\mu(v) := i;$
$i := i + 1;$
end
For each $e = (uv) \in E(G)$ do
$E(H) := E(H) \cup \{e' = (\mu(u), \mu(v))\}$

Print EH);

Из всего вышесказанного следует справедливость следующих утверждений.

Утверждение 1. Для распознавания конечного, неориентированного, без петель и кратных ребер графа G, необходимо построит на его вершинах некоторую F-нумерацию и обойти все его ребра.

Замечание 1. Исходя из конкретной постановки задачи, оба цикла «Алгоритм 1» можно выполнять как последовательно, так и параллельно. В связи с этим метка на вершине графа G необходима до тех пор, пока все ребра из ее 1 – окрестности не будет добавлены в граф H. После этого метку можно удалить (если подобное допускается постановкой задачи) и повторно использовать для пометки другой вершины графа H.

Замечание 2. В силу того, что метку вершины агент может реализовывать при помощи комбинации красок на элементах графа G, то F-нумерация может быть реализована, как явно, путем присвоения вершине метки в виде целого числа, так и неявно, при помощи уникальных комбинаций красок на элементах графа G, которые , в свою очередь , можно перенумеровать. В силу замечания 1 этих уникальных наборов меток может быть меньшее число, чем вершин в исследуемом графе. Кроме того, агент может «вычислять» M – номер вершины, то есть совершать дополнительный обход и раскраску вершин и ребер графа G, для определения M – номера некоторой выбранной вершины.

Учитывая *Замечание 2*, переформулируем *Утверждение 1* следующим образом.

Утверждение 2. Для распознавания конечного, неориентированного, без петель и кратных ребер графа G, необходимо пометить каждую его вершину уникальной меткой и обойти все его ребра.

Сходимость алгоритма «Алгоритм 1» будет следовать из того, что мы рассматриваем конечный граф G, а значит, каждый цикл алгоритма будет выполнен за конечное число шагов.

Распознавание графа при помощи построения на его вершинах M – нумерации. Существует множество способов нумерации вершин графа [6]. Однако, в силу специфики постановки задачи (агент может перемещаться из вершины в вершину только по ребру, соединяющему их) желательно использовать нумерацию, для построения которой агент выполнит как можно меньшее количество переходов по ребрам графа G. Такому критерию соответствует M-нумерация.

Рассмотрим алгоритм, реализующий распознавание графа при помощи построения на его вершинах M-нумерации. Будем предполагать, что у агента имеется достаточно красок для реализации меток $1,..,n$.

«Алгоритм 2». Распознавание графа при помощи построения на нем M-нумерации.

Идея: Построить на вершинах графа G M-нумерацию, которая будет соответствовать номерам вершин в графе H, затем пройти по всем ребрам e графа G и добавить в $E(H)$ ребра соответствующие им, используя для этого M-номера вершин графа G.

Вход: Граф G, вершины которого не помечены. Агент помещается в произвольную вершину графа G. $E(H):=\varnothing$.

Выход: Граф H изоморфный графу G с точностью до меток на вершинах графов; на вершинах графа G построена M-нумерация.

Обход графа в глубину есть регулярный обход графа по правилам:

1). Находясь в вершине x, нужно двигаться в любую другую ранее не пройденную вершину, если таковая найдется, одновременно запоминая дугу, по которой впервые попали в нее.

2). Если из вершины x мы не можем пройти в ранее не пройденные вершину или таковой вообще нет, то мы возвращаемся в вершину z и продолжаем обход в глубину из нее.

Для простоты описания алгоритма будем предполагать, что вначале все вершины графа G имеют метку 0.

$i:=1;$
$\mu(s)=i;$
While $\exists e \in O_s : \mu(\delta(s,e))=0$ do
begin
$s:=\delta(s,e);$
$\mu(s,e):=r$
$i:=i+1;$
$\mu(s)=i;$
While $\neg(\exists e \in O_s : \mu(\delta(s,e))=0$ and $\exists e \in O_s : \mu(s,e)=r)$ do
begin
For each $e \in O_s$ do $E(H):=E(H)\cup\{e'=(\mu(s),\mu(\delta(s,e)))\}$
$s:=\delta(s,e)$: (s,e) $r;$
end;

end;

Утверждение 3. Для выполнения «Алгоритм 2» агенту потребуется $n+1$ метка.

Доказательство: В процессе выполнения алгоритма агент использует метки $1,..,n$, для построения M - нумерации на вершинах графа G, а так же метку r для того, что бы отметить инцидентор ребра, по которому агент впервые попал в текущую вершину. Следовательно, всего потребуется $n+1$, что и требовалось показать.

Утверждение 4. Для выполнения «Алгоритм 2» агенту потребуется $2m$ ячеек памяти.

Доказательство: Поскольку агент строит список ребер графа H, то ему необходимо будет хранить в памяти m ребер. Для каждого ребра хранится пара вершин, следовательно, для хранения каждого ребра потребуется две ячейки памяти. Всего потребуется $2m$ ячеек памяти, что и требовалось показать.

Будем предполагать, что за один шаг агент выполняет все или часть следующих действий: 1). воспринимает и анализируем метки из O_s; 2). переходит по ребру из вершины в вершину; 3) добавляет в граф H ребра соответствующие ребрам из O_s; 4). изменяет метку на вершине и / или на инциденторе.

Утверждение 5. Для выполнения «Алгоритм 2» агенту потребуется $2(n-1)$ шагов.

Доказательство: Будем предполагать, что наиболее трудоемким из 1) – 4) является переход из вершины в вершину по ребру соединяющему их. Поэтому подсчитаем количество переходов выполняемых агентом. В процессе обхода графа в глубину, агент проходит древесные ребра в прямом и обратном направлении. Следовательно, будет выполнено $2(n-1)$ переходов, что и требовалось показать.

Замечание 3. Поскольку процесс добавления в граф H ребер соответствующих ребрам из O_s может быть достаточно трудоемким, то будем предполагать, что добавление одного ребра в граф H будет выполняться за время θ. Тогда временная сложность «Алгоритм 2» будет равна $2(1)mm\ \theta$.

«Алгоритм 2» использует $n+1$ метку при достаточно больших, но конечных n, агенту потребуется иметь при себе достаточно большое количество различных красок, для реализации такого большого числа меток,

что на практике может быть трудно реализуемо. Поэтому на практике более предпочтительно использовать неявную нумерацию вершин. Рассмотрим «Алгоритм 3», реализующий распознавание графа при помощи построения на нем неявной M-нумерации. При добавлении каждого нового ребра в граф H, соответствующего ребру e в графе G происходит вычисления M-номера для вершин $\delta(s,e)$. Поскольку способ вычисления неявного M-номера будет зависеть от конкретной прикладной задачи, то введем формально в рассмотрение процедуру $Count(v)$, которая будет вычислять неявный M-номер для вершины v. Предположим, что эта процедура выполняется за время $\theta = f(n)$.

«Алгоритм 3». Распознавание графа при помощи построения на нем неявной M-нумерации.

Идея: Построить на вершинах графа G неявную M-нумерацию, которая будет соответствовать номерам вершин в графе H. Вновь посещенная вершина графа G метится меткой b, что означает, что вершина уже была ранее посещена. Затем пройти по всем ребрам $e = (u,v)$ графа G, выполнить для каждого из них процедуру $Count(x)$, где $x = v$, если $u = s$ или $x = v$ в противном случае и добавить в $E(H)$ ребра соответствующие им.

Вход: Граф G, вершины которого не помечены. Агент помещается в произвольную вершину графа G. $E(H) := \varnothing$.

Выход: Граф H изоморфный графу G с точностью до меток на вершинах графа, все вершины графа G помечены меткой b.

Для простоты описания алгоритма будем предполагать, что все вершины графа G имеют метку 0.

$i := 1$;
$\mu(s) := b$;
While $\exists e \in O_s : \mu(\delta(s,e)) \neq b$ do
begin
  $s := \delta(s,e)$;
  $\mu(s,e) := r$
  $\mu(s) := b$;
  While $\neg(\exists e \in O_s : \mu(\delta(s,e)) \neq b$ and $\exists e \in O_s : \mu(s,e) = r)$ do
  begin
    For each $e \in O_s$ do $E(H) := E(H) \cup \{e' = (\mu(s), Count(\delta(s,e)))\}$

```
    s := Out(se ): (se ) r;
  end;
end;
```

Легко видеть, что для выполнения «Алгоритм 3 » агенту потребуется 2 различные метки b и r.

Утверждение 6. Для выполнения «Алгоритм 3 » агенту потребуется $2(n-1)$ $f(n)m$ шагов.

Доказательство: Справедливость утверждения 6 будет следовать из замечания 3. Будем предполагать, что наиболее трудоемким из 1) – 4) является переход из вершины в вершину по ребру , соединяющему их. Поэтому подсчитаем количество переходов выполняемых агентом. В процессе обхода графа в глубину, агент проходит древесные ребра в прямом и обратном направлении. Следовательно, будет выполнено $2(n-1)$ переходов, что и требовалось показать.

Таким образом, использование неявной M-нумерации позволяет использовать число различных меток, а, соответственно, и красок, не зависящее от n, что ухудшает временную сложность на величину $f()nm$.

Далее рассмотрим алгоритмы реализующее распознавание графа при помощи построения на его вершинах неявной M-нумерации.

Алгоритмы, реализующие распознавание графов при помощи M-нумерации. Как уже говорилось ранее, для реализации номеров меток $1,..,n$ необязательно использовать только краски на вершинах. Можно использовать комбинации меток на элементах графа. Эти комбинации могут быть перенумерованы и , таким образом, позволят реализовывать метки $1,..,n$. Это позволит снизить количество различных красок. Кроме того, для реализации меток $1,..,n$ можно использовать временные уникальные метки. Использование комбинаций меток на элементах графа и временных уникальных меток позволят довести количество различных красок до некоторой , заранее известной константы. Рассмотрим алгоритмы, реализующие распознавание графов при помощи построения на них неявной M-нумерации.

Алгоритм, предложенный в [2].

Идея алгоритма: Алгоритм распознавания графа основан на методе обхода графа «в глубину». Агент идет в ранее не посещенные вершины и присваивает им номера $i+1$, где i максимальный из номеров меток текущей вершины и вершин смежных с ней. При этом агент сообщает ин-

формацию о метках вершин экспериментатору. На основе этой информации в граф H добавляется новая вершина и, исходя из информации, полученной из сообщения, соединяется ребрами с уже существующими вершинами. Это возможно, поскольку до тех пор, пока агент не совершит отход из текущей вершины, в вершину из которой он попал впервые, она имеет уникальную метку в пределах всего графа G. Метка уникальна в силу того, что переходы происходят из вершины в вершину и номер следующей вершины, не будет меньше номера предыдущей. Когда же агент попадает в вершину, из которой нельзя пройти в ранее не посещенную вершину, то он идет назад по вершинам $i-1$, где i – номер текущей вершины, до тех пор, пока не появится возможность идти в ранее не посещенную вершину. Такой отход возможен в силу того, что при проходе вперед, агент метит вершины $i+1$, где i максимальный номер из меток текущей вершины и вершин, смежных с ней, а, следовательно, обратный путь надо искать по меткам $\alpha-1$, где α – номер текущей вершины.

M-номерам соответствуют номера $1,2,..,l$, до тех пор, пока агент не попал в вершину, из которой нельзя попасть в вершину ранее не пройденную. После этого при отходе назад, на i шагов, где $0<i<l$ пока не появится вершина, из которой можно пройти в глубину графа, тогда M-номеру $l+1$, будет соответствовать метка $li+-1$, при этом агенту необходимо знать, что он отступил на i шагов назад при обходе в глубину. При переходе в следующую вершину и M-номеру $l+2$ будет соответствовать номер $li+-2$, при этом агенту необходимо знать, что он отступил на i шагов назад при проходе в глубину. И так далее. Когда агент передает данные об 1 – окрестности вершины, то это можно интерпретировать как переходы из текущей вершины с номером i в вершину с номером j. Таким образом, алгоритм, предложенный в работе [2], реализует распознавание графа при помощи построения на его вершинах M-нумерации и является частным случаем общего алгоритма распознавания графа при помощи построения на нем M-нумерации.

Алгоритм, предложенный в [8].

Идея алгоритма. Алгоритм основан на методе обхода графа «в глубину». Путь «назад» отмечается специальной краской r на инциденторе ребра при проходе в глубину. Метка вершины $1,..,m$. Метка каждой вершины уникальна до тех пор, пока агент не посетит все вершины смежные с ней. После чего краску снова можно использовать для уникальной пометки вершины. При невозможности идти вперед агент использует метки r, находит путь назад и идет по нему до тех пор, пока не сможет пройти в

ранее не пройденные вершины. Таким образом, некоторому M-номеру вершины в графе H всегда будет соответствовать уникальная метка вершины в графе G соответствующая наименьшему номеру «свободной краски». Таким образом, M-номеру будет соответствовать номер краски i, при этом агенту необходимо знать, сколько вершин смежных с вершиной помеченной i агент посетил.

Таким образом, алгоритм, предложенный в [8], реализует распознавание графа при помощи построения на его вершинах M-нумерации и является частным случаем общего алгоритма распознавания графа при помощи построения на нем M-нумерации.

Алгоритм, предложенный в работе [4].

Идея алгоритма: Алгоритм основан на методе обход графа «в глубину». Агент идет в глубину графа в ранее не посещенные вершины до тех пор, пока это возможно. По пути агент метит инциденторы ребер меткой r (путь назад). При отсутствии возможности идти вперед, агент исследует вершину. Для этого агент метит инциденторы не посещенных ребер меткой l. После чего агент начинает отход по ребрам, инциденторы которых помечены r, с целью поиска ребер, инциденторы которых помечены l. Если же нет не посещенных ребер, то агент переходит по ребру с меткой r на инциденторе, в вершину, из которой впервые попал в текущую вершину. Если же идти назад невозможно, то агент завершает работу.

Агент обходит граф G в глубину, во время обхода он добавляет в граф H переходы по древесным ребрам. При этом каждой вершине G ставится в соответствие (неявно) M-номер, равный количеству переходов, сделанных агентом из начальной вершины в текущую при проходе в глубину плюс количество переходов сделанных агентом при отходе назад по древесным ребрам (на данный момент времени). Переходы по обратным ребрам добавляются в граф H при исследовании вершины. После пометки инциденторов ребер исследуемой вершины метками l, агент вычисляет M-номера всех вершин, смежных с исследуемой. Этот номер будет равен максимальному M-номеру (на данный момент) минус количество переходов назад по ребрам, инциденторы которых помечены меткой r, с момента начала процедуры исследования вершины. Таким образом, M-номер любой вершины смежной с исследуемой вершиной будет определяться однозначно.

Таким образом, алгоритм, предложенный в [4], реализует распознавание графа при помощи построения на его вершинах M-нумерации и явля-

ется частным случаем общего алгоритма распознавания графа при помощи построения на нем M-нумерации.

Алгоритм, предложенный в [7].

Идея алгоритма. Алгоритм является модификацией алгоритма, предложенного в [4]. Основан на методе обхода графа «в глубину». Агент идет в глубину графа по ранее не посещенным вершинам. При этом агент присваивает каждой вершине неявный M-номер равный количеству переходов по ребрам в глубину графа G плюс количество переходов по древесным ребрам при отходе назад. При отсутствии возможности идти в глубину агент исследует вершину. Для этого он переходит по ребру из исследуемой вершины, которая имеет максимальный M-номер на данный момент времени, в вершину, у которой M-номер, которой необходимо определить. Для этого агент идет по ребрам, инциденторы которых помечены r, к исследуемой вершине и считает количество переходов p. Легко видеть, что максимальный M-номер (на данный момент) – p будет равен M-номеру вершины, в которую агент попал при исследовании ребра. Следовательно, агент может добавить в граф H ребро, соответствующее этому даному ребру в графе G. Подобные переходы агент выполняет для всех неисследованных ребер инцидентных исследуемой вершины.

Таким образом, алгоритм, предложенный в [7], реализует распознавание графа при помощи построения на его вершинах M-нумерации и является частным случаем общего алгоритма распознавания графа при помощи построения на нем M-нумерации.

Выводы. Предложены методы распознавания графов при помощи построения на них нумераций. Рассмотрены алгоритмы распознавания графов при помощи построения на них неявной M-нумерации. Найдены нижняя и верхняя оценки для времени выполнения алгоритмов распознавания, линейная и кубическая функции от n, соответственно. Подсчитано количество различных меток необходимых для реализации конкретных алгоритмов распознавания, при этом меньшим количество является две, а наибольшим n. Найдена верхняя оценка емкостной сложности для всех методов и алгоритмов – квадратичная функция от n. Рассмотрены алгоритмы, реализующие распознавание графа, при помощи построения на нем неявной M-нумерации. Рассмотрены способы замены явных M-номеров комбинациями меток на элементах исследуемого графа.

Библиографические ссылки

1. **Dudek G., Jenkin M.** Computational principles of mobile robotic − Cambridge Univ. Press. 2000 − 280 p.
2. **Грунский И.С.** О распознавании графов конечным автоматом. / И.С. Грунский, М.Ю. Тихончев // Вестник Донецкого университета. Серия А. Естественные науки − 2001, − вып.2. - С. 351-356.
3. **Grunsky I.S., Tatarinov E.A.,** Operating system recognation by automata. 9th International Conference «Pattern Recognition and Image Analysis: New Information Technologies» (PRIA-9-2008): Conference Proceedings. Vol. 1.-Nizhni Novgorod, 2008. − P. 197 − 200
4. **Грунский И.С.** Алгоритм распознавания графов / И.С. Грунский, Е.А. Татаринов // Тр. 4 междунар. конф. «Параллельные вычисления и задачи управления» PACO'2008, Москва, Ин -т Проблем Управления им. В. А. Трапезникова РАН, 2008 – С. 1483 − 1498.
5. **Грунский И.С.** Распознавание графов при помощи блуждающего по нему агента / И.С. Грунский, Е.А. Татаринов // Тезисы VIII Сибирской научной школы – семинара с международным участием «Компьютерная безопасность и криптография» - SIBECRIPT'09 (Омск, ОмГТУ, 8 – 11 сентября 2009 г.) – С. 96 − 98.
6. **Касьянов В.Н.** Графы в программировании, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев – СПБ., 2003. − 1104 с.
7. **Грунский И.С.** Распознавание конечного графа блуждающим по нему агентом / И.С. Грунский, Е.А. Татаринов Вестник Донецкого университета. Серия А. Естественные науки, 2009, вып. 1. – С. 492 − 497
8. **Татаринов Е. А.** Распознавание графа с помеченными вершинами / Е.А. Татаринов // Курс. работа, ДонНУ, мат. ф – т, 3 курс, 2005. – 20 с.

Надійшла до редколегії 12.01.10

УДК 519.2+616.345–007.272.1

©**В.М. Турчин[*], Я.С. Бондаренко[*], А.В. Гапонов[**], В.В. Гапонов[**]**
[*]Дніпропетровський національний університет імені Олеся Гончара
[**]Дніпропетровська державна медична академія

ЩОДО ВИЗНАЧЕННЯ ЙМОВІРНОСТЕЙ СТУПЕНІВ ТОВСТОКИШКОВОЇ НЕПРОХІДНОСТІ

Запропоновано методику оцінювання ймовірності декомпенсованого (субкомпенсованого) ступеня кишкової непрохідності залежно від вектора показників стану хворого, яка дає можливість хірургу дійти остаточного рішення щодо невідкладного оперативного втручання.

