УДК 681.3

**О.М. Подоляка, ст. викл.**
*Національний аерокосмічний університет «ХАІ»*
**О.О. Подоляка, к.т.н., доц.**
*Харківський національний автомобільно-дорожній університет*

# АЛГОРИТМ ПОШУКУ МНОЖИНИ РОЗВ'ЯЗКІВ БАГАТОКРИТЕРІАЛЬНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

*Розглянуто ефективний поліноміальний алгоритм пошуку множини розв'язків багатокритеріальної задачі про призначення, множина всіх критеріїв якої упорядкована за їх важливістю.*

**Вступ.** Практична цінність наведених результатів міститься у можливості їх використання для розробки складних систем оперативного управління на транспорті та виробництві, коли кожна з робіт, що виконується у системі, характеризується множиною упорядкованих за значимістю параметрів. В якості цих параметрів можуть виступати, наприклад: собівартість, прибуток, час виконання, належна кваліфікація виконавця, необхідні об'єми ресурсів, тривалість етапів тощо. Також слід зазначити, що задача упорядкування параметрів робіт або критеріїв у більшості практичних систем не є складною. Це упорядкування є прямим слідством властивостей багатьох систем й має бути надане менеджерами проектів або керівництвом фірм.

**Постановка проблеми у загальному вигляді.** Необхідно знайти довершене паросполучення (перестановку) $\pi^*$ з мінімальною вагою на множині усіх можливих паросполучень $P$ для заданої матриці $\beta^{MM}$, де $M$ – порядок матриці. Кожне з можливих довершених паросполучень (перестановок) у матриці $\beta^{MM}$ будемо позначати $\pi$.

Функціонал задачі про призначення.

$$\rho(\pi^*) = \min_{\pi \in P} \sum_{i,j \in \pi} \beta_{ij} , \tag{1}$$

де $\rho(\pi) = \sum_{i,j \in \pi} \beta_{ij}$ – вага паросполучення $\pi$; $\beta_{ij}$ – елемент матриці, який належить паросполученню.

У класичній задачі про призначення елемент матриці $\beta_{ij}$ являє собою один критерій, а у багатокритеріальній – елемент $\beta_{ij}$ – складний тип даних, кожне з полів якого – це критерій, який характеризує один з параметрів роботи. Тобто класична однокритеріальна задача відрізняється від багатокритеріальної лише представленням елемента $\beta_{ij}$, який означає вартість призначення.

Тому для багатокритеріальної задачі маємо:
$\beta_{ij} = c = (c_1,...,c_i,...,c_n)$, де $c_i$ – вага $i$-го критерію вектора $c$; $n$ – кількість критеріїв.

У статті розглядається багатокритеріальна задача про призначення, кожен елемент матриці якої являє множину упорядкованих за важливістю критеріїв. Будемо вважати, що критерії упорядковані у відповідності до значення їх індексів.

Тому $c_1 > c_2 > ... > c_i > ... > c_n$.

**Фунціонал багатокритеріальної задачі.** Оскільки у загальному випадку деякі критерії вектора $\beta_{ij}$ потрібно мінімізувати, а інші, навпаки, максимізувати, то може скластися враження, що функціонал (1) не може бути використаний для багатокритеріальної задачі. Але застосування властивості двоїстості задач лінійного програмування дозволяє представити функціонал багатокритеріальної задачі у вигляді, коли всі критерії або мінімізуються, або максимізуються. Ці еквівалентні перетворення матриці $\beta^{MM}$ задачі виконуються на основі наступного твердження.

**Твердження 1. Симетрія задач на мінімум та максимум (двоїстість)**

$$\rho(\pi^*) = \max_{\pi \in P} \sum_{i,j \in \pi} \beta_{ij}(c_k) = \min_{\pi \in P} \sum_{i,j \in \pi} \beta_{ij}(-c_k), \tag{2}$$

де $\beta_{ij}(c_k)$ – $k$-й критерій елемента матриці $\beta_{ij}$.

Зрозуміло, що класична однокритеріальна задача про призначення $(n=1)$ є окремим випадком багатокритеріальної задачі. Пошук множини розв'язків класичної задачі про призначення та задачі про призначення, функціонал якої побудовано із застосуванням операції добутку, розглянуто у роботах [1, 2]. Для розв'язання цих задач використовувався нормалізаційний алгоритм. Його обчислювальна складність становить $M^3 Log(M)$. Ідея алгоритму полягає у пошуку множини елементів матриці $\beta^{MM}$, які належать множині оптимальних розв'язків. Ці елементи утворюють *оптимальну множину*. Пошук одного розв'язку задачі на основі оптимальної множини може бути виконаний за поліноміальний час $O(M^2)$. Для знаходження одного розв'язку можна використати алгоритми побудови довершеного паросполучення на основі елементів оптимальної множини. Найбільш ефективні алгоритми для побудови максимального паросполучення використовують ланцюги, що чергуються [3, 4].

**Теоретичні основи нормалізаційного алгоритму.** Оптимальний розв'язок неможливо поліпшити. Тобто будь-яка допустима заміна (перестановка) елементів оптимального розв'язку не приводить до його поліпшення, у кращому випадку, заміна може дати еквівалентний розв'язок. Це твердження, по суті, є визначенням оптимального розв'язку.

Тому, по-перше, якщо розв'язок не можна поліпшити, то він є оптимальним; по-друге, необхідно завершити роботу оптимізаційного алгоритму, який було використано для розв'язання задачі (умова завершення алгоритму). Щоб поліпшити розв'язок, треба замінити деякі його елементи на інші. Ця заміна виконується наступним чином.

**Заміна елементів розв'язку.** Розглянемо рисунок, на якому схематично представлена матриця $\beta$.

| стовпець / рядок | $b$ | | $d$ |
|---|---|---|---|
| $a$ | $\beta_{a,b}$ | $\dots$ | $\beta_{a,d}$ |
| $\dots$ | $\dots$ | | $\dots$ |
| $c$ | $\beta_{c,b}$ | $\dots$ | $\beta_{c,d