Предложена методика оценивания вероятности декомпенсированной (субкомпенсированной) степени кишечной непроходимости в зависимости от вектора показателей состояния больного, которая дает возможность хирургу принять окончательное решение относительно неотложного оперативного вмешательства.

In work the technique of estimate of probability decompensated (subcompensated) degree of intestinal impassability, which depend on a vector of parameters of a status of the patient, is offered. This technique will enable the surgeon to accept the final decisions concerning urgent operative intervention.

Ключові слова: товстокишкова непрохідність, ступінь непрохідності, показник, імовірність, критерій, незалежність, логістична функція, функція ризику, дециль ризику.

Вступ. Товста кишка є одним з найбільш розповсюджених місць локалізації злоякісного процесу. Незважаючи на можливості сучасних діагностичних методів, кількість ускладнених форм раку товстої кишки залишається високою. За структурою ускладненого раку товстої кишки провідне місце займає товстокишкова непрохідність [2;5]. Розрізняють три ступеня товстокишкової непрохідності [1]. *Компенсований ступінь* – це ранній ступінь непрохідності. *Субкомпенсований ступінь* характеризується частковим порушенням проходження вмісту травного каналу, зумовленим пухлиною. *Декомпенсований ступінь* характеризується повним порушенням проходження вмісту травного каналу внаслідок пухлини.

Наявність декомпенсованого ступеня непрохідності є показанням до невідкладного хірургічного втручання. Наявність субкомпенсованої або компенсованої товстокишкової непрохідності дозволяє належним чином підготувати хворого до оперативного втручання, а головне, дає можливість одночасно провести додаткові обстеження (маючи дані цих обстежень планування характеру майбутнього оперативного втручання стає більш детальним) [1].

Розподіл кишкової непрохідності на компенсовану, субкомпенсовану та декомпенсовану, з клінічної точки зору, є досить складним і становить собою значні труднощі при документальному підтвердженні. Такий поділ, з огляду доказової медицини, підтверджувався в основному не на клініко-інструментальному обстеженні, а більш візуально під час самої операції [4; 7].

Постановка задачі. Існують певні уявлення в якісному відношенні безумовно правильні, але в кількісному вельми неоднозначні, щодо диференціації ступеню кишкової непрохідності у хворих з гострою обструктивною непрохідністю товстої кишки, а саме – диференціації компенсованого ступеню непрохідності (відповідну групу хворих позначатимемо через А), субкомпенсованого ступеню непрохідності (група хворих В) та декомпенсованого ступеню непрохідності (групу хворих С).

Тому виникає задача знаходження показників, по значенням яких можна було б диференціювати стан хворого як декомпенсований (субкомпенсований) ступінь непрохідності або як компенсований ступінь непрохідності. Такими показниками можуть виступати показники загального аналізу крові, індекси реактивності та інтоксикації крові тощо.

У роботі запропоновано статистичний критерій диференціації декомпенсованого (субкомпенсованого) ступеня кишкової непрохідності залежно від вектора показників стану хворого.

Модель логістичної регресії. Позначимо ймовірність події «ступінь кишкової непрохідності є декомпенсованим» через p, тоді ймовірність події «ступінь кишкової непрохідності є компенсованим» дорівнює $1-p$.

Нехай при цьому щільність розподілу ймовірностей вектора показників $\mathbf{x}=(x_1,...,x_k)$ при декомпенсованому ступеню непрохідності – $f_1(x_1,...,x_k)$, а при компенсованому ступеню – $f_0(x_1,...,x_k)$. Тоді за формулою Байєса ймовірність $P(x_1,...,x_k)$ того, що хворий з вектором пока-

зників $\mathbf{x} = (x_1,...,x_k)$ має декомпенсований ступінь кишкової непрохідності (ступінь С) дорівнює

$$P(x_1,...,x_k) = \frac{pf_1(x_1,...,x_k)}{pf_1(x_1,...,x_k) + (1-p)\,f_0(x_1,...,x_k)} =$$

$$= 1\Big/\left[1 + \frac{1-p}{p}\,\frac{f_0(x_1,...,x_k)}{f_1(x_1,...,x_k)}\right]. \qquad (1)$$

Будемо виходити з припущення, що вектор показників $\mathbf{x} = (x_1,...,x_k)$ розподілений нормально зі щільністю

$$f_0(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \mathbf{C}}} \exp\left\{-\frac{(\mathbf{C}^{-1}(\mathbf{x}-a_0),\mathbf{x}-a_0)}{2}\right\}$$

для компенсованого ступеню кишкової непрохідності й

$$f_1(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \mathbf{C}}} \exp\left\{-\frac{(\mathbf{C}^{-1}(\mathbf{x}-a_1),\mathbf{x}-a_1)}{2}\right\}$$

для декомпенсованого ступеню кишкової непрохідності. Матриця коваріацій $\mathbf{C}$ одна й та сама. Тоді

$$\frac{1-p}{p}\,\frac{f_0(x_1,...,x_k)}{f_1(x_1,...,x_k)} = \exp\left\{-\left(\alpha_0 + \sum_{i=1}^{k}\alpha_i x_i\right)\right\}, \qquad (2)$$

де коефіцієнти α_0 та α_i подані через $p, a_0, a_1, \mathbf{C}^{-1}$.

Таким чином, отримано ймовірність $P(x_1,...,x_k)$ віднесення хворого з вектором показників $\mathbf{x} = (x_1,...,x_k)$ до групи С

$$P(x_1,...,x_k) = 1\Big/\left(1 + \exp\left(-\alpha_0 - \sum_{i=1}^{k}\alpha_i x_i\right)\right). \qquad (3)$$

Імовірність $P(x_1,...,x_k)$ – функція від лінійної комбінації показників $\mathbf{x}=(x_1,...,x_k)$: $y=\alpha_0+\sum_{i=1}^{n}\alpha_i x_i$. Функцію $y=\alpha_0+\sum_{i=1}^{n}\alpha_i x_i$ називатимемо *функцією ризику по значенням показників* $x_1,...,x_k$.

Оцінки невідомих коефіцієнтів α_0,α_i, $i=1,...,k$, знаходяться за методом максимальної правдоподібності

Визначення показників тяжкості стану хворого. Досліджуються 289 хворих: 102 належить до групи А, 103 – до групи В, 84 – до групи С. Кожному хворому ставиться у відповідність низка показників (характеристик), які описують його стан. Виходячи з досвіду хірургів, які працюють у сфері невідкладної абдомінальної хірургії, показниками, що можуть характеризувати ступінь тяжкості стану хворого були вибрані: *вік хворого* ($x_в$) та параметри загального аналізу крові: *гемоглобін* ($x_{гем}$), *еритроцити* ($x_{ер}$), *кольоровий показник* ($x_{кол}$), *гематокрит* ($x_{гемат}$), *лейкоцити* ($x_{лейк}$), *швидкість осідання еритроцитів* ($x_{шое}$), *еозинофіли* ($x_{еозин}$), *нейтрофіли паличкоядерні* ($x_{н/п}$), *нейтрофіли сегментоядерні* ($x_{н/с}$), *лімфоцити* ($x_{лімф}$), *моноцити* ($x_{мон}$).

При дослідженні перерахованих показників виявилося, що показники $x_{лімф}$ і $x_{лейк}$, $x_{лімф}$ і $x_{н/п}$, $x_{лімф}$ і $x_{н/с}$, $x_{гем}$ і $x_{ер}$, $x_{гем}$ і $x_{гемат}$ не є незалежними (гіпотези про незалежність показників перевірялися за допомогою критерію χ^2 незалежності ознак на рівні значущості 0,05). Тому показники $x_{лімф}$ та $x_{гем}$ виключимо з дослідження.

Незалежні показники загального аналізу крові хворих та вік хворих є нормально розподіленими. Гіпотези про нормальний розподіл показників перевірено за допомогою критерію χ^2 (гіпотетичний розподіл залежить від невідомих параметрів) на рівні значущості 0,05. Гіпотези про рівність середніх значень параметрів перевірено за допомогою критерію Стьюдента на рівні значущості 0,1 (див. табл. 1–3).

Для хворих груп А та С була виявлена суттєва відмінність таких показників: вік, лейкоцити, нейтрофіли паличкоядерні та сегментоядерні.

Таблиця 1

Результати перевірки гіпотези про рівність середніх значень показників хворих у ступенях компенсації та декомпенсації

Показник	Статистика $\|t\|$	Значення $t_{0,05;n+m-2}$	Гіпотеза $H_0 : a_\xi = a_\eta$
$x_в$	2,29	1,65	відхиляється
$x_{лейк}$	5,80	1,65	відхиляється
$x_{н/п}$	6,96	1,65	відхиляється
$x_{н/с}$	2,33	1,65	відхиляється

Для хворих груп А та В була виявлена суттєва відмінність таких показників: вік, лейкоцити, нейтрофіли паличкоядерні.

Таблиця 2

Результати перевірки гіпотези про рівність середніх значень показників хворих у ступенях компенсації та субкомпенсації

Показник	Статистика $\|t\|$	Значення $t_{0,05;n+m-2}$	Гіпотеза $H_0 : a_\xi = a_\eta$
$x_в$	1,82	1,65	відхиляється
$x_{лейк}$	1,93	1,65	відхиляється
$x_{н/п}$	3,68	1,65	відхиляється

Для хворих груп В та С була виявлена суттєва відмінність таких факторів: лейкоцити, нейтрофіли паличкоядерні та сегментоядерні.

Таблиця 3

Результати перевірки гіпотези про рівність середніх значень показників хворих у ступенях субкомпенсації та декомпенсації

Показник	Статистика $\|t\|$	Значення $t_{0,05;n+m-2}$	Гіпотеза $H_0 : a_\xi = a_\eta$
$x_{лейк}$	3,60	1,65	відхиляється
$x_{н/п}$	3,74	1,65	відхиляється
$x_{н/с}$	2,14	1,65	відхиляється

Статистичний критерій диференціації компенсованого та декомпенсованого ступеню кишкової непрохідності. Вихідними даними були показники 102 хворих групи А і 84 хворих групи С. З них дані по 28 хво-

рим були відібрані для контролю результатів, інші – 158 (серед них 87 належать групі А, 71 – групі С) використані для побудови критерію диференціації компенсованого та декомпенсованого ступеню кишкової непрохідності.

Кожного хворого будемо характеризувати вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$.

Розглянемо диференціацію хворих групи А та групи С у такому сенсі. Спочатку встановимо зв'язок між значеннями показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ та ймовірністю диференціації у хворого ступеню С кишкової непрохідності. Будемо виходити з того, що ймовірність $P(y) = P(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ віднесення хворого з вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ до групи С є функцією

$$P(y) = P(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік}) = \frac{1}{1+\exp\{-y\}}$$

від лінійної комбінації

$$y = \beta_0 + \beta_1 x_{лейк} + \beta_2 x_{н/п} + \beta_3 x_{н/с} + \beta_4 x_{вік}$$

показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$.

Коефіцієнти $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ оцінюються за вибіркою так, щоб залежність імовірності $P(y) = P(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ віднесення хворого до групи С від лінійної комбінації $y = \beta_0 + \beta_1 x_{лейк} + \beta_2 x_{н/п} + \beta_3 x_{н/с} + \beta_4 x_{вік}$ показників була, у певному сенсі, найкращою.

Оцінки параметрів $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$, отримані за вибіркою, позначимо через $\tilde{\beta}_0, \tilde{\beta}_1, \tilde{\beta}_2, \tilde{\beta}_3, \tilde{\beta}_4$. Функцію

$$\tilde{y} = \tilde{\beta}_0 + \tilde{\beta}_1 x_{лейк} + \tilde{\beta}_2 x_{н/п} + \tilde{\beta}_3 x_{н/с} + \tilde{\beta}_4 x_{вік} \qquad (4)$$

називатимемо функцією ризику за значеннями $x_{лейк}, x_{н/п}, x_{н/с}, x_{вік}$. У нашій задачі значення оцінок виявилися такими:

$$\tilde{\beta}_0 = -8{,}59,\ \tilde{\beta}_1 = 0{,}18,\ \tilde{\beta}_2 = 0{,}22,\ \tilde{\beta}_3 = 0{,}05,\ \tilde{\beta}_4 = 0{,}03,$$

а функція ризику набула вигляду

$$\tilde{y} = -8{,}59 + 0{,}18x_{лейк} + 0{,}22x_{н/п} + 0{,}05x_{н/с} + 0{,}03x_{вік}. \quad (5)$$

Імовірність віднесення хворого з вектором показників $(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ до групи С дорівнює

$$P(y) = P(x_{ëåéê}, x_{í/ï}, x_{í\ ñ}, x_{âîç}) = 1/(1+\exp\{8{,}59 - 0{,}18\ x_{ëåéê} - 0{,}22x_{í/ï} - 0{,}05x_{í\ ñ} - 0{,}03x_{âîç}\}). \quad (6)$$

На рис. 1 зображено логістичну функцію. На графіку точками позначено значення функції, чисельно рівні ймовірностям $P(y) = P(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ віднесення хворого з вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ до групи С. Одиницею позначено ймовірності (6) для хворих групи С, нулем – для хворих групи А.

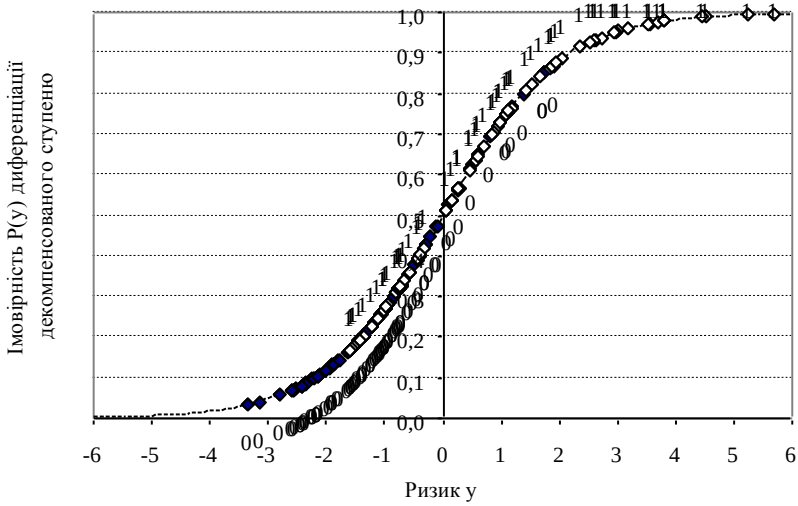


Рис. 1. Значення функції логістичної регресії

Очікувана та фактична кількість випадків віднесення хворого до групи С. Для з'ясування того, наскільки добре погоджується запропонована модель залежності ймовірності $P(y) = P(x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ від показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$, було впорядковано значення функції ризику (5) за величиною (від менших до більших) і всі отримані значення функції ризику (для всіх хворих) були розбиті на 10 рівних за чисельністю груп.

До першої групи ввійшли 10 % хворих, значення функції ризику яких найменші, до другої – 10 % хворих, що мали більші значення функції ризику і т. д. і в останню групу було включено 10 % хворих, значення функції ризику яких набуло найбільшого значення. Ці групи називають *децилями*.

Для кожної децилі обчислили очікуване число настання наслідку С (підсумовуючи для цієї групи значення $P(y) = 1/(1 + \exp\{-y\})$).

Дійсно, нехай імовірність події «i-тий хворий має декомпенсовану ступінь кишкової непрохідності» дорівнює p_i, тоді ймовірність події «i-тий хворий має компенсовану ступінь непрохідності» дорівнює $1 - p_i$. Індикатор події «хворий має декомпенсовану ступінь непрохідності» має розподіл $P\{I_i = 1\} = p_i$, $P\{I_i = 0\} = 1 - p_i$, $i = 1...n$. Математичне сподівання індикатора дорівнює $MI_i = p_i$. Тоді очікуване число настання наслідку С у хворих певної децилі дорівнює $M\sum_{i=1}^{n} I_i = \sum_{i=1}^{n} MI_i = \sum_{i=1}^{n} p_i$.

Очікувана кількість випадків віднесення хворого до групи С і фактична кількість випадків наведена в таблиці 4. Вони досить добре погоджуються.

Таблиця 4

Очікувана та фактична кількість випадків непрохідності декомпенсованого ступеня за децилями ризику

Дециль ризику	Кількість випадків віднесення хворих до групи С		Фактичний % хворих у групі С
	очікувана	фактична	
1	1	0	0.00
2	3	3	18.75
3	3	4	25.00
4	4	4	25.00
5	5	6	37.50
6	6	5	31.25
7	9	11	68.75
8	12	11	68.75
9	14	13	81.25
10	14	14	100.00
Разом	71	71	44.94

Запропонована модель (6) залежності ймовірності $P(y)$ віднесення хворого до групи С від показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ була перевірена на даних контрольної вибірки, в яку ввійшли 28 хворих (15 – з групи А, 13 – з групи С). Для кожного хворого відомі значення вектора показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ та ступінь кишкової непрохідності: компенсований (А) чи декомпенсований (С).

За вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік})$ кожного хворого контрольної групи обчислено значення функції ризику (за формулою (5)) та ймовірність віднесення хворого до групи С (за формулою (6)). Ці значення наведено в двох останніх стовпцях таблиці 5.

Наприклад, для хворого з вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік}) = (9{,}9; 1; 67; 43)$ ймовірність віднесення його до групи С становить 0,14, а для хворого з вектором показників $x = (x_{лейк}, x_{н/п}, x_{н/с}, x_{вік}) = (10{,}5; 23; 65; 79)$ ймовірність віднесення його до групи С становить 0,98.

Методика диференціювання хворих групи А та групи В, хворих групи В та групи С аналогічна.

Таблиця 5

Значення ймовірності віднесення хворих контрольної групи до групи С

Спостерігається ступінь непрохідності	$x_в$	$x_{лейк}$	$x_{н/п}$	$x_{н/с}$	Функція ризику y	Імовірність диференціації декомпенсов. ступеню $P(y)$
компенсований	67	4,5	12	42	−0,922	0,285
компенсований	**43**	**9,9**	**1**	**67**	**−1,816**	**0,140**
компенсований	64	4,5	5	64	−1,436	0,192
компенсований	70	13,5	9	63	1,237	0,775
компенсований	51	6	10	39	−1,732	0,150
компенсований	49	6,5	13	57	−0,126	0,469
компенсований	51	6,3	2	60	−2,368	0,086
компенсований	55	4	9	42	−2,044	0,115
компенсований	70	8	4	56	−1,232	0,226
компенсований	66	5,7	2	62	−1,915	0,128
компенсований	66	10	2	73	−0,563	0,363
компенсований	57	6,6	6	71	−0,689	0,334

Закінчення таблиці 5

компенсований	51	5.2	14	16	−2.169	0.103
компенсований	50	10.2	11	54	−0.006	0.499
компенсований	56	9.3	20	41	1.332	0.791
декомпенсований	76	10.8	14	60	1.872	0.867
декомпенсований	78	5.2	6	52	−1.265	0.220
декомпенсований	77	9.6	7	73	0.803	0.691
декомпенсований	68	10.7	9	61	0.558	0.636
декомпенсований	**79**	**10.5**	**23**	**65**	**4.143**	**0.984**
декомпенсований	69	10.5	15	75	2.583	0.930
декомпенсований	53	9.6	17	60	1.601	0.832
декомпенсований	69	4.9	14	38	−0.550	0.366
декомпенсований	78	5.1	5	59	−1.148	0.241
декомпенсований	61	14	13	73	2.440	0.920
декомпенсований	56	5.6	32	47	3.596	0.973
декомпенсований	69	8.5	8	77	0.777	0.685
декомпенсований	80	7.7	10	74	1.256	0.778

Висновки. В роботі розроблена методика оцінювання ймовірності віднесення хворого до групи пацієнтів з декомпенсованим (субкомпенсованим) ступенем кишкової непрохідності залежно від вектора показників стану хворого, яка дає можливість хірургу дійти остаточного рішення щодо невідкладного оперативного втручання.

Знайдені показники, за якими можна диференціювати ступені кишкової непрохідності, ними виявилися: лейкоцити, нейтрофіли паличкоядерні, нейтрофіли сегментоядерні, вік хворого.

Оцінку якості запропонованих моделей логістичних регресій планується провести методами ROC_аналізу.

Бібліографічні посилання

1. **Бондарь Г.В.** Современные аспекты лечения рака толстой кишки, осложненного непроходимостью кишечника. Ч. 1. Классификация, хирургическая тактика, результаты лечения / Г.В. Бондарь, В.Х. Башеев, Г.Г. Псарас, и др. // Клінічна хірургія. – 2000. – №8. – С. 48-49.

2. **Дарвин В.В.** Лечение больных с осложнениями злокачественных опухолей ободочной кишки / В.В. Дарвин, А.Я. Ильканич, С.В. Онишенко, Н.В. Климова // Хирургия. – 2007. – №6. – С. 8-12.

3. **Ивченко Г.И.** Математическая статистика: Учеб. пособие для втузов / Г.И. Ивченко, Ю.И. Медведев – М., 1984. – 248 с.

4. **Ковальчук В.С.** Лечение больных с острой непроходимостью толстой кишки опухолевого генеза в отделении общехирургического профиля / В.С. Ковальчук, В.Г. Васильченко, М.А. Койко // Клінічна хірургія. – 2000. – №2. – С. 46-48.

5. **Помазки В.И.** Тактика оперативного лечения при опухолевой обтурационной толстокишечной непроходимости / В.И. Помазкин, Ю.В. Мансуров // Хирургия. – 2008. – №9. – С.15-18.

6. Прикладная статистика: Классификация и снижение размерности: Справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин; под ред. С.А. Айвазяна. – М., 1989. – 607 с.

7. **Топузов Э.Г.** Диагностика и лечение острой кишечной непроходимости при раке толстой кишки / Э.Г. Топузов // Вестник хирургии. – 1989. – №12. – С. 76-78.

8. **Турчин В.Н.** Теория вероятностей и математическая статистика. Учебник. Д., 2008. – 656 с.

9. **Юнкеров В.И**. Математико -статистическая обработка данных медицинских исследований / В.И. Юнкеров, С.Г. Григорьев – СПб., 2002. – 266 с.

Надійшла до редколегії 12.01.10

УДК 532.516

И.С. Тонкошкур
Днепропетровский национальный университет имени Олеся Гончара

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕЧЕНИЯ ЖИДКОЙ ПЛЕНКИ ПО КОНИЧЕСКОЙ ПОВЕРХНОСТИ

Розглянута задача про просторову безхвильову плівочну течію нелінійно - в'язкої рідини по поверхні твердого тіла під дією сили тяжіння. За допомогою методу малого параметра одержано наближений розв'язок рівнянь динаміки рідкої плівки вздовж конічної поверхні.

Рассмотрена задача о пространственном безволновом пленочном течении нелинейно-вязкой жидкости по поверхности твердого тела под действием силы тяжести. С помощью метода малого параметра получено приближенное решение уравнений динамики жидкой пленки вдоль конической поверхности.

The problem of spatial waveless film flow of non-linear viscous fluid on a solid surface under the influence of gravity is considered. Using the method of small parameter an approximate solution of the dynamics of liquid film along the conical surface is obtained.

Ключевые слова: течение жидкой пленки, конус, поверхностное натяжение, сила тяжести.

Введение. Пленочные течения жидкости широко применяются в тепломассообменных аппаратах, в производстве кислот и резинотехнических изделий, при нанесении лакокрасочных и полимерных покрытий на различные поверхности и в других технологических процессах. При этом часто используют неньютоновские жидкости, имеющие особые свойства. В связи с этим представляет интерес разработка методов расчета течений реологически сложных сред.

Моделированию течений жидкой пленки по поверхности твердого тела посвящен ряд теоретических исследований (например, [1 ;5;6]), но при этом число работ в которых исследуются трехмерные течения довольно ограничено [2]. В [4] рассматривалась задача о пространственном течении пленки около тел вращения. В настоящей работе предлагается приближенная методика расчета движения нелинейно -вязкой жидкости по поверхности конических тел с некруговыми поперечными сечениями .

Постановка задачи. Рассматривается ламинарное течение жидкой пленки по конической поверхности твердого тела. Предполагается, что ось конуса расположена под некоторым углом γ к вертикали, а пленка стекает от его вершины вниз. На жидкость действуют силы тяжести и поверхностного натяжения.

Введем криволинейную ортогональную систему координат (x_1, x_2, x_3), связанную с поверхностью тела: координата x_1 отсчитывается от вершины конуса вдоль образующей, x_2 – полярный угол в плоскости, перпендикулярной оси тела, x_3 – расстояние по нормали к поверхности. Уравнение поверхности тела в сферической системе координат (r, θ, φ) задается в виде

$$\theta = \theta(\varphi),$$

где θ – угол между образующей и осью конуса.

Для описания течения жидкой пленки принимается модель вязкой несжимаемой жидкости, основанная на уравнениях импульса и неразрывности. В векторной форме эти уравнения имеют вид

$$\rho \frac{d\bar{V}}{dt} = -\,grad p + Div\bar{\bar{\tau}} + \rho \bar{g},$$

$$div\bar{V} = 0. \qquad (1)$$

Здесь $\bar{V}$ – вектор скорости движения жидкости, p – давление, ρ – плотность жидкости, $\bar{\bar{\tau}}$ – тензор вязких напряжений, $\bar{g}$ – интенсивность силы тяжести.

В качестве граничных условий используются условие прилипания на поверхности твердого тела, а также условия непрерывности напряжений и нормальной составляющей вектора скорости – на поверхности Г, разделяющей жидкость и газ

$$\bar{V} = 0 \quad \text{при} \quad x_3 = 0, \qquad (2)$$

$$\bar{\bar{\tau}}\bar{N} - (p - p_0)\bar{N} = 2\sigma H\bar{N} + \nabla_{\Gamma}\sigma, \qquad (3)$$

$$\bar{V}\bar{N} = W_N \quad \text{при} \quad x_3 = h. \qquad (4)$$

Здесь $h = h(x_1, x_2)$ – уравнение свободной поверхности Г, $\bar{N} = \bar{N}(x_1, x_2)$ – единичная нормаль к Г, H – средняя кривизна поверхности Г, σ – коэффициент поверхностного натяжения, $\nabla_{\Gamma}\sigma$ – поверхностный градиент коэффициента σ, p_0 – атмосферное давление в газе, W_N – ско-

рость перемещения поверхности Г в направлении нормали $\bar{N}$. Для замыкания системы уравнений (1)–(4) используется степенной реологический закон [6]:

$$\tau_{ij} = 2k(2I_2)^{\frac{n-1}{2}} e_{ij}, \qquad (5)$$

где k, n постоянные величины, I_2 – второй инвариант тензора скоростей деформаций.

Компоненты тензора скоростей деформаций определяются по формуле

$$2e_{ij} = \frac{1}{h_j}\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{h_i \partial x_i} + \delta_{ij}\sum_{k=1}^{3}\frac{V_k}{h_k}\frac{\ln h_i}{\partial x_k} -$$

$$-\frac{1}{h_i h_j}\left(V_i\frac{\partial h_i}{\partial x_j} + V_j\frac{\partial h_j}{\partial x_i}\right),$$

где δ_{ij} – символы Кронекера; V_1, V_2, V_3 – составляющие вектора скорости по направлениям x_1, x_2, x_3; h_1, h_2, h_3 – коэффициенты Ламэ.

Метод решения. Для упрощения системы дифференциальных уравнений (1) с граничными условиями (2)–(4) используется метод малого параметра, в качестве которого выбрана относительная толщина пленки $\varepsilon = h_0/l_0$ (h_0, l_0 – характерные поперечный и продольный размеры).

Введем безразмерные переменные по формулам:

$$x_1 = l_0\xi, \quad x_2 = \eta, \quad x_3 = \varepsilon l_0\zeta,$$

$$V_1 = U_0 u, \quad V_2 = U_0 w, \quad V_3 = \varepsilon U_0 v,$$

$$p = \rho U_0^2 p'', \quad h = \varepsilon l_0 F, \quad \sigma = \sigma_0\sigma.$$

Здесь U_0, σ_0 – характерные значения скорости движения жидкости и коэффициента поверхностного натяжения, $F = F(\xi, \eta)$ – уравнение свободной поверхности. Введем также вспомогательную функцию $P(\xi, \eta, \zeta)$, связанную с давлением p соотношением

$$p = p_{00} - 2\sigma\rho H + U^2 p.$$

Представим неизвестные функции (составляющие вектора скорости и давление) в виде разложений в ряд по ε

$$A = A^{01} + \varepsilon A \qquad (6)$$

и подставим их в систему полученных уравнений.

В работе рассматриваются течения жидкой пленки, для которых обобщенное число Рейнольдса $Re = \rho U_0^{2-n} h^n / k$ имеет порядок единицы, т.е. $\varepsilon Re \sim 1$. Учитывая главные члены разложений, получим упрощенную систему уравнений (для стационарного течения)

$$\frac{\partial u}{\partial \xi} + \frac{1}{\xi \psi(\eta)} \frac{\partial w}{\partial \eta} + \frac{\partial v}{\partial \zeta} + \bar{\zeta} u = 0,$$

$$\frac{\partial}{\partial \zeta}\left\{\left[\left(\frac{\partial u}{\partial \zeta}\right)^2 + \left(\frac{\partial w}{\partial \zeta}\right)^2\right]^{\frac{n-1}{2}} \frac{\partial u}{\partial \zeta}\right\} = -\frac{Re}{Fr}(\bar{e}\bar{e}_1),$$

$$\frac{\partial}{\partial \zeta}\left\{\left[\left(\frac{\partial u}{\partial \zeta}\right)^2 + \left(\frac{\partial w}{\partial \zeta}\right)^2\right]^{\frac{n-1}{2}} \frac{\partial w}{\partial \zeta}\right\} = -\frac{Re}{Fr}(\bar{e}\bar{e}_2), \tag{7}$$

$$\frac{\partial P}{\partial \zeta} = \frac{(\bar{e}\bar{n})}{Fr}.$$

Граничные условия: при

$$u = w = v = 0,$$

при $\zeta = F$

$$\frac{\partial u}{\partial \zeta} = \frac{\partial w}{\partial \eta} = P = 0, \quad v - u\frac{\partial F}{\partial \xi} - \frac{w}{\xi \psi(\eta)} \frac{\partial F}{\partial \eta} = 0. \tag{8}$$

Здесь $Fr = U_{00}^2/(gh)$ – число Фруда, $\bar{e}$ – единичный вектор, задающий направление действия силы тяжести, $\bar{e}_1, \bar{e}_2, \bar{n}$ – базисные векторы криволинейной системы координат (ξ, η, ζ), g – ускорение свободного падения.

Решение системы уравнений (7)-(8) может быть получено в виде

$$u = -\frac{n}{n+1} \varphi_1 B F^{\frac{n+1}{n}} \left[1 - \left(1 - \frac{\zeta}{F}\right)^{\frac{n+1}{n}}\right],$$

$$w = -\frac{n}{n+1}\varphi_2 B F^{\frac{n+1}{n}}\left[1-\left(1-\frac{\zeta}{F}\right)^{\frac{n+1}{n}}\right].$$

(9)

$$v = \left(\varphi_1\frac{\partial F}{\partial \xi}+\frac{\varphi_2}{\xi\psi(\eta)}\frac{\partial F}{\partial \eta}\right)BF^{\frac{n+1}{n}}\left\{\frac{\zeta}{F}+\frac{n}{n+1}\left[\left(1-\frac{\zeta}{F}\right)^{\frac{n+1}{n}}-1\right]\right\}+$$

$$+\frac{n}{n+1}\frac{\varphi_3}{\xi}\frac{\partial F}{\partial \eta}F^{\frac{2n+1}{n}}\left\{\frac{\zeta}{F}+\frac{n}{2n+1}\left[\left(1-\frac{\zeta}{F}\right)^{\frac{2n+1}{n}}-1\right]\right\}.$$

$$P = -\varphi_4\left(1-\frac{\zeta}{F}\right).$$

Где

$$\varphi_1(\eta) = -\frac{Re}{Fr}(\bar{e}\bar{e}_1),\quad \varphi_2(\eta) = -\frac{Re}{Fr}(\bar{e}\bar{e}_2),\quad \varphi_4(\eta) = \frac{(\bar{e}\bar{n})}{Fr},$$

$$\varphi_{11}(\eta) = \frac{nB}{2n+1}\left(\varphi_1' + \frac{\varphi_2'}{\psi(\eta)} + \frac{\ }{\psi(\eta)}\right)',$$

$$B(\eta) = (\varphi_{12}^2+\varphi^2)^{\frac{1-n}{2n}},\quad \psi(\eta) = (\theta'^2+\sin^2\theta)^{0.5}.$$

Подставив выражения (9) в последнее граничное условие (8), получим следующую начально-краевую задачу для определения неизвестной толщины жидкой пленки F

$$\varphi_1\frac{\partial F}{\partial \xi}+\frac{\varphi_2}{\xi\psi(\eta)}\frac{\partial F}{\partial \eta}+\frac{\varphi_3}{\xi}F = 0.$$

(10)

$$F(\xi_0,\eta) = 1,\qquad (\xi,\eta_p) = 0.$$

Система уравнений (10) решается численно с использованием разностной схемы бегущего счета [3]. Функция $F_0(\xi)$ находится в результате решения дифференциального уравнения на линии растекания $\eta=\eta_p$, на которой $\varphi_2 = 0$.

Анализ полученных результатов. По описанной методике проведены расчеты течения жидкой пленки по поверхности конических тел с поперечным сечением в виде круга, эллипса и многоугольника со скругленными кромками. Форма эллиптического конуса задается отношением полуосей эллипса δ и полууглом раствора конуса в плоскости большой полуоси θ_k, а пирамидального тела – числом боковых граней m, полууглом раствора конуса θ_k, вписанного в пирамиду r, и радиусом скругления r в плоскости поперечного сечения.

Результаты расчетов для тел с углом θ_k = 30º при значениях параметров Re=1, Fr=1 представлены на рис.1 -3. На рис.1 показаны профили скорости в жидком слое на линии растекания η=0 кругового конуса при угле скоса потока γ=10^0 и различных значениях координаты ξ (рис. 1.а, n=1) и параметра нелинейности вязкой жидкости n (рис. 1.б, ξ=2). Распределения толщины пленки F в продольных ($б$) и поперечных ($а$) сечениях эллиптического конуса приведены на рис. 2. (δ=2, n=1, γ=15^0).

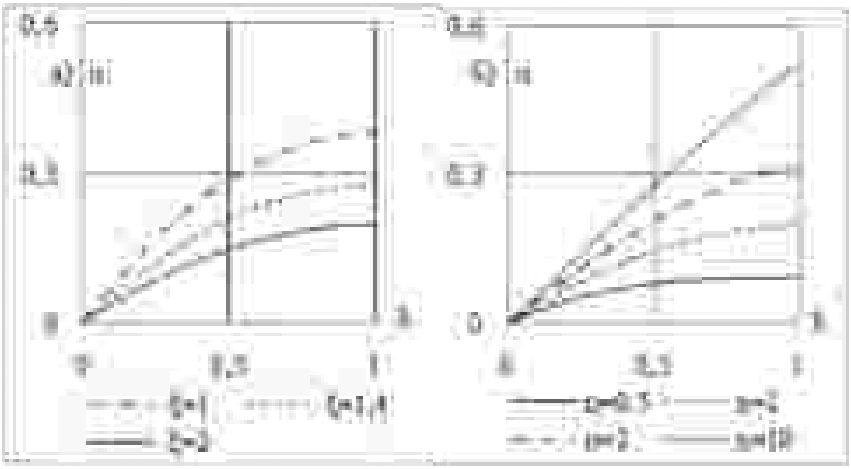


Рис. 1. Профили скорости

На рис. 3. а представлены зависимости толщины F от угла η для четырех тел с различными поперечными сечениями (γ=0^0, ξ= 2, r=0.5, δ =2). На остальных рисунках показано влияние на эти зависимости радиуса скругления r пирамидального тела (γ=0^0, m=3), коэффициента эллиптичности δ конуса (γ=15^0) и угла скоса потока γ (m=3, r=0.5).

Результаты расчетов показывают, что геометрические и физические параметры могут существенно влиять на распределение толщины пленки по поверхности тела. В частности, увеличение угла скоса потока γ, коэффициента эллиптичности δ, а также уменьшение радиуса скругления r усиливают неравномерность распределения толщины пленки в поперечном сечении тела.

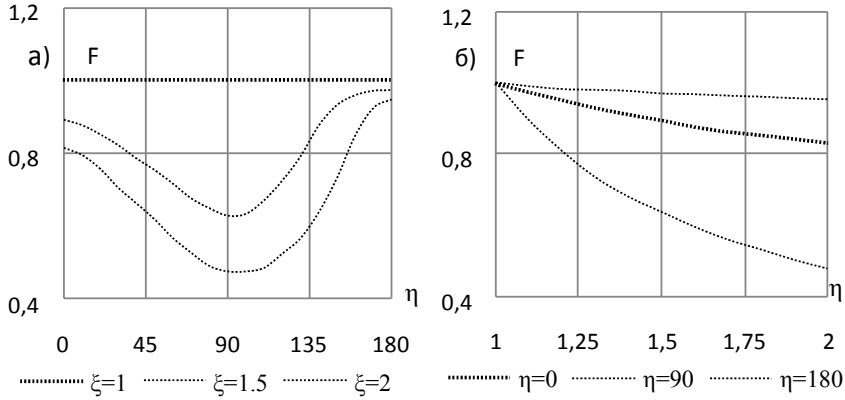
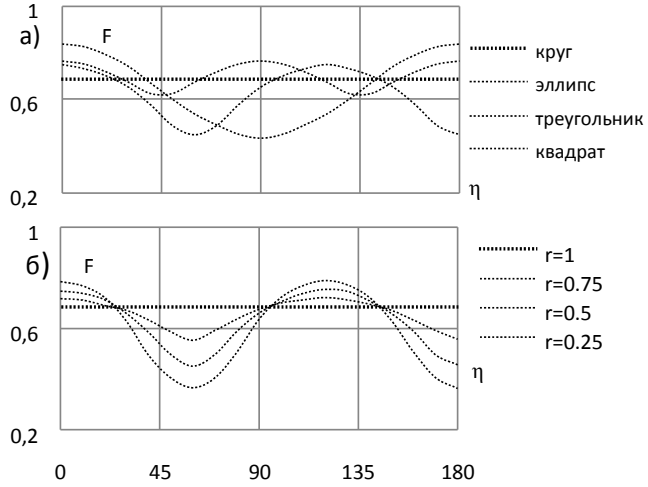


Рис. 2. Распределение толщины пленки F в сечениях эллиптического конуса



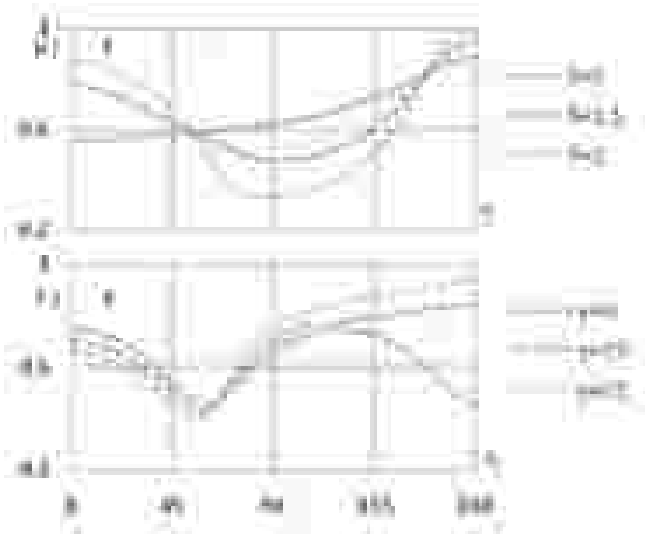


Рис. 3 .Влияние геометрических параметров на распределение толщины пленки в поперечном сечении конуса

Выводы. С помощью метода малого параметра разработана методика решения задачи о течении нелинейно -вязкой жидкости по конической поверхности под действием силы тяжести. Приведены примеры численного решения задачи для тел, не обладающих осевой симметрией , которые подтверждают эффективность предложеной методики и возможность ее применения в инженерной практике.

Библиографические ссылки

1. **Бояджиев Х.** Массоперенос в движущихся пленках жидкости / Х. Бояджиев, В. Бешков. – М., 1988.
2. **Волченко Ю.А.** Динамика жидкой пленки на искривленной твердой стенке с учетом фазового превращения на свободной поверхности / Ю.А. Волченко, И.И. Иевлев // Механика жидкости и газа. – 1994. – №4. – С. 42–50.
3. **Калиткин Н.Н.** Численные методы. – М., 1978.
4. **Тонкошкур И.С.** Моделирование течения жидкой пленки по поверхности тела вращения / И.С. Тонкошкур // Питання прикладної математики і математичного моделювання. –Д., 2003. – С. 178–183.
5. **Холпанов Л.П.** Гидродинамика и тепломассообмен с поверхностью раздела / Л.П. Холпанов, В.Я. Шкадов. – М., 1990.
6. **Шульман З.П.** Реодинамика и тепломассоомен в пленочных течениях / З.П. Шульман, В.Н. Байков. – Минск, 1979.

Надійшла до редколегії 18.06.10

УДК 519.6

©**В. А. Турчина*, О. Ю. Лебідь**, Є. В. Козаченко***
**Дніпропетровський національний університет імені Олеся Гончара*
***Академія митної служби України*

МЕТОД ОЦІНЮВАННЯ ЯКОСТІ ІНТЕРФЕЙСУ КОРИСТУВАЧА СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ

Розроблено метод оцінки якості інтерфейсу користувача систем дистанційного навчання, що враховує фізіологічні та психологічні аспекти сприйняття візуальної інформації людиною. Розроблено програмний продукт, яким можна вільно користуватися для оцінки якості власних інтерфейсів користувача.

Разработан метод оценки качества пользовательского интерфейса систем дистанционного обучения, учитывающий физиологические и психологические аспекты восприятия визуальной информации человеком. Разработан программный продукт, которым можно свободно пользоваться для оценки качества собственных интерфейсов пользователя.

A method for assessing the quality of the user interface of distance learning systems, taking into account the physiological and psychological aspects of perception of visual information man is elaborated. The software product which can be freely used for assessing the quality of their user interfaces is developed.

Ключові слова: інтерфейс користувача, система дистанційного навчання, фіксації погляду.

Постановка проблеми. Системи дистанційного навчання стають все більш популярними в умовах невпинного росту кількості Інтернет користувачів, рівня урбанізації та необхідності отримання якісної освіти.

Сьогодні за допомогою систем дистанційного навчання проходять навчання мільйони людей по всьому світу. Для будь якого вищого навчального закладу одним з показників його високого рівня стає наявність системи дистанційного навчання, яку можна оцінити за якістю знань, що вона надає та по якості інтерфейсу системи, що впливає на засвоєння інформації.

Отже, якість інтерфейсу користувача системи дистанційної освіти є важливим фактором, що впливає на якість системи дистанційної освіти в цілому.

Під час створення системи дистанційної освіти розробник приділяє багато уваги питанням технологій, даних, розробки моделі системи, що навчає тощо, проте дуже рідко звертає увагу на те, що інтерфейс системи сильно впливає на якість та швидкість засвоєння інформації. Тому розробка методу, що дозволяє оцінити якість інтерфейсу користувача є дуже актуальною задачею не тільки для розробників систем дистанційної освіти.

Окрім розробки методу необхідно розробити систему оцінки якості інтерфейсу користувача, яка могла б давати відповідь на питання щодо якості розробленого інтерфейсу. Зважаючи на ріст чисельності користувачів Інтернет та збільшення якості Інтернет послуг в Україні доцільно розробити Інтернет_сервіс, який був би доступним і міг швидко надавати розробникам послуги по оцінюванню якості інтерфейсу користувача.

Аналіз досліджень та публікацій. Поява eye_tracking пов'язана з дослідженнями руху погляду, починаючи з 1879 року, коли Луїс Еміль Жаваль зробив спостереження, що при читанні не відбувається гладкого ковзання очами по тексту, як передбачалося раніше, а відбувається серія коротких зупинок, названих фіксаціями, і швидкий стрибкоподібний рух очей [2]. Інформацію про навколишній світ ми візуально сприймаємо тільки через фіксації. Це відкриття порушило важливі питання про процес читання, які досліджувалися на початку 1900_х років.

З тих пір і дотепер вчені намагалися визначити як впливають рухи очей на зір у цілому та на сприйняття зображень [1].

У 1950 р. Альфред Ярбус написав про взаємозв'язок між фіксаціями погляду й зацікавленістю людини в певному об'єкті. Рухи погляду відображають людські процеси мислення, тому думка респондента випливає, певною мірою, із записів рухів погляду. Тобто можна легко визначити, які елементи привертають увагу спостерігача й, отже, його думки, в якому порядку і як часто.

До появи сучасної комп'ютерної техніки вченим доводилося досліджувати рухи очей на піддослідних (наприклад, дослідження Edmund Huey). З появою сучасних комп'ютерних технологій ай_трекінг вийшов на новий рівень. Тепер поведінку піддослідних моделює комп'ютер. Проте, ці методи мають досить високу алгоритмічну складність, і, крім того, не досить високу швидкість роботи.

У 1980 р. М. Джаст і П. Карпентер сформулювали важливу гіпотезу сили взаємодії погляду й думки, відповідно до якої немає істотних відмінностей між тим, на чому зафіксований погляд і тим, що обмірковує людина. Дана гіпотеза сьогодні дуже часто розглядається як основа для початку досліджень у сфері eye_tracking.

Сьогодні існують алгоритми комп'ютерного моделювання ай-трекінгу, тобто комп'ютерні програми, які можуть без використання спеціальної апаратури та без допомоги піддослідних змоделювати рухи очей, проте вони не можуть гарантувати точність. По-перше тому, що деякі з саккад є свідомими, по друге – тому, що не усі маркери уваги (точки фіксації) можна знайти за допомогою обчислювальної техніки та програм.

Іншим підходом до розв'язання задачі оцінювання інтерфейсу чи зображень у цілому є метод експертних оцінок. Очевидно, що такий підхід є занадто дорогим для такої задачі, бо для нього необхідна експертна група та приміщення і занадто довгим (при великих експертних групах процес оцінювання може займати до місяця).

Постановка задачі. Кожна людина сприймає візуальну інформацію, а значить і інтерфейс системи дистанційної навчання суб'єктивно, тому необхідно оволодіти знаннями щодо факторів, що впливають на сприйняття інформації людиною і розробити метод оцінювання інтерфейсу користувача з метою покращення сприйняття інформації учнями систем дистанційного навчання.

Eye-tracking дослідження, проведені Якобом Нільсеном [5], показують, що користувачі читають вміст, який нагадує форму F, а це означає, що читання починається з лівої верхньої частини веб-сторінки, і далі вона рухається трохи вниз, починаючи знову зліва (рис. 1).

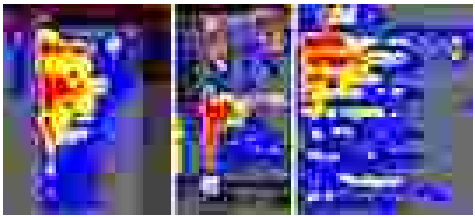


Рис. 1. Рух ока при читанні вмісту сайту [5]

Нільсен також заявляє про наслідки читання за схемою: користувачі не читатимуть вміст веб-сторінки по словах, вони витягнуть важливі параграфи, жирний текст, тощо; перші два параграфи повинні містити найбільш важливу інформацію; підрубрики і списки виділяються від регулярних параграфів.

Було досліджено, що інформація розміщена у пропорціях чисел Фібоначчі, «золотого перетину» чи на уявній синусоїді сприймається краще, ніж просто блок інформації через те, що за таких умов інформація набуває більш природного вигляду (рис. 2).

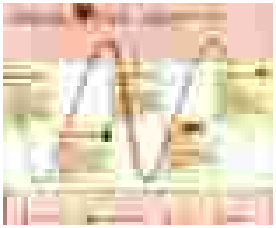


Рис. 2. Приклад розташування інформації на уявній синусоїді [3]

Розглянемо психологічні аспекти сприйняття. Уся інформація, що сприймається має або знаковий або аналоговий вираз. Сенсорні системи сприйняття людини утворюють внутрішню модель світу в 3 формах: звуках, образах і відчуттях.

У сенсорному сприйнятті знакової інформації існують дві системи – зорова і слухова. Механізми «дешифрування» знакової інформації можуть задіяти будь ‗яку систему сприйняття або всі три. Іншими словами, слово «яблуко» укладає до собе зміст форми, кольору, ваги, структури, смаку, запаху, звуку і таке ін.

Увага людини дивна та складна система, тому досі не існує обґрунтованої і точної математичної моделі уваги, а всі спроби хоч якось наблизитись до мети зводяться до моделювання фізіологічних процесів сприйняття, але людина – це не просто організм, це, у першу чергу, психологічний індивід, якому для сприйняття інформації необхідно звернути на неї увагу .

Шляхом лабораторних досліджень вченими було встановлено, що людина звертає увагу на деякі сектори зображення першочергово, а деякі не помічає зовсім. Те, що людина побачить першим може привернути її увагу і змінити хід її роботи, тобто важливо виявити ті місця, які людина побачить у першу чергу для того, щоб контролювати чи спробувати спрогнозувати дії, які зробить людина.

Сприйняття сайту має свої межі і знаходиться в залежності від особливостей механізму нервової системи людини.

Читання сайту відноситься до візуального сприйняття інформації. При цьому на те як людина помітить, прочитає, запам’ятає інформацію на сайті впливають відразу декілька факторів:

- кольорова гама сайту;
- завантаженість сторінки інформацією і графікою;
- кегль шрифту;
- контрастність тексту;

- геометрична «спрямованість» дизайну;
- наявність маркерів уваги.

Таким чином, на сприйняття інформації людиною впливає багато факторів, не залежно від того яке джерело. При цьому сприйняття – це суб'єктивний процес, тому оцінювати якість сприйняття зручніше засобами теорії нечітких множин та нечіткої логіки.

Розглянемо фізіологічні аспекти сприйняття. Оскільки людина сприймає сайт як візуальний об'єкт, то розглянемо лише ті інструменти, які відповідають за візуальне сприйняття, тобто зорову систему [4].

Зорова система починається з ока людини. Сітківка є аналогом фотосенсора фотоапарату, перероблюючи світло в набори електричних імпульсів та виконуючи первісну обробку зорової інформації, ще до «відправлення» її в зоровий відділ мозку.

На сітківці під двома шарами нервових клітин знаходяться зорові фоторецептори, які діляться на рецептори, що слабко залежать від ступеня освітлення (палички) – вони реагують на світло що надходить до ока, та колбочки, які реагують на більш сильне освітлення, їх більше, проте вони довше активізуються.

Розглянемо тільки колбочки. Колбочки відповідають за денний зір. Вони розділяються на 3 типи за кольорами, які сприймають (червоний, зелений і синій). Жовтий сприймається відразу двома видами колб, від того він значно більш інтенсивний, ніж синій. Через це генерується більш потужний електричний імпульс і жовтий нам здається більш яскравим і насиченим, ніж синій, на тому самому зображенні (цей факт перетинається з психологічним аспектом сприйняття описаним вище).

Усі фоторецептори об'єднуються у сектори, їх можна уявити як пікселі матриці фотоапарата. Ці сектори відповідають не лише за сприйняття інформації про колір, вони також виконують операції детекції руху та локалізують лінії на зображені.

У процесі сприйняття інформації сітківка, у першу чергу, оцінює яскравість сприйнятого кольору. Чим вище інтенсивність, тим більш яскравим здається колір. Однак при одній і тій же інтенсивності деякі кольори здаються більш яскравими.

Колірне коло Ньютона (рис. 3) відповідає максимальній насиченості кольору. Кольори розташовані так, щоб підкреслити деякі закономірності в їх сприйнятті. А саме, кожен колір має свій компліментарний колір, який займає в колірному колі діаметрально протилежну позицію. Суміш компліментарних кольорів, узятих за певним співвідношенням, утворює білий або сірий колір. Пари компліментарних кольорів є кольорами-антагоністами, оскільки вони анулюють вплив один одного на зорову сис-

тему. Приклади компліментарних кольорів: синій і жовтий, червоний і синьо_зелений, зелений і пурпурний, причому пурпурний лежить поза колірного кола, оскільки не має своєї колірної хвилі, а є відчуттям, що виникає при змішуванні спектрально_чистих кольорів.

Лабораторно доведено, що перепади яскравості та кольору сітківка сприймає виключно на границях перепаду. Цей факт добре демонструє ілюзія Маха (рис. 4).

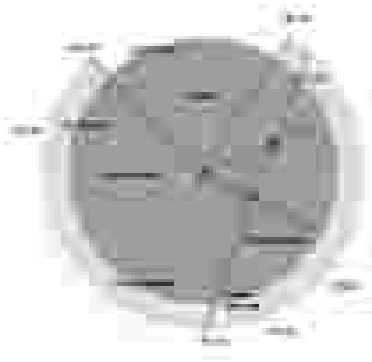


Рис. 3. Колірне коло Ньютона [4]

Границі між сусідніми по колірному колу кольори мають схильність сприйматися свідомістю як менш важливі, менш різкі.

Таким чином, ще до початку обробки зорової інформації мозок отримує від сітківки інформацію про лінії та контури на зображенні.

Рис. 4. Ілюзія Маха [4]

З властивостей роботи сітківки вченим вдалося вивести деякі закони сприйняття кольорів та яскравості: Вебера, Фехнера та Стівена.

Відносно чітке зображення око може сприйняти за 2.5 секунди (0,5 секунди на локалізацію і 2 секунди на точне налаштування) але при куті зору лише 10°. Можливість чітко бачити об’єкти, що знаходяться у більшому полі зору забезпечується завдяки зоровим саккадам, тобто рефлектор-

ним переривчастим рухам очей, які переводять погляд з одного об'єкта на інший.

Сітківка працює виключно завдяки саккадам, бо спочатку була спроектована на розпізнавання рухів. Саккади бувають свідомі (коли можна свідомо перевести погляд на об'єкт) та несвідомі (відбуваються рефлекторно): прослідкувати за об'єктом, що перемістився; периферійним зором помітити натяк на психологічний об'єкт (перетин з психологічним аспектом сприйняття); периферійним зором помітити візуально активний об'єкт (контрастні об'єкти, великі об'єкти).

Несвідомі саккади можна відловлювати, тобто змоделювати чи спрогнозувати. Очевидно, якщо на зображенні є людське обличчя, яке знаходиться у полі периферійного зору, то наступна саккада буде націлена на область з обличчям. Так само можна оцінити кольори: якщо на зображенні у полі периферійного зору є яскравий об'єкт на фоні компліментарного кольору, то саккада буде направлена на вивчення цього об'єкта.

Такий самий ефект відбувається і з контурами зображення. Так зображення з багатьма вигинами контуру притягує саккади, а значить і увагу, більше ніж зображення з точними геометричними контурами та прямими лініями. Крім того витіюватий контур має кращий психологічний ефект для запам'ятовування.

Очевидно, що психологічні та фізіологічні аспекти сприйняття тісно перетинаються між собою, тому для досконалої оцінки якості інтерфейсу ефективно було б оцінювати одразу два аспекти.

Зрозуміло, що на рухомі об'єкти (наприклад, анімовані картинки) користувач зверне увагу в першу чергу, тому виключимо їх із розгляду і обмежимося статичним зображенням.

Опишемо більш детально основні кроки реалізації методу.

Крок 1. Першим кроком методу є детекція переважаючого кольору сайту, оскільки, коли людина бачить сайт вперше, декілька секунд очі розфокусовані, вона не бачить деталей і контурів, вона бачить виключно кольори.

Для оцінки переважаючого кольору можна використовувати два підходи:

- «розмиття» зображення до отримання ефекту розфокусованих очей (наприклад, за допомогою фільтра Гауса);
- сильна пікселізація зображення з подальшим аналізом отриманих пікселів.

Слід зазначити, що для отримання придатного для оцінювання кольору розмиття необхідно використовувати фільтр Гауса з дуже великою дисперсією, що дає велике вікно фільтра та подовжує час виконання операції.

Тобто час роботи алгоритму великий, і тому він не придатний для використання.

Алгоритм пікселізації працює на декілька порядків швидше ніж алгоритм фільтра Гауса, що дозволяє застосувати його для первісної обробки зображення.

Як і в більшості алгоритмів обробки зображення, в алгоритмі пікселізації зображення рухаємо вікно обробки по зображенню. На відміну від методу Гауса, де рух вікна відбувається з кроком в один піксель, у даному методі щоразу зміщуємо вікно на відстань, яка дорівнює довжині вікна, що значно економить час. На кожному кроці необхідно обчислити середній колір вмісту вікна обробки. Для цього:

- в кожному пікселі вікна обробки отримаємо колір, тобто обчислюємо колір зображення в конкретній точці;
- розкладаємо отриманий колір на складові за гамою RGB;
- додаємо отримані компоненти кольору до суми усіх кольорів вікна обробки (також покомпонентно);
- після того, як пройдемо крізь усі пікселі вікна обробки обчислимо середні значення компонент кольору:

$$\rho = \frac{\sum_{i=0}^{n-1} R_i}{T}, \quad \gamma = \frac{\sum_{i=0}^{n-1} G_i}{T}, \quad \beta = \frac{\sum_{i=0}^{n-1} B_i}{T}, \qquad (1)$$

де ρ, γ, β – це середні значення кольору у вікні обробки покомпонентно відповідно червоного, зеленого та синього компонентів; T – загальна кількість оброблених пікселів у вікні; R_i, G_i, B_i – відповідні компоненти кольору кожного з пікселів вікна, n – кількість пікселів.

У результаті роботи алгоритму отримаємо середнє значення кольору зображення у вікні обробки. Тепер можна зафарбувати вікно обробки в отриманий колір, змістити його і перейти до обробки наступного блоку пікселів. У результаті отримаємо набір прямокутників пофарбованих у різні кольори (рис. 5).

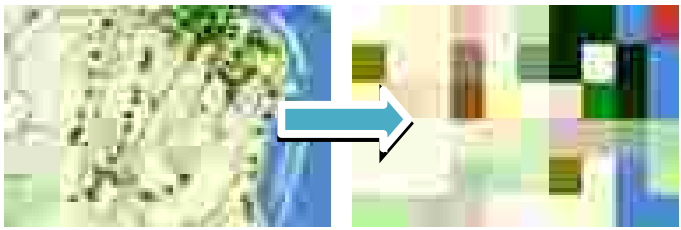


Рис. 5. Приклад роботи алгоритму пікселізації

Для реалізації було обрано розмір вікна обробки 100 x100 пікселів. Отримаємо n*10 квадратів, де n – результат ділення висоти зображення на 100. Обчислюємо середній колір усіх квадратів і визначаємо переважаючий колір на зображенні. Цей колір, у подальшому, можна проаналізувати і встановити чи сприятливий він для користувача.

Крок 2. Людські обличчя відносяться до психологічних атракторів уваги, тому в методі оцінки інтерфейсу користувача доцільно знайти та локалізувати обличчя для подальшого аналізу їх розташування на зображенні і оцінки впливу на розподіл уваги користувача.

Локалізація облич – це комп'ютерна технологія, що виявляє людські обличчя будь-якого розміру на зображенні, відокремлюючи риси обличчя від непотрібного шуму (дерев, будівель тощо) [6].

До методів локалізації облич відносяться: локалізація облич як задача класифікації шаблону, локалізація облич на контрольованому фоні, локалізація облич за кольором.

У даній роботі відбулась свідома відмова від використання методів локалізації облич, що базуються на нейронних мережах, оскільки такі мережі потребують великої навчальної вибірки і, крім того, мають досить складну структуру, через що дуже довго навчаються.

Алгоритм для локалізації облич базується на праці [8]:

- відшукати можливі області, де може бути обличчя, керуючись міркуваннями про колір шкіри;
- бінаризувати зображення в тих областях, де може знаходитись обличчя;
- зіставити отримані області із бінарним шаблоном облич, якщо шаблони співпадають, то маємо обличчя на зображенні.

Для реалізації алгоритму необхідно використати спосіб кодування RGB інформації – YC_BC_R. Фактичні кольори відображаються залежно від фактичного RGB і використовуються для відображення сигналу [10].

Для переведення зображення з кольорового простору RGB до YC_BC_R, необхідно обчислити наступні значення для кожного пікселя:

$$Y = K_R R + (1 - K_R - K_B)G + K_B B, \tag{2}$$

$$C_B = \frac{1}{2} \cdot \frac{B - Y}{1 - K_B}, \tag{3}$$

$$C_R = \frac{1}{2} \cdot \frac{R - Y}{1 - K_R}, \tag{4}$$

де $K_R = 0{,}114,\ K_B = 0{,}299$ – визначені константи.

Після конвертування зображення у YC_BC_R проходимо зображення, порівнюючи колір кожного пікселя з множиною порогових значень

$$Y = 0{,}5,\ C_B \in [-0{,}15; 0{,}05],\ C_R \in [0{,}05; 0{,}20]. \qquad (5)$$

Якщо колір пікселю потрапляє в описані межі, то відмічаємо його, як «підозрілий» і ставимо на зображенні білу точку, інакше ставимо чорну точку та отримуємо карту (рис. 6).

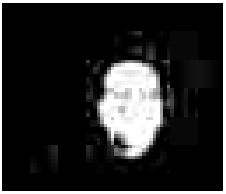


Рис. 6. Карта кольорів, що можуть бути обличчям [9]

Тепер необхідно прибрати зайві білі точки, шум, тобто точки, які не мають багатьох сусідніх. Для цього використовуємо алгоритм ерозії [11], у результаті роботи якого прибираємо з карти кольорів зайвий шум, але із зменшенням розмірів «підозрюваних» на обличчя областей, що може ускладнити роботи на наступному кроці. Для того, щоб повернути початкові розміри областей застосуємо метод математичної морфології метод розширення [12]. Основний ефект дії методу полягає в тому, що межі об'єктів розширюються, а незаповнені проміжки у них зменшуються.

Отримали зображення, відфільтроване за кольором шкіри. Необхідно сегментувати зображення для локалізації обличчя. Для цього потрібно знайти області на зображенні з найбільшим вмістом білих пікселів. Обчислимо суми білих пікселів за кожним стовпцем і за кожним рядком. Місця з максимальною кількістю білих пікселів за стовпцями та рядками одночасно і будуть центром обличчя.

Для того, щоб визначити межі обличчя використовуємо обчислені суми білих пікселів. Межі обличчя починаються при відхиленні від суми білих пікселів, що відповідають центру обличчя, на деяке порогове значення. Відсіюючи всі значення, менші за порогове, визначаємо межі обличчя.

Крок 3. Локалізація тексту на довільному зображенні є нетривіальною задачею, оскільки на відміну від облич текст має складнішу структуру і майже не піддається шаблонізації [7].

Текст є об'єктом з високим вмістом прямих ліній та гострих кутів, тому для того, щоб знайти області, які гіпотетично можуть містити текст слід обробити зображення за допомогою алгоритмів пошуку границь. У результаті отримаємо набір областей, «підозрюваних» на вміст тексту. Далі кожну з цих областей зможемо обробити специфічним алгоритмом детекції тексту.

Після локалізації тексту необхідно визначити його читабельність. Для цього необхідно знайти колір самого тексту та колір фону, на якому він розташований.

Після локалізації тексту та з'ясування кольорів літер та фону, кольори треба оцінити, для чого рекомендовано використовувати прості методи оцінки кольорової різниці, різниці яскравості та контрастності. Усі ці методи оцінюють два кольори як розкладання у спектр RGB.

У роботі використано метод локалізації тексту, що базується на методі CED [13] (англ. Color edge detection – детекція границі кольору) та виділення рядків тексту. Причини такого вибору: складна і невідома структура фону зображення, що буде оброблятися, яка відкидає методи пошуку, котрі базуються на кольорах; можлива низька контрастність тексту, що шукається, відкидаюча методи, які базуються на виявлені контрастних повторюваних структур; у більшості зображень, які оброблятимуться (сайтах) текст розташований консолідованими блоками по рядках, тому доцільно проводити класифікацію областей «підозрюваних» на текст за допомогою аналізу рядків.

CED-оператор є оператором виділення границь, який в обробці керується повним впливом колірного простору YIQ [14]. Сигнал I називається синфазним, Q – квадратурним. Конверсія в RGB і обернено здійснюється за наступними формулами:

$$R = Y + 0{,}956I + 0{,}623Q, \quad G = Y - 0{,}272I - 0{,}648Q,$$
$$B = Y - 1{,}105I + 0{,}705Q, \qquad (6)$$
$$Y = 0{,}299R + 0{,}587G + 0{,}115B, \quad I = 0{,}596R - 0{,}274G - 0{,}322B,$$
$$Q = 0{,}211R - 0{,}522G + 0{,}311B, \qquad (7)$$

де R, G, B – відповідають інтенсивності кольорів червоного, зеленого і синього; Y – складова якравості; I та Q – кольорорізницеві складові.

CED об'єднує вживані вище морфологічні оператори, але в комбінації із конвертацією зображення в YIQ, що дозволяє обробляти не тільки зображення у сірих відтінках, а й повно_кольорові зображення.

У подальших викладках використовується оператор Робертса [15].

Висока точність і здатність до видалення шумів – важливі вимоги для виявлення границь на кольорових зображеннях, так само, як і на сірих зо-

браженнях. Тут традиційний оператор Робертса перетворюється в CED, який використовує YIQ систему кольору. Оператор CED описується наступним чином

$$CED = \sqrt{\delta_1^2 \quad \delta_2^2}, \tag{8}$$

де δ_1 та δ_2 визначаються

$$\delta_1 = Dis(i, j, i+1, j+1), \tag{9}$$

$$\delta_2 = Dis(i+1, j, i, j+1), \tag{10}$$

де $Dis(\cdot)$ – евклідова норма двох пікселів на зображенні в YIQ колірній системі.

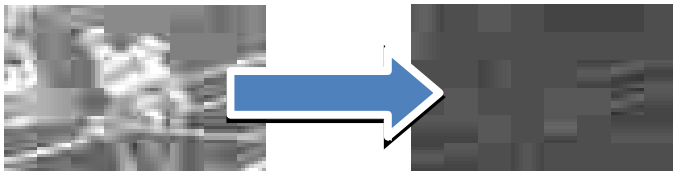


Рис. 7. Результат роботи оператора Робертса [15]

Постобробка важлива для сегментації тексту і фону в тих зображеннях, які обробили CED. Оскільки текстові лінії зазвичай горизонтальні, необхідно посилити горизонтальні краї зображення. Так, тут використовується оператор виділення границь, щоб виділити край зображення знову після того, як CED виділив вперше. Для цього використовується оператор Собеля [16].

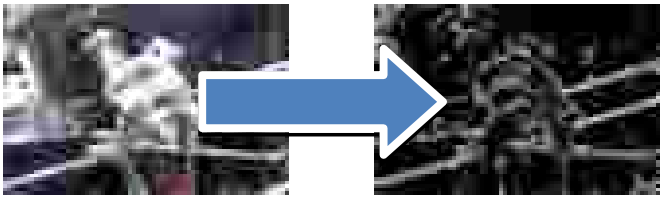


Рис. 8. Результат роботи оператора Собеля [16]

Таким чином, отримано бінарне зображення і можна локалізувати блоки, «підозрювані» на текст морфологічними методами.

Алгоритм для кроку локалізації тексту.

1. Вхідне зображення I_1 обробляється CED для отримання бінарного зображення I_2.

2. I_2 обробляється оператором Собеля і отримуємо бінарне зображення границь I_3.

3. I_3 обробляється морфологічними методами, і отримуємо бінарне зображення I_4 без шуму та з покращеними границями. Зважаючи на горизонтальну орієнтацію тексту рекомендовано використовувати відкриваючий, а потім закриваючий морфологічний оператор.

4. Після того, як обробка закінчена, слід застосувати важливі правила, щоб визначити деякі очевидні текстові блоки і видалити деякі очевидні нетекстові блоки. Для цього використовують як горизонтальні так і вертикальні проекції зображення I_4 та його щільність.

Позначимо через P_h – горизонтальну проекцію блоку зображення $m \times n$, а через P_v – вертикальну проекцію. Якщо не виконується нерівність

$$P_h > \mu_1 \text{ та } P_v > \mu_2, \qquad (11)$$

де μ_1, μ_2 – порогові константи, то блок класифікується нетекстовим.

Якщо щільність блоку $m \times n$ менша за іншу порогову константу μ_3, то блок класифікується як нетекстовий. Якщо щільність блоку $m \times n$ більша за μ_3 та виконується нерівність (1 1), то блок класифікується як текстовий.

Таким чином процес локалізації тексту завершено (рис. 9).

а) б) в)

Рис. 9. Етапи обробки зображення за CED: а) вхідне зображення; б) після обробки CED; в) після застосування морфологічних операторів

Маємо набір блоків ‑кандидатів на присутність тексту. Необхідно визначити чи є в цих блоках текст.

Щоб прибрати зайвий шум, отриманий у ході обробки зображення оператором Собеля (шум з'являється через неоднорідність фону) застосуємо до сегментів, оброблених оператором Собеля методи математичної морфології, такі як ерозія та розширення (рис. 10).

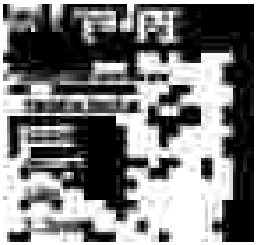


Рис. 10. Результат обробки сегментів оператором Собеля та математичною морфологією

Тепер можна виділити рядки на зображенні. Задача виділення рядків зводиться до знаходження верхніх і нижніх граней рядків тексту, зображеного на початковому зображенні [17].

Алгоритм сегментації рядків ґрунтується на тому, що середня яскравість у зображеннях міжрядкових проміжках істотно нижче за середню яскравість зображень текстових рядків.

Спочатку для усіх піксельних рядків початкового зображення знаходимо їх середні значення яскравості

$$s_j = s_j(B) = \frac{1}{n}\sum_{i=1}^{n} b_{ij}, \tag{12}$$

де B – матриця яскравості точок вигляду

$$B = \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{nm} \end{pmatrix}, \tag{13}$$

а b_{ij} – значення яскравості конкретної точки $0 \le b_{ij} \le b^{\max}$, $b^{\max}$ – максимально можлива яскравість.

Обраховуємо середнє значення яскравості для всього зображення

$$s(B) = \frac{1}{m}\sum_{j=1}^{m} s_j(B). \tag{14}$$

Середня яскравість у міжрядкових проміжках тексту має бути невелика (в ідеальному випадку вона дорівнює нулю). Тому яскравість верхньої межі текстового рядка можна виразити через середню яскравість зображення

$$s^t = k^t s(B), \tag{15}$$

де k^t – коефіцієнт, причому $0 \le k^t \le 1$.

Аналогічно яскравість нижньої межі текстового рядка, також може бути виражена через середню яскравість усього зображення

$$s_b = k_b s(B), \tag{16}$$

де k_b – коефіцієнт, причому $0 \le k_b \le 1$.

Робота алгоритму сегментації рядків полягає у послідовному перегляді масиву середніх значень ($s_1, \ldots, s_m$) і виявленні безлічі пар індексів (s_i^t s_{b_j}) піксельних рядків, що відповідають верхній s_i^t і нижній s_{b_j} граням зображення i-го рядка, що задовольняють наступним умовам.

Умови верхньої межі текстового рядка. Початок текстового рядка або області стійкого підвищення яскравості фіксується, якщо

$$\begin{aligned}(s_{j-2} < s^t) \& (s_{j-1} < s^t) \& (s_j > s_b) \& (s_{j+1} > s_b) \& \\ \& (s_{j+2} > s_b) \& (s_{j+3} > s_b)\end{aligned} \tag{17}$$

Умови нижньої межі текстового рядка. Кінець області стійкого підвищення яскравості визначається, якщо

$$((s_j > s^t) \& (s_{j+1} < s_b)) OR ((s_{j+1} < s_b) \& (s_{j+2} > s_b) \& (s_{j+3} > s_b)) \tag{18}$$

У результаті формується безліч пар індексів верхніх і нижніх меж рядків. Різниця між цими індексами дає висоти текстових рядків. Проте такий алгоритм знаходить середню висоту кожного текстового рядка і «зрізає» символи, виступаючі по висоті за цю середню висоту.

Щоб уникнути цього, необхідно розширити знайдені межі: серед знайдених текстових рядків визначається рядок з мінімальною висотою $H_{\min}$ і, потім усі межі з кожного боку розширюються на величину $0{,}3H_{\min}$. Це не призводить до злиття рядків, оскільки міжрядкові інтервали тексту, як правило, більше ніж висота рядка.

Ті сегменти, які задовольняють наведеним вище умовам класифікуються як текст, інші видаляються з розгляду.

Отримали сегменти зображення, які містять текст і постала задача об'єднання їх за належністю до однакових блоків тексту [18].

Для цього будемо керуватися простим алгоритмом.

1. Оберемо перший необроблений сегмент, позначимо його як А та додамо його до списку блоків. Позначимо сегмент, як оброблений.

2. Знайдемо усі прилеглі до нього сегменти, позначимо їх, як оброблені та розширимо блок А значеннями прилеглих сегментів.

3. Рекурсивно повторимо крок 2 для сегментів, прилеглих до блоку А.

4. Якщо прилеглих сегментів не має, то позначимо блок А закритим, та перейдемо до кроку 1.

У результаті роботи цього алгоритму отримаємо блоки тексту.

Окрім локалізації тексту нам необхідно оцінити його читабельність. Для цього нам потрібно визначити колір фону, на якому розміщено текст, та значення кольору самого тексту.

Перш за все для обчислення значення кольору нам необхідно визначити де на виділеному текстовому блоці знаходиться текст, а де фон. І тут знову нам знадобиться зображення отримане в результаті обробки сегментів_кандидатів оператором Собеля та математичною морфологією (ерозією та розширенням). До цього зображення застосуємо пошук за шаблоном бччбб, де через б позначено білий піксель, а через ч – чорний. Зрозуміло, що в результаті обробки текстового блока оператором Собеля, ерозіїю та розширенням отримаємо точні контури тексту чорного кольору, фон, на якому знаходиться текст – білого кольору, сама літера знаходиться між контурами, що були отримані в результаті обробки операторами виділення границь. Шаблон пошуку: один піксель фону, два пікселі контуру та два пікселі літери. Координати, що відповідають правій границі шаблону – внутрішня частина літери, а лівій – фону. На зображенні визначаємо значення кольорів фону та переднього плану за отриманими координатами.

Для оцінки читабельності тексту використовують 3 фактори [19]:

– різницю кольорів:

$$dC = \max(R_1, R_2) - \min(R_1, R_2) + \max(B_1, B_2) - \min(B_1, B_2) + \max(G_1, G_2) - \min(G_1, G_2) \quad (19)$$

де R_i, G_i, B_i – відповідні складові 1_го та 2_го кольорів у гамі RGB;

– яскравісну різність за кольоровим простором YC_BC_R:

$$dB_{r_1} = \left| B_{r_1} - B_{r_2} \right|, \quad (20)$$

$$B_{r_1} = 0{,}299R_1 + 0{,}587G_1 + 0{,}114B_1, \quad (21)$$

$$B_{r_2} = 0{,}299R_2 + 0{,}587G_2 + 0{,}114B_2, \quad (22)$$

де R_i, G_i, B_i – відповідні складові 1_го та 2_го кольорів у гамі RGB;

– освітленість:

$$L_1 = 0{,}2126\left(\frac{R_1}{255}\right)^{2,2} + 0{,}7152\left(\frac{G_1}{255}\right)^{2,2} + 0{,}0722\left(\frac{B_1}{255}\right)^{2,2}, \quad (23)$$

$$L_2 = 0{,}2126\left(\frac{R_2}{255}\right)^{2,2} + 0{,}7152\left(\frac{G_2}{255}\right)^{2,2} + 0{,}0722\left(\frac{B_2}{255}\right)^{2,2}. \quad (24)$$

Якщо $L_1 > L_2$, то $dL = \frac{L_1 + 0{,}05}{L_2 + 0{,}05}$. Якщо $L_2 > L_1$, то $dL = \frac{L_2 + 0{,}05}{L_1 + 0{,}05}$.

Після того, як отримано оцінки читабельності тексту їх слід порівняти з константами.

Якщо $dC > 500 \;\&\; dBr > 125 \;\&\; dL > 5,$ то текст вважають добре читабельним. Ці константи є емпіричними, і невеликим відхиленням значень оцінок від них у менший бік можна знехтувати.

Крок 4. Локалізація об'єктів з великим числом вигинів контуру. Окрім психологічно активних об'єктів «цікавими» для людського ока є і незвичайні об'єкти, які мають велику кількість вигинів контурів, тож у методі оцінки інтерфейсу користувача необхідно врахувати цей фактор.

Локалізація таких об'єктів є задачею досить схожою на задачу локалізації тексту, оскільки розв'язок тієї задачі також базується на виділенні контурів.

Крок 5. Геометрична направленість зображення має безпосередній вплив на психологічний стан користувача. Так, наприклад, зображення, яке більшістю об'єктів вказує вгору та вправо викликає у людини враження перспективності та прогресу, руху вперед, і, навпаки, зображення направлене вниз і вліво – з регресом та занепадом.

Крок 6. Отримані на попередніх кроках дані необхідно проаналізувати для того, щоб видати відповідь користувачу.

Кожен із блоків даних, отриманих на попередніх кроках, представляє собою набір із різнотипних не нормалізованих показників, таких, як, наприклад, позиції і розміри облич чи позиції, розміри та читабельність блоків тексту.

Оскільки інформації буде досить багато , доцільно аналізувати кожен із блоків даних, отриманих на різних кроках методу окремо, отримуючи окрему відповідь по кожному із блоків, а потім оцінювати відповіді по блокам.

Кожен із блоків даних, крім даних отриманих на першому кроці методу, має бути проаналізований наступним чином:

- перш за все слід виділити великі за розміром об'єкти (великі обличчя, великі блоки тексту), оскільки вони є впливовішими на сприйняття користувача;
- кожен із виділених об'єктів має бути проаналізованим щодо його розташування на зображенні (відомо, що найбільш впливовими на користувача є об'єкти розташовані у верхній та правій частинах зображення), тож виділимо тільки ті великі об'єкти, які «впливово» розташовані;
- тепер кожен із об'єктів має пройти відповідний аналіз (наприклад, для блоків із текстом слід оцінити їх читабельність).

У результаті отримаємо дані про впливовість кожного із блоків даних, отриманих у ході обробки зображення.

Представимо їх у вигляді лінгвістичних змінних, величин, множин та функцій належності [8]:

$$\varphi = \begin{cases} \varphi_1, \\ ..., \\ \varphi_n, \end{cases} \quad T = \begin{cases} T_1, \\ ..., \\ T_n, \end{cases} \quad \mu = \begin{cases} \mu_{\varphi_1}, \\ ..., \\ \mu_{\varphi_n}, \end{cases} \tag{25}$$

де φ – сім'я лінгвістичних величин, T – набір нечітких величин, а μ – сім'я функцій належності.

Описавши ці дані і сформувавши набір правил логічного висновку можна видати відповідь користувачу щодо якості досліджуваного інтерфейсу.

Розглянемо приклад зображення з текстом та обличчям та локалізуємо обличчя на цьому зображенні (рис. 11).

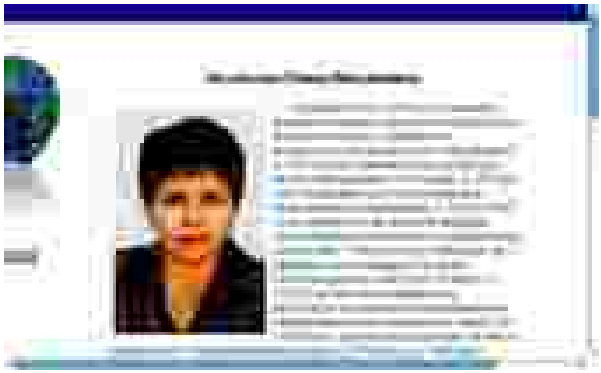


Рис. 11. Результат локалізації обличчя

Проаналізуємо знайдені обличчя на зображенні. У результаті процедури локалізації обличчя знайдено одне обличчя. Обличчя знаходиться у центрі зображення, отже притягне увагу людини першочергово, і оскільки обличчя знаходиться безпосередньо в тексті, то це розташування притягне увагу до тексту. Крім того, обличчя є досить великим (14 % у ширину і 13 % у висоту відносно усього зображення).

Для подальшого аналізу зображення, та оцінки читабельності тексту проведемо процедуру локалізації тексту на зображенні та оцінку його кольору і кольору фону (рис. 12).

Рис. 12. Результат локалізації тексту

Проаналізуємо читабельність тексту: $dC = 765 > 500$, $dB = 255 > 125$, $dL = 21 > 5$. Отже, маємо гарну читабельність тексту.

Висновки. Було розроблено метод оцінювання якості інтерфейсу користувача, який базується на оцінці психологічних та фізіологічних факторів, що впливають на сприйняття візуальної інформації людиною.

На основі розробленого методу створено програмний продукт, за допомогою якого можна проводити частину досліджень для отримання оцінки інтерфейсу користувача.

Програмний продукт має наступні функціональні можливості: проводити аналіз переважаючого кольору дистанційної системи, що навчає; локалізувати обличчя в інтерфейсі досліджуваної системи та оцінювати їх розташування і розміри відносно всього інтерфейсу; локалізувати текст в інтерфейсі досліджуваної системи і аналізувати його читабельність.

Надалі авторами планується повна реалізація запропонованого методу у вигляді програмного засобу для вільного користування.

Бібліографічні посилання

1. «Eye tracking» http://en.wikipedia.org/wiki/Eye_tracking
2. http://eyetracking.com.ua
3. «Applying Mathematics to Web Design» http://www.smashingmagazine.com/2010/02/09/applying-mathematics-to-web-design/
4. «Моделирование зрения» http://ymik.habrahabr.ru/blog/57679/
5. http://www.webdesignerdepot.com/2008/12/10-usability-tips-for-web-designers/
6. «Face detection» http://en.wikipedia.org/wiki/Face_detection

7. Datong Chen, Jean- Marc Odobez, Hervé Bourlard. Text detection and recognition in images and video frames. – IDIAP.: 2003. – 14 p.
8. **Ободан Н. І.** Моделі та методи систем штучного інтелекту. / Н. І. Ободан, В. А. Громов – Д., 2008. – 119 с.
9. http://blogs.msdn.com/b/coding4fun/archive/2010/03/24/9984015.aspx
10. http://en.wikipedia.org/wiki/YCbCr
11. http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm
12. http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm
13. Yan Hao, Zhang Yi, Hou Zeng-guang, Tan Min. Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network And CED. – Plzen, Czech Republic.: 2003. – 6 p.
14. http://en.wikipedia.org/wiki/YIQ
15. Davis LS. A survey of edge detection techniques. Computer Graphics and Image Processing, vol 4, no. 3, 1975, pp 248-260
16. http://ru.wikipedia.org/wiki/Оператор_Собеля
17. http://mechanoid.narod.ru/misc/segmentator/
18. **Хорн Б.** Психология машинного зрения. / Б. Хорн, М. Минский, Й. Сараи и др. – М., 1978. – 337 с.
19. **Марр Д.** Зрение. Информационный подход к изучению представления и обработки зрительных образов. / Д. Марр – М., 1987. – 374 с.

*Надійшла до редколегії 30.01.*10

УДК 519.8

©**В.А. Турчина, Н.К. Федоренко**
Дніпропетровський національний університет імені Олеся Гончара

НАБЛИЖЕНІ АЛГОРИТМИ ПОБУДОВИ ОПТИМАЛЬНИХ ПАРАЛЕЛЬНИХ УПОРЯДКУВАНЬ ЗАДАНОЇ ДОВЖИНИ

Запропоновано три наближені алгоритми поліноміальної складності розв'язання задачі побудови оптимального паралельного упорядкування мінімальної ширини та заданої довжини для класичної задачі паралельного упорядкування.

Предложены три приближенных алгоритма полиномиальной сложности для решения задачи построения оптимального параллельного упорядочения минимальной ширины и заданной длины для классической задачи параллельного упорядочения.

The article includes the description of three approximate algorithms of the polynomial complexity for creating schedules of the minimal width and given length for the classical scheduling problem.

Ключові слова: паралельне упорядкування, дискретна оптимізація, графи.

Вступ. На практиці, поряд із класичною задачею побудови розкладів виконання частково упорядкованих завдань за мінімальний час, існують задачі виконання всіх завдань за заданий час мінімальними ресурсами. Залежно від конкретних практичних задач, розмірність може бути дуже великою (наприклад, при розпаралеленні обчислень, вона може складати сотні тисяч одиничних завдань), а отже використання точних алгоритмів експоненційної складності для цих задач на практиці неможливе. Тому особливо актуальними є наближені алгоритми поліноміальної складності, що дозволяють за реальний час визначити порядок виконання завдань.

Постановка задачі. За заданим ациклічним графом G і заданій довжині упорядкування $l(S)$ побудувати паралельне упорядкування S мінімальної ширини $h(S)$.

Методи розв'язання . Розглянемо поставлену задачу ($S(G,l,h)$) за умови, що довжина упорядкування, що будується, дорівнює довжині критичного шляху орграфа ($l = l$). Очевидно, що цю умову завжди можна виконати, додавши до графа ланцюг необхідної довжини. Тоді перед нами стане наступна задача оптимізації. Окрім того, що буде мінімізуватися ширина упорядкування (кількість ресурсів у практичній задачі), довжина упорядкування буде мінімальною можливою. А отже, час виконання задач буде також мінімально можливим.

Як відомо, задача побудови упорядкування мінімальної ширини, як і двоїста до неї задача побудови упорядкування мінімальної довжини, відносяться до класу NP -складних задач. Запропоновані нижче алгоритми мають поліноміальну складність, але є наближеними для даної задачі.

У даних алгоритмах перед безпосереднім пошуком розв'язку, ми маємо, певним чином, оцінити ширину оптимального паралельного упорядкування перед початком розв'язання. Від того, наскільки вдала буде ця оцінка залежить час, за який ми знайдемо розв'язок. Адже, якщо оцінка буде занижена і ми будемо будувати упорядкування виходячи з цього, доведеться декілька разів повторювати алгоритм з початку, збільшуючи «теоретичну» ширину.

У розглянутих нижче алгоритмах ми будемо користуватися оцінкою, запропонованою в [1]. А саме:

$$h \geq \max\left(\left\lceil \frac{n}{l} \right\rceil, \tilde{h} + \left\lceil \max\left(0, \frac{\left|\hat{S}\right| - \sum_{i=1}^{r} (\tilde{h} - \left|\tilde{S}[i]\right|)}{r}\right) \right\rceil\right),$$

$$\text{äå } \tilde{S}[i] = \overline{S}[i] \cap S[i] (i = \overline{1,l}),\ \tilde{h} = h(\tilde{S}),$$

$$\hat{S}[i] = \overline{S}[i] \setminus \tilde{S}[i], r : \hat{S}[r] \neq \varnothing \quad \text{è } \forall i \in (r,l] : \hat{S}[i] = \varnothing.$$

Ця оцінка дозволяє оцінити ширину оптимального упорядкування із достатньо великою точністю.

Алгоритм 1 (Алгоритм заснований на побудові списку пріоритетів). Ідея алгоритму: алгоритм базується на відомому алгоритмі запропонованому Р.Л. Грехемом у [2], що передбачає наявність для графа певного списку пріоритетів, який однозначно дозволяє визначити порядок додання вершин до паралельного упорядкування, що будується. У розглянутому у статті випадку заданого списку пріоритетів немає, а отже нам необхідно

побудувати його самостійно. Причому побудувати так, щоб, послідовно обираючи вершини з цього списку, можна було побудувати оптимальне упорядкування заданої довжини та мінімальної ширини.

Таким чином, запропонований алгоритм складається із двох кроків:

1. Для вершин будується список пріоритетів за певними правилами.
2. За алгоритмом Грехема будується паралельне упорядкування за шириною отриманою за оцінкою.
3. Якщо оптимальне упорядкування побудувати неможливо, робимо висновок, що оцінка ширини була занижена. Збільшуємо «теоретичне» значення ширини на один та повторюємо побудову оптимального паралельного упорядкування.

Побудова списку пріоритетів. Нехай $L = \{v_{i_1}, v_{i_2}, \cdots, v_{i_n}\}$ – це список пріоритетів, що задає порядок додання вершин до оптимального упорядкування, що будується. Будемо казати, що вершина, яка знаходиться у списку пріоритетів лівіше має більш високий пріоритет, тобто , має бути додана до упорядкування раніше. Введемо такі позначки:

- $L[v_i]$ – номер місця, яке займає вершина v_i у списку пріоритетів L;
- $\underline{\mu}_i$ – номер місця, яке займає вершина v_i в упорядкуванні $\underline{S}$ (упорядкування, в якому кожна вершина займає крайнє ліве можливе місце);
- $\overline{\mu}_i$ – номер місця, яке займає вершина v_i в упорядкуванні $\overline{S}$ (упорядкування, в якому кожна вершина займає крайнє праве можливе місце).
- $\mu_i = \left\{ m_{ij} : m_{ij} = \overline{\underline{\mu}_i, \overline{\mu}_i} \right\}$ – множина місць, які вершина v_i може займати в упорядкуванні S довжини l.

Будуємо список пріоритетів так, щоб виконувалися наступні умови:

1. Якщо $\overline{\mu}_i < \overline{\mu}_j$, то $L[v_i] < L[v_j]$ (тобто вершина для якої крайнє праве допустиме місце менше, має мати більший пріоритет).
2. Якщо $\overline{\mu}_i = \overline{\mu}_j$ і $|\mu_i| < |\mu_j|$, то $L[v_i] < L[v_j]$ (тобто із двох вершин, для яких крайнє праве допустиме місце однакове, більший пріоритет матиме та, яка має меншу кількість допустимих місць).
3. Якщо $\overline{\mu}_i = \overline{\mu}_j$ і $|\mu_i| = |\mu_j|$ і кількість дуг, що виходять із v_i, більше за кількість дуг, що виходять із v_j, то $L[v_i] < L[v_j]$.

4. Якщо $\overline{\mu}_i = \overline{\mu}_j$ і $|\mu_i| = |\mu_j|$ і кількість дуг, що виходять із v_i та v_j, однакова, але кількість дуг, що входять до v_i, менше за кількість дуг, що входять у v_j, то $L[v_i] < L[v_j]$.

5. Інакше вершини v_i та v_j можуть бути розташовані у списку пріоритетів у будь-якому порядку.

Розглянемо приклад роботи алгоритму.

Нехай задано граф G (рис. 1). Розв'яжемо для цього графу задачу $S(G, l, h)$ за умови, що $l = l$.

Побудуємо для графу упорядкування S та $\overline{S}$.

$$S = \left\{ \begin{matrix} 1 & 4 & 7 & 9 \\ 2 & 5 & 15 & 8 \\ 3 & 6 & & \\ 10 & 12 & & \\ 11 & 13 & & \\ 14 & & & \end{matrix} \right\},$$

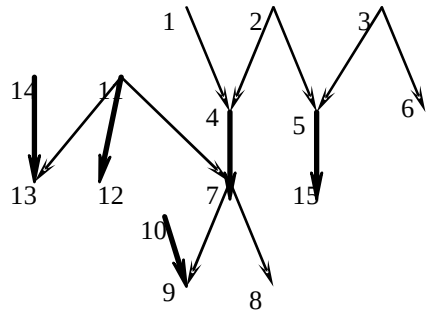


Рис. 1. Граф G.

$$\overline{S} = \left\{ \begin{matrix} 1 & 11 & 5 & 6 \\ 2 & 4 & 7 & 8 \\ & 3 & 10 & 9 \\ & & 14 & 12 \\ & & & 13 \\ & & & 15 \end{matrix} \right\}.$$

Отже, будуватимемо упорядкування довжини $l = 4$ і оцінивши $h \geq 4$. На початку візьмемо $h = 4$. Побудуємо множини μ_i для всіх вершин орграфу, згрупувавши їх за $\overline{\mu}_i$:

$\mu_1 = \{1\}$, $\mu_2 = \{1\}$,

$\mu_3 = \{1,2\}$, $\mu_4 = \{2\}$, $\mu_{11} = \{1,2\}$

$\mu_5 = \{2,3\}$, $\mu_7 = \{3\}$, $\mu_{10} = \{1,2,3\}$, $\mu_{14} = \{1,2,3\}$

$\mu_6 = \{2,3,4\}$, $\mu_8 = \{4\}$, $\mu_9 = \{4\}$, $\mu_{12} = \{2,3,4\}$, $\mu_{13} = \{2,3,4\}$, $\mu_{15} = \{3,4\}$.

Використовуючи наведені вище правила, отримаємо такий список пріоритетів $L = \{1,2,4,11,3,7,5,10,14,8,9,15,6,12,13\}$. Користуючись цим списком пріоритетів отримаємо упорядкування:

$$S = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & 5 & 15 & 9 \\ 3 & 10 & 6 & 13 \\ 11 & 14 & 12 & \end{Bmatrix}.$$

Це упорядкування є оптимальним.

Алгоритм 2 (Алгоритм заснований на розстановці поміток із S).

Ідея алгоритму базується на особливості S, в якому кожна вершина займає крайнє ліве допустиме місце. А отже, номер місця вершини в оптимальному упорядкуванні не може бути меншим за номер її місця в упорядкуванні S. Тоді, будуючи упорядкування з кінця, перевагу віддаватимемо тим вершинам, які мають більше значення $\underline{\mu}_i$.

При цьому, для зменшення числа операцій врахуємо також те, що вершини критичного шляху, тобто вершини, що належать $\tilde{S}$ ($\tilde{S}[i] = \overline{S}[i] \cap S[i], i = \overline{1,l}$), займають чітко визначені місця в упорядкуванні і можуть бути виключені із розгляду.

Алгоритм складається із таких кроків:

1. Вважаємо, що оптимальне упорядкування $S = \tilde{S}$.
2. Приписуємо всім вершинам орграфу мітки $\underline{\mu}_i$.
3. Вважаємо на початку $i = l$.
4. На кожному кроці вибираємо ті вершини, що не належать критичному шляху, наступники яких уже додані до упорядкування. Сортуємо їх за не зростанням міток.
5. Розташовуємо на i-те вільне місце перші $h - \left|\tilde{S}[i]\right|$ із вибраних вершин (або всі вершини, якщо кількість вершин, які не мають вихідних дуг менша за $h - \left|\tilde{S}[i]\right|$).
6. Зменшуємо i на 1. Якщо $i = 0$, кінець роботи алгоритму.

7. Якщо на i-ому кроці виявляється, що вершин із поміткою i більше за $h - \left|\tilde{S}[i]\right|$, то збільшуємо «теоретичне» h на 1 і повторюємо процес побудови упорядкування з початку.

Зауважимо, що в разі рівності поміток при сортуванні доцільно користуватися таким правилом: лівіше розташовуємо ту вершину, яка має більше вхідних дуг. Це сприятиме тому, що на кожному наступному кроці ми забезпечуватимемо існування більшої кількості вершин, що не мають вихідних дуг.

Розглянемо приклад роботи алгоритму.

Нехай задано граф G (рис. 1). Розв'яжемо для цього графу задачу $S(G,l,h)$ за умови, що $l = l$.

У попередньому прикладі наведені упорядкування $\underline{S}$ та $\overline{S}$ для цього орграфу.

Отже, почнемо роботу алгоритму. Вважаємо

$$S = \tilde{S} = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & & & 9 \end{Bmatrix}.$$

Перший крок. Вершини, що не мають вихідних дуг та не належать критичному шляху: 6,12,13,15. Вершини 6,12,13 мають мітку $\underline{\mu}_i = 2$, а вершина 15 має мітку $\underline{\mu}_i = 3$. Сортуємо вершини 6, 12, 13 за кількістю вхідних дуг і отримаємо: 15,13,6,12. Вибираємо дві вершини (15,13). Отримуємо таке упорядкування:

$$S = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & & & 9 \\ & & & 15 \\ & & & 13 \end{Bmatrix}.$$

Другий крок. Вершини, що не мають вихідних дуг та не належать критичному шляху: 5, 6, 10,12, 14. Вершини 5,6,12 мають мітку $\underline{\mu}_i = 2$, всі інші мають мітку $\underline{\mu}_i = 1$, крім того вершина 5 має дві вхідні дуги. Тому сортуємо вершини таким чином: 5,6,12,10,14. Вибираємо перші три. Маємо упорядкування:

. $S = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & & 5 & 9 \\ & & 6 & 15 \\ & & 12 & 13 \end{Bmatrix}$.

Діючи аналогічно отримаємо оптимальне упорядкування:

$$S = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & 10 & 5 & 9 \\ 3 & 11 & 6 & 15 \\ & 14 & 12 & 13 \end{Bmatrix}.$$

Алгоритм 3 (Алгоритм заснований на розстановці поміток із $\overline{S}$).

Ідея алгоритму базується на особливості упорядкування $\overline{S}$, в якому кожна вершина займає крайнє праве допустиме місце. А отже, номер місця вершини в оптимальному упорядкуванні не може бути більшим за номер її місця в упорядкуванні $\overline{S}$. Тоді, будуючи, упорядкування з початку перевагу віддаватимемо тим вершинам, які мають менше значення $\overline{\mu}_i$.

Зауважимо, що в разі рівності поміток при сортуванні доцільно користуватися таким правилом: лівіше розташовуємо ту вершину, яка має більше вихідних дуг.

Алгоритм складається із таких кроків:

1. Вважаємо, що оптимальне упорядкування $S = \tilde{S}$.
2. Приписуємо всім вершинам орграфу мітки $\overline{\mu}_i$.
3. Вважаємо на початку $i = 1$.
4. На кожному кроці вибираємо ті вершини, що не належать критичному шляху, попередники яких вже додані до упорядкування. Сортуємо їх за не зростанням міток.
5. Розташовуємо на i-те вільне місце перші $h - \left|\tilde{S}[i]\right|$ із вибраних вершин (або усі вершини, якщо кількість вершин, які не мають вхідних дуг менша за $h - \left|\tilde{S}[i]\right|$).
6. Збільшуємо i на 1. Якщо $i = l +$, кінець роботи алгоритму.

7. Якщо на i-ому кроці виявляється, що вершин із поміткою i більше за $h - \left|\tilde{S}[i]\right|$, то збільшуємо «теоретичне» h на 1 і повторюємо процес побудови упорядкування з початку.

За цим алгоритмом оптимальне упорядкування для графу (рис. 1) матиме вигляд:

$$S = \begin{Bmatrix} 1 & 4 & 7 & 8 \\ 2 & 5 & 6 & 9 \\ 3 & 10 & 12 & 15 \\ 11 & 14 & 13 & \end{Bmatrix}.$$

Порівняння алгоритмів. Унаслідок особливостей побудови оптимального упорядкування кожним із алгоритмів доцільно користуватися наступним правилом. Якщо у графі для більшості вершин півступінь заходу більше ніж півступінь виходу (тобто у вершини більше вхідних ніж вихідних дуг), то доцільно користуватися другим алгоритмом. У протилежній ситуації доцільно користуватися третім алгоритмом. Якщо ж однозначно визначитись не можливо, то доцільно користуватися першим алгоритмом, який враховує як кількість вхідних, так і кількість вихідних дуг для кожної вершини.

Висновки. Запропоновані у статті алгоритми дозволяють розв'язати задачу побудови паралельного упорядкування заданої довжини та мінімальної ширини за поліноміальний час.

Алгоритми можуть бути легко реалізовані на ЕОМ. Це робить їх використання особливо корисним при розв'язанні практичних задач великої розмірності.

Бібліографічні посилання

1. **Турчина В.А.** Оцінка знизу ширини оптимального упорядкування для задачі S(G,l,h) / В.А. Турчина, Н.К. Федоренко // Питання прикладної математики і математичного моделювання: зб. наукових праць. – Д., – 2008. – С.273-279.

2. **Graham R.L**. Bounds on multiprocessing timing anomalies.//SIAM Journal on Applied Mathematics –1969. – Т. 17, №. 2. – С. 416-429.

Надійшла до редколегії 15.03.10

УДК 519.8

©**С.А. Ус**
Национальный горный университет

О МОДЕЛЯХ ОПТИМАЛЬНОГО РАЗБИЕНИЯ МНОЖЕСТВ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

Запропоновано нові моделі оптимального розбиття множин в умовах нечіткої вихідної інформації.

Предложены новые модели оптимального разбиения множеств в условиях нечеткой исходной информации.

New models of optimal set partitioning in the conditions of fuzzy initial information are offered.

Ключевые слова: оптимальное разбиение множеств, неопределенность, нечеткая информация.

При решении прикладных задач информация, требуемая для построения математической модели, как правило, не является точно определенной, и поэтому построение моделей и разработка методов решения, которые позволяют учесть эту неопределенность является актуальной задачей.

В [1] рассмотрены задачи оптимального разбиения множеств в условиях неопределенности для случаев, когда неопределенность имеет стохастическую природу, а также некоторые модели оптимального нечеткого разбиения множеств. Эти модели соответствуют задачам, в которых нечетким является полученное оптимальное разбиение. При этом исходные данные предполагаются полностью определенными. Однако, как правило, именно исходная информация является довольно расплывчатой, нечеткой.

В данной работе предложены математические модели, в которых нечеткими являются именно исходные данные. При этом получаемое решение должно быть вполне определенным.

Рассмотрим бесконечномерную задачу размещения, общая постановка которой дана в [1].

Пусть потребитель некоторой однородной продукции распределен в области Ω. Конечное число N производителей, расположенных в изоли-

рованных точках τ_i, $i=\overline{1,N}$ области Ω, образуют систему точек $\tau_1,\tau_2,\cdots,\tau_N$, причем координаты некоторых из них, или даже всех, могут быть заранее неизвестны. Известен спрос $\rho(x)$ на продукцию в каждой точке области Ω, а также стоимость доставки продукции $c_i(x,\tau_i)$, $i=\overline{1,N}$ – из пункта производства τ_i в пункт потребления x. предполагается, что прибыль производителя зависит только от транспортных расходов. Мощность i-го производителя определяется суммарным спросом обслуживаемых потребителей и не должна превышать заданных объемов b_i, $i=\overline{1,N}$. Необходимо разбить область Ω на зоны обслуживания каждым из производителей, т. е. на подмножества Ω_1, Ω_2, …, Ω_N, так, чтобы суммарные затраты на доставку продукции были минимальны.

В реальных задачах точно определить спрос, задать его количественную оценку, часто невозможно, но при этом можно оценить его как «высокий», «низкий» и т. д. Таким образом, в каждой точке области мы можем задать некоторое качественное значение, которое удобно описывать с помощью нечетких множеств. Функция спроса при этом будет задана в виде некоторого нечеткого отображения множества Ω на множество $R^+=[0;+\infty)$. Т. е. на множестве Ω будет задано нечеткое отображение ρ с функцией принадлежности $\mu_\rho(\tilde{o},r)\colon \Omega\times R^+\to[0;1]$. Каждому потребителю x из Ω в этом случае ставится в соответствие некоторое нечеткое множество на R^+, описывающее спрос на продукцию. А именно, для каждого фиксированного $\tilde{o}'$ соответствующее ему нечеткое значение функции ρ будет описываться функцией принадлежности $\mu_\rho(\tilde{o}',r)$.

Аналогично, функцию стоимости также можно определить как нечеткое отображение множества Ω^2 в R^+ с функцией принадлежности $\mu_{\tilde{n}}(\tilde{o},\tau_i,r)\colon \Omega\times\Omega\times R^+\to[0;1]$. Нечеткое значение функции для каждого фиксированного $\tilde{o}'$ и τ_i будет нечетким множеством на R^+ с функцией принадлежности $\mu_{\tilde{n}}(\tilde{o}',\tau_i,r)$.

Кроме того, значения мощности производителей b_i, $i=\overline{1,N}$ также могут быть заданы нечетко, в виде нечетких множеств.

Таким образом, получаем следующие задачи оптимального разбиения множеств при нечетких исходных данных:

Задача 1 . (Задача оптимального разбиения при нечетких исходных данных с фиксированными центрами подмножеств без ограничений).

Пусть Ω – замкнутое ограниченное измеримое по Лебегу множество евклидова пространства E^n. Необходимо разбить его на N измеримых по Лебегу подмножеств $\Omega_1, \Omega_2, \cdots, \Omega_N$, центры которых $\tau_1, \tau_2, ..., \tau_N$ заданы, так, чтобы функционал

$$F(\Omega_1, \Omega_2, ..., \Omega_N) = \sum_{i=1}^{N} \int_{\Omega_i} c_i(x, \tau_i) \rho(x) dx \qquad (1)$$

достигал минимального значения при ограничениях:

$$\text{mes}(\Omega_i \cap \Omega_j) = 0, \; i \neq j, \; i, j = \overline{1, N}, \qquad (2)$$

$$\bigcup_{i=1}^{N} \Omega_i = \Omega. \qquad (3)$$

Здесь $c_i(x, \tau_i), \; i = \overline{1, N}$ – нечеткие отображения множества Ω^2 в R^+ с функцией принадлежности $\mu_{\tilde{n}}(\tilde{o}, \tau_i, r)$: $\Omega \times \Omega \times R^+ \to [0;1]$, $\rho(x)$ – нечеткое отображение с функцией принадлежности $\mu_\rho(\tilde{o}, r)$: $\Omega \times R^+ \to [0;1]$.

Задача 2 . (Задача оптимального разбиения при нечетких исходных данных с размещением центров подмножеств без ограничений).

Пусть Ω – замкнутое ограниченное измеримое по Лебегу множество евклидова пространства E^n. Необходимо разбить его на N измеримых по Лебегу подмножеств $\Omega_1, \Omega_2, \cdots, \Omega_N$, и разместить центры подмножеств $\tau_1, \tau_2, ..., \tau_N$ в области Ω так, чтобы функционал

$$F(\Omega_1,\Omega_2,...,\Omega_N,\tau_1,\tau_2,\cdots,\tau_N)=\sum_{i=1}^{N}\int_{\Omega_i}c_i(x,\tau_i)\rho(x)dx \qquad (4)$$

достигал минимального значения при ограничениях:

$$\text{mes}\,(\Omega_i\cap\Omega_j)=0,\ i\neq j,\ i,j=\overline{1,N}, \qquad (5)$$

$$\bigcup_{i=1}^{N}\Omega_i=\Omega. \qquad (6)$$

$c_i(x,\tau_i),\ i=\overline{1,N}$, $\rho(x)$ описываются также как в задаче 1.

Задача 3 . (Задача оптимального разбиения при нечетких исходных данных с фиксированными центрами подмножеств и ограничениями на мощность подмножеств).

Пусть Ω – замкнутое ограниченное измеримое по Лебегу множество евклидова пространства E^n. Необходимо разбить его на N измеримых по Лебегу подмножеств $\Omega_1,\Omega_2,\cdots,\Omega_N$, центры которых $\tau_1,\tau_2,...,\tau_N$ заданы так, чтобы функционал

$$F(\Omega_1,\Omega_2,...,\Omega_N)=\sum_{i=1}^{N}\int_{\Omega_i}c_i(x,\tau_i)\rho(x)dx \qquad (7)$$

достигал минимального значения при ограничениях:

$$\int_{\Omega_i}\rho(x)dx\le b_i,\quad i=\overline{1,N} \qquad (8)$$

$$\text{mes}\,(\Omega_i\cap\Omega_j)=0,\ i\neq j,\ i,j=\overline{1,N}, \qquad (9)$$

$$\bigcup_{i=1}^{N}\Omega_i=\Omega. \qquad (10)$$

Здесь $c_i(x,\tau_i)$, $i=\overline{1,N}$ – нечеткие отображения множества Ω^2 в R^+ с функцией принадлежности $\mu_{\tilde{n}}(\tilde{o},\tau_i,r)$: $\Omega\times\Omega\times R^+\to[0;1]$; $\rho(x)$ нечеткое отображение с функцией принадлежности $\mu_\rho(\tilde{o},r)$: $\Omega\times R^+\to[0;1]$; b_i, $i=\overline{1,N}$ заданные действительные положительные числа, удовлетворяющие условию разрешимости задачи ($\sum_{i=1}^{N} b_i \ge S$, $S=\int_\Omega \rho(x)dx$).

Задача 4 . (Задача оптимального разбиения при нечетких исходных данных с размещением центров подмножеств и ограничениями на мощность подмножеств).

Пусть Ω – замкнутое ограниченное измеримое по Лебегу множество евклидова пространства E^n. Необходимо разбить его на N измеримых по Лебегу подмножеств $\Omega_1, \Omega_2, \cdots, \Omega_N$, и разместить центры подмножеств $\tau_1,\tau_2,...,\tau_N$ в области Ω так, чтобы функционал

$$F(\Omega_1,\Omega_2,...,\Omega_N,\tau_1,\tau_2,\cdots,\tau_N)=\sum_{i=1}^{N}\int_{\Omega_i} c_i(x,\tau_i)\rho(x)dx \qquad (11)$$

достигал минимального значения при ограничениях:

$$\int_{\Omega_i}\rho(x)dx \le b_i, \quad i=\overline{1,N} \qquad (12)$$

$$\text{mes}\,(\Omega_i\cap\Omega_j)=0,\ i\ne j,\ i,j=\overline{1,N}, \qquad (13)$$

$$\bigcup_{i=1}^{N}\Omega_i=\Omega. \qquad (14)$$

Здесь функции $c_i(x,\tau_i)$, $i=\overline{1,N}$ и $\rho(x)$, а также параметры b_i, $i=\overline{1,N}$ определяются как в предыдущей задаче.

Учитывая возможную нечеткость в задании $b_i, \quad i=\overline{1,N}$, получим задачу вида:

Задача 5 . (Задача оптимального разбиения при нечетких исходных данных с размещением центров подмножеств и ограничениями на мощность подмножеств).

Пусть Ω – замкнутое ограниченное измеримое по Лебегу множество евклидова пространства E^n. Необходимо разбить его на N измеримых по Лебегу подмножеств $\Omega_1, \Omega_2, \cdots, \Omega_N$, и разместить центры подмножеств $\tau_1, \tau_2, ..., \tau_N$ в области Ω так, чтобы функционал

$$F(\Omega_1, \Omega_2, ..., \Omega_N, \tau_1, \tau_2, \cdots, \tau_N) = \sum_{i=1}^{N} \int_{\Omega_i} c_i(x, \tau_i)\rho(x)dx \qquad (15)$$

достигал минимального значения при ограничениях:

$$\int_{\Omega_i} \rho(x)dx \le \tilde{b}_i, \quad i=\overline{1,N} \qquad (16)$$

$$\text{mes}\,(\Omega_i \cap \Omega_j) = 0, \ i \neq j, \ i, j = \overline{1,N}, \qquad (17)$$

$$\bigcup_{i=1}^{N} \Omega_i = \Omega. \qquad (18)$$

Здесь $c_i(x, \tau_i), \ i=\overline{1,N}$ – нечеткие отображения множества Ω^2 в R^+ с функцией принадлежности $\mu_{\tilde{n}}(\tilde{o}, \tau_i, r)$: $\Omega \times \Omega \times R^+ \to [0;1]$; $\rho(x)$ – нечеткое отображение с функцией принадлежности $\mu_\rho(\tilde{o}, r)$: $\Omega \times R^+ \to [0;1]$; $\tilde{b}_i, \quad i=\overline{1,N}$ - заданные нечеткие числа, удовлетворяющие условию разрешимости задачи.

Интегралы в постановках отображают суммирование по всем x из Ω и в зависимости от выбранного метода решения задачи рассматриваются как интегралы Лебега либо как нечеткие интегралы.

Предложенные модели сформулированы для случая, когда должно быть получено четкое решение задачи в виде обычного разбиения множества Ω и могут быть обобщены на класс нечетких разбиений.

Киселева Е.М. Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М.Киселева, Н.З. Шор – К., 2005 – 564 с.

Надійшла до редколегії 15.05.10

УДК: 004.021

©**В.А. Чепурко**
Государственный университет информатики и искусственного интеллекта

МИНИМИЗАЦИЯ ОРИЕНТИРОВАННЫХ АЦИКЛИЧЕСКИХ ГРАФОВ С ОТМЕЧЕННЫМИ ВЕРШИНАМИ

Поняття мінімізації використовується для зменшення графів і широко застосовується в багатьох галузях. Основним завданням є знайти еквівалентні стани графа з поміченими вершинами. Запропоновано новий алгоритм мінімізації ациклічних орієнтованих графів із зазначеними вершинами. Визначено тимчасова складність.

Понятие минимизации используется для уменьшения графов и широко применяется во многих областях. Основной задачей является найти эквивалентные состояния графа с помеченными вершинами. Предложен новый алгоритм минимизации ориентированных ациклических графов с отмеченными вершинами. Определена временная сложность.

The notion minimization are used for graph reduction and are widely employed in many areas. The main task is to find the equivalent of states graphs with marked vertices. Proposed a new algorithm minimize oriented, acyclic graphs with marked vertices. Defined by its time complexity.

Ключевые слова: алгоритм, минимизация, эквивалентность, разбиение.

Вступление. Графы с отмеченными вершинами являются одной из основных моделей при рассмотрении двух направлений. Первое – это задачи, связанные с анализом операционной среды с помощью блуждающих по ним агентов (мобильных роботов, автоматов, поисковых программ и т. п.) относятся к категории традиционных задач искусственного интеллекта. Одной из центральных и актуальных как в теоретическом, так и в прикладном аспектах проблем, возникающих при исследованиях взаимодействия автоматов и операционной среды, является проблема анализа или распознавания свойств этой среды при различной априорной информации и при различных способах взаимодействия автомата и операционной среды. В таких задачах рассматриваются различные геометрические модели сред.

Операционная среда рассматривается как неориентированный граф с помеченными вершинами [1]. Такие графы возникли первоначально как блок_схемы и схемы программ, а в настоящее время находят применение в задачах навигации роботов [2].

Второе направление – это блок_схемы программ. Задача эквивалентности программ относится к числу наиболее важных задач теории программирования. Особое значение задачи эквивалентности обусловлено тем, что именно к этой проблеме эквивалентных преобразований сводится большинство задач оптимизации, верификации и анализа программ [2]. Поэтому разработка и внедрение эффективных алгоритмов проверки эквивалентности программ оказывает заметное влияние на повышение производительности и качества многих инструментальных средств анализа и преобразования программ. В последние годы наряду с задачами повышения эффективности и надёжности программ не меньшую актуальность приобрела задача своевременного обнаружения постороннего кода, которая включает выявление недекларированных возможностей программных продуктов, а также обнаружение и устранение программ_вирусов.

В обоих случаях такие графы могут содержать большое число вершин, поэтому возникает задача уменьшения их количества с сохранением всех необходимых свойств графа. Эта задача известна как задача минимизации.

В настоящей работе рассматривается проблема минимизации ориентированных ациклических графов с отмеченными вершинами. Проблема позволяет находить эквивалентные графы, с помощью которых можно представить различные системы. Анализ графов проводится методами, аналогичными методам теории автоматов. Эти методы созданы для графовых систем, не являющихся конечными автоматами, но являющимися, в некотором смысле, автоматоподобными системами.

Проблема минимизации состоит в нахождении разбиения всех вершин графа на классы эквивалентных. В [1] предложен алгоритм минимизации графа временной сложности $O(2^n)$ в общем случае и $O(n^2)$ для так называемых детерминированных [2] графов.

Постановка задачи. Пусть $G=(V, E, M, \mu)$ конечный, связный, помеченный, ориентированный ациклический граф, без петель и кратных дуг [1], где V – конечное множество вершин, E – множество смежных вершин графа, такое что $E(v)=\{u\in V|(v,u)$ – дуга$\}$ и $E^{-1}(v)=\{u\in V|(u,v)$ – дуга$\}$, M – множество номеров меток, $\mu: S \to M$ – сюррективная функция размечивания вершин. Конечную последовательность вершин $p=g_1\cdots g_k$ такую, что

$g_{i+1} \in E(g_i)$, $1<i<=k$, назовём путём длины $(k-1)$ в графе G. Слово $\mu(p)=\mu(g_1)\ldots\mu(g_k)$ назовём отметкой пути p. Отметку любого пути, исходящего из вершины $g \in V$, будем называть словом, порожденным вершиной g. Язык Lg определим как множество всех слов, порождённых вершиной g. Вершины g, h – назовём эквивалентными, если $Lg=Lh$, и отличимыми в противном случае. Граф G называется приведенным, если все его вершины попарно отличимы. Задача сводится к построению классов эквивалентных вершин.

Идея алгоритма.

1. Строим начальное разбиение, номера окончательно разбитых классов помещаем в множество ОЖИДАНИЕ.

2. Установим начальное значение параметру преждевременной остановки алгоритма

3. Пока ОЖИДАНИЕ не пусто выполняем следующее:

3.1. Уменьшаем степень предшествующих вершин.

3.2.1. Построим временные классы разбиения.

3.2.2. Строим разбиение временных классов. Номера окончательно разбитых классов помещаем в множество ПОДГОТОВ.

3.2.ОЖИДАНИЕ: = ПОДГОТОВ.

Метод решения.

ВХОД: граф G.

ВЫХОД: разбиение на классы эквивалентных вершин p'={B[1],…,B[q]} множества V.

ДОПОЛНИТЕЛЬНЫЕ ОПРЕДЕЛЕНИЯ:

n – количество вершин графа;

m – количество меток графа;

$E(v)$ – множество последующих вершин, задано изначально;

$E^{-1}(v)$ – множество предыдущих вершин, для построения требует $O(e)$ шагов;

$klass(v)$ – номер класса вершины v;

$\mu(v)$ – номер метки вершины;

$B[k]$ – класс эквивалентных вершин, где к – номер класса;

sum – параметр преждевременной остановки алгоритма;

$|\mu[E(v)]=\{\mu(u)|(v,u)$ - дуга$\}|$ – количество различных номеров меток видимых из вершины v, для построения требует $O(e)$ шагов;

$st(v)$ – степень вершины изначально равна $|E(v)|$.

АЛГОРИТМ

Для всех $v_i \in V$ выполняем

klass(v_i)= (m+1)*(μ(v_i) -1)+ |μ[E(v_i)]|+1
Если B[klass(v_i)] = null то
B[klass(v_i)] выделить память
Если |μ[E(v_i)]| = 0 то
ОЖИДАНИЕ добавить $(m+1)*(\mu(v_i)-1)+0+1$
B[klass(v_i)] добавить v_i
sum=q
для $k \in$ ОЖИДАНИЕ
sum=sum+|B[k]|-1
3. Пока ОЖИДАНИЕ не пусто или sum<n выполняем следующее:
3.1. Для всех k принадлежащих ОЖИДАНИЕ выполняем
Уменьшаем степень предшествующих вершин.
Для $v_i \in$ B[k] выполняем
Для $v_j \in$ E-1(v_i) выполняем
st(v_j) := st(v_j) − 1
3.2. Для всех k принадлежащих ОЖИДАНИЕ выполняем
Берём последовательно $k \in$ ОЖИДАНИЕ и удаляем
3.2.1. Построение временных классов для B[k]
Для $v_i \in$ B[k] выполняем
Для $v_j \in$ E−1(v_i) выполняем
Если st(v_j) == 0 то
Если BT0[klass(v_j)]==null то
Выделим память для BT0[klass(v_j)]
ВРЕМ0 добавить klass(v_j)
BT0[klass(v_j)] добавить v_j
Иначе
BT0[klass(v_j)] добавить v_j
Иначе
Если BT[klass(v_j)]==null то
Выделим память для BT[klass(v_j)]
ВРЕМ добавить klass(v_j)
BT[klass(v_j)] добавить v_j
Иначе
BT[klass(v_j)] добавить v_j
3.2.2. Разобьём временные классы BT и BT0 полученные из B[k].
Для всех $k \in$ ВРЕМ0
Если $|B[k]| \neq |BT0[k]|$ то
q=q+1 // увеличиваем количество классов
(адрес)B[q] : = (адрес)BT0[k]
BT0[k] := null

```
            B[k]:=B[k]-B[q] // разбиваем B[k]
            sum:=sum+|B[q]|
            ПОДГОТОВ добавить q
        Иначе
            sum:=sum+|B[q]|-1
            ПОДГОТОВ добавить q
    Для всех k ∈ ВРЕМ
        Если |B[k]| ≠ |BT[k]| то
            q=q+1  // увеличиваем количество классов
            (адрес)B[q] : = (адрес)BT[k]
            BT[k] := null
            B[k]:=B[k]-B[q] // разбиваем B[k]
            sum:=sum+1
```

4.3.ОЖИДАНИЕ: = ПОДГОТОВ.

Обоснование алгоритма

МИНИМАЛЬНЫЙ ШАГ АЛГОРИТМА

Элементарный шаг – сравнение двух чисел, операция присвоения, операции сложения вычитания, и т. д.

За минимальный шаг времени выполнения алгоритма берутся операции добавления удаления значения в множество. Множества всегда являются упорядоченными по возрастанию, следовательно операции занимают следующее количество элементарных шагов. Будем считать, что $\log(n)$ это целая часть логарифма по основанию 2.

Добавление – $\log(n)+1$, где n текущее количество элементов множества.

Удаление – $\log(n)+1$, где n текущее количество элементов множества.

Это время необходимо для нахождения положения элемента в множестве.

Так как изначально множества являются пустыми, то добавление n элементов в пустое множество будет занимать не более $n*\log(n)$ шагов.

Выведена формула суммы ряда по которой можно посчитать количество необходимых шагов для заполнения множества n элементами. Для помещения первого элементов множество нужен 1 шаг, далее количество шагов возрастает на 1 для каждого увеличения разряда двойки:

$$1+2^0(0+2)+2^1(1+2)+\ldots+2^{\log(n)-1}((\log(n)-1)+2)+(N-2^{\log(n)})(\log(n)+2)$$

Теорема 1. *Число классов k расширенного начального разбиения удовлетворяет неравенству $m \le k \le m(m+1)$, причём обе оценки достижимы.*

Доказательство. Для графов делается начальное разбиение следующим образом: вершины графа делятся на классы эквивалентности по двум признакам:

- по номеру отметки,
- по $E(v)$ среди вершин с одинаковой отметкой.

Вершины с разными отметками не могут быть эквивалентны, так как все слова таких вершин отличаются на первой букве слова. В одинаковые классы могут попасть только те вершины с одинаковыми отметками, из которых видно одинаковое количество различных отметок. Это значит, что если $|\mu[E(v_i)]| \neq |\mu[E(v_j)]|$ из вершин v_i и v_j видно различно количество отметок, то $Lv_i \neq Lv_j$ языки этих вершин не равны. Другими словами если у двух вершин с одной отметкой количество слов длины 2 различное, то эти вершины попадают в разные классы эквивалентности. Из каждой вершины графа может выходить от 0 до m дуг, таким образом из вершин с одинаковой отметкой в лучшем случае может получиться $m+1$ класс. Так как всего отметок в графе m, то в результате такого разбиения мы можем получить максимум $m(m+1)$ классов, а не m при обычном разбиении. В худшем случае мы можем получить m классов, данная ситуация возможна, когда $|\mu[E(v)]|$ равна для всех $v \in V$. Для ациклических графов наихудшее разбиение не достижимо, так как в таких графах всегда существуют висячие вершины, у которых степень равна 0. А так как граф связный, то существуют вершины у которых $st(v)>0$. Данное разбиение будем называть расширенным начальным разбиением.

Для выполнения расширенного начального разбиения нам необходимо просмотреть все вершины графа – это n шагов. Для каждой вершины выполняются действия присвоения номера класса и помещение во множество класса, проверка существования класса, добавление номера в PREPARED. Если после обработки мы получили меньше n классов, то перенумеруем классы из PREPARED, что требует не более n шагов. Сложность данного этапа работы $O(n)=3n+n=n$.

Пример. Рассмотрим примеры, для которых оценки $m(m+1)$ и m достижимы. На рисунке 1 изображен детерминированный граф G.

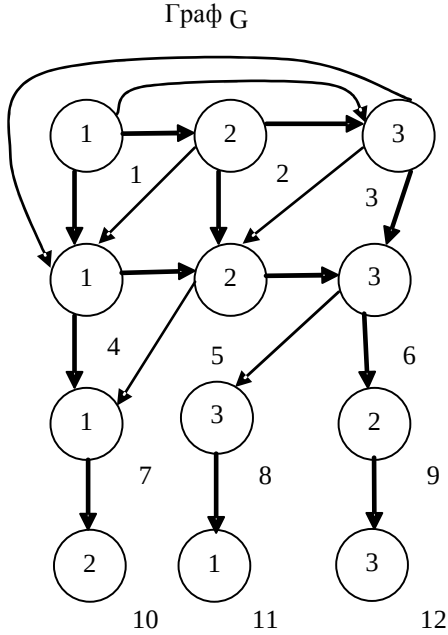


Рис. 1 Достижимость m(m+1)

Номер класса, в который помещается вершина, вычисляется по формуле $(m+1)*(\mu(v_i)-1)+|\mu[E(v_i)]|+1$. Получаем следующее разбиение $\{(11)_{4*(1-1)+0+1}, (7)_{4*(1-1)+1+1}, (4)_{4*(1-1)+2+1}, (1)_{4*(1-1)+3+1}, (10)_{4*(2-1)+0+1}, (9)_{4*(2-1)+1+1}, (5)_{4*(2-1)+2+1}, (2)_{4*(2-1)+3+1}, (12)_{4*(3-1)+0+1}, (8)_{4*(3-1)+1+1}, (6)_{4*(3-1)+2+1}, (3)_{4*(3-1)+3+1}\}$. Теперь запишем текущее разбиение с посчитанными номерами классов $\{(11)_1, (7)_2, (4)_3, (1)_4, (10)_5, (9)_6, (5)_7, (2)_8, (12)_9, (8)_{10}, (6)_{11}, (3)_{12}\}$. В данном примере мы получили полное разбиение графа G на классы эквивалентных вершин, не приступая к основному алгоритму. При использовании обычного метода мы получим следующее начальное разбиение $\{(1,4,7,11)_1, (2,5,9,10)_2, (3,6,8,12)_3\}$.

На рисунке 2 изображен граф у которого начальное разбиение равно m классам. Так как из всех вершин графа наблюдаем одинаковое количество различных отметок, то начальное разбиение будет следующим $\{(1,4,5)_1, (2)_2, (3)_3\}$.

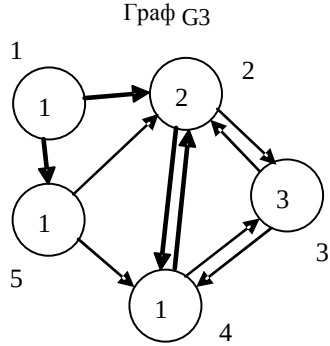


Рис. 2. Достижимость m

Теорема 2. *Данный алгоритм выполняет правильное разбиение на классы эквивалентных вершин за O(e) шагов.*

Доказательство. В первом пункте алгоритма проводится начальное разбиение на классы эквивалентных вершин, описанное в теореме 1. Выбираются классы, у которых степень вершины равна 0. Данные классы являются окончательно разбитыми, так как язык вершин $Lg(v_i)$ равен одному слову, состоящему из одной отметки. В пункте 2 устанавливается начальное значение для sum. Этот параметр позволяет остановить алгоритм раньше полного обхода графа при наличии эквивалентных вершин в графе. Он содержит не только количество классов эквивалентных вершин, но и количество вершин, которые уже не могут быть разбиты. Рассмотрим применение sum на примере рисунка 4.

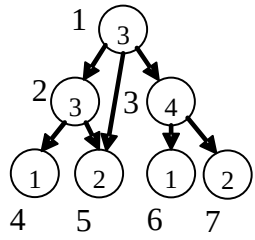


Рис. 4. Ациклический граф G3

Начальное разбиение для графа G_3 равно $\{(1)_1,(2)_2,(3)_3,(4,6)_4,(5,7)_5\}$. Так как классы 4 и 5 содержат эквивалентные вершины, то больше ничего разбиться не может. На 2 шаге sum=5, на 3-м шаге алгоритма sum=5+|B[5]|-1=6 и потом sum=6+|B[5]|-1=7. Мы получили sum=N, значит, на 3 пункте алгоритм остановится, что сокращает много времени работы алгоритма. Далее в алгоритме sum увеличивается соответствующим образом, что бы содержать в себе и новые полученные классы, и окончательно разбитые вершины. Следующий пункт алгоритма является основным, и выполняется, пока мы не обойдем все вершины, или алгоритм не остановится преждевременно. В пункте 3.1 уменьшается степень всех предшествующих вершин для окончательно разбитых классов из множества ОЖИДАНИЕ. Просматриваются все предшествующие вершины и каждый раз $st(v_j):= st(v_j)-1$ уменьшается степень вершин. После выполнения данного цикла появляются вершины со степенью равной 0, значит, эти вершины могут быть разбиты только классами из множества ОЖИДАНИЕ, так как видны только нижним уровнем графа, поэтому после разбиения полученные классы станут окончательно разбитыми. Вершины, степень которых не равна 0 тоже могут быть разбиты, но полученные классы не являются окончательно разбитыми, так как они пересекают не только самый нижний уровень графа. Таким образом, при уменьшении степени вершин мы удаляем нижний уровень графа. Шаг 4.2 выполняется для каждого класса $\in$ ОЖИДАНИЕ в 2 этапа. Класс, который берётся на обработку удаляется из ОЖИДАНИЕ. На первом этапе мы строим временные классы разбиения графа шаг 4.2.1. Для этого необходимо 2 типа временных классов. $BT0$ – классы содержащие окончательно разбитые вершины и BT – классы содержащие остальные вершины. Для их построения мы просматриваем все предшествующие вершины v_j, вершин $v_i \in B[k]$. Если степень равна 0 то вершина помещается в $BT0[v_j.klass]$ иначе в $BT[v_j.klass]$. Вершины попадают в классы с номерами соответствующими текущему классу вершины, таким образом, мы сразу создаём разбиение и не нужно проверять наличие вершины во всех классах. Так как нужно будет потом обрабатывать классы, то их номера добавляются в множества ВРЕМ0 и ВРЕМ. Временные классы постоянно не хранятся в памяти, память для элементов класса выделяется только при необходимости добавления вершины. После обработки классы обнуляются. Таким образом экономится используемая память. На втором этапе, шаг 4.2.2, мы обрабатываем полученные временные классы. Осуществляем разбиение. Поочерёдно просматриваем классы из множеств ВРЕМ0 и ВРЕМ. Если мощность реально класса и временного равна, тогда временный класс полностью пересекает класс, временный класс обнуляется. В противном случае – увеличивается q количество клас-

сов эквивалентности. Адрес в памяти временного класса присваивается B[q], временный класс при этом обнуляется. Из класса B[k] удаляются вершины класса B[q]. Классы полученные при обработке ВРЕМ0 помещаются в множество ПОДГОТОВ, которое собирает все окончательно разбитые классы данного уровня. Параметр sum увеличивается в зависимости от типа полученных классов. После обработки всех классов на шаге 4.2 множество ОЖИДАНИЕ является пустым и на шаге 4.3 номера классов из ПОДГОТОВ помещаются в ОЖИДАНИЕ. ПОДГОТОВ обнуляется и далее алгоритм обрабатывает следующий уровень графа. Если выполнения шага 4.2 множество ПОДГОТОВ остаётся пустым, значит мы прошли по всем вершинам и алгоритм останавливается.

Пункт 1. Выполняется за N шагов, что доказано в теореме 1.

Пункт 2. Выполняется за 1 шаг.

Пункт 3. Данный цикл выполняется |ОЖИДАНИЕ| раз. Количество классов в ОЖИДАНИЕ не превышает M. В цикле выполняется 1 операция.

Пункт 4. В его цикле обработку проходят только окончательно разбитые вершины, значит каждая вершина обрабатывается в пункте 4 только 1 раз. Для каждой вершины мы просматриваем множество предшествующих вершин, это количество равно количеству дуг входящих в рассматриваемую. После обработки уровни графа удалятся, значит и удаляются рассмотренные дуги, поэтому каждая дуга обрабатывается в цикле только 1 раз. В пункте 4.1 мы уменьшаем степень предшествующих вершин, значит проходим каждую дугу 1 раз и выполняем 1 операцию. В цикле пункта 4.2.1 дуги просматриваются так же 1 раз, и выполняются операции добавления в множество, выделение памяти для класса и добавление номера класса в множество. Классов за 1 проход не может быть больше просмотренных дуг, количество вершин добавляемых в множество так же не превышает количество дуг, так как по 1 дуге мы можем увидеть только 1 предшествующую вершину. Циклы в пункте 4.2.2 выполняются за количество шагов равное количеству дуг, так как вершины и классы создавались в пункте 4.2.1, и их не может быть больше количества дуг данного уровня. В циклах выполняются только элементарные операции, и операция удаления вершин из множеств. Так как количество вершин не превышает количество дуг уровня, то и удалений не может быть больше. Следовательно, для каждой дуги выполняется конечное количество элементарных операций, которое является константой. В пункте 4.3 выполняются операции добавления и удаления номеров классов из множества. В множествах мо-

гут быть номера только окончательно разбитых классов, это значит что всего добавлений и удалений за время работы алгоритма будет n. Следовательно, верхняя асимптотическая временная сложности алгоритма не превышает количество дуг графа. Что и требовалось доказать.

Теорема 2 доказана.

Для различных видов ациклических графов может быть разное количество дуг зависящее от n. Так для деревьев $e = O(n)$, для детерминированных ациклических графов $e = O(n*m)$, так как в них из каждой вершины исходит не более m дуг. Для недетерминированных ациклических графов последнее ограничение снимается и $e = O(n^2)$. Исходя из сказанного имеет место:

Следствие. Число шагов алгоритма минимизации не превосходит k

- $k \leq O(n)$, для деревьев;
- $k \leq O(mn)$, для детерминированных ациклических графов;
- $k \leq O(n^2)$, для недетерминированных ациклических графов.

Выводы. Предложен новый метод начального расширенного разбиения графов. Число классов k расширенного начального разбиения удовлетворяет неравенству $m \leq k \leq m(m+1)$, причём, обе оценки достижимы.

Предложен алгоритм минимизации ориентированных ациклических графов с отмеченными вершинами. Доказано что данный алгоритм выполняет правильное разбиение на классы эквивалентных вершин за $O(e)$ шагов.

Для различных видов ациклических графов может быть разное количество дуг зависящее от n. Так для деревьев $e = O(n)$, для детерминированных ациклических графов $e = O(n*m)$, так как в них из каждой вершины исходит не более m дуг. Для недетерминированных ациклических графов последнее ограничение снимается и $e = O(n^2)$. Исходя из сказанного имеет место:

Следствие. Число шагов алгоритма минимизации не превосходит k

- $k \leq O(n)$, для деревьев;
- $k \leq O(mn)$, для детерминированных ациклических графов;
- $k \leq O(n^2)$, для недетерминированных ациклических графов.

Выражаю благодарность И.С. Грунскому за помощь в работе.

Библиографические ссылки

1. **Сапунов С.В.** Анализ графов с помеченными вершинами: Дисс… канд. физ. - мат. наук: 27.09.07. /С.В. Сапунов – Донецк, 2007. – 150 с.

2. **Dudek J., Jenkin M., Milios E., Wilkes D.** Map validation and Robot Self-Location in a Graph-Like World // Robotics and Autonomous Systems, 1997.-V.22(2).-P.159-178.

3. **R. Gentilini, C. Piazza and A. Policriti** From Bisimulation to Simulation. Coarest partition problems // Dip. Di Matematica e Informatica – Universita di Udine Via Le Scienze, 206 – 33100 Udine – Italy.

Надійшла до редколегії 11.04.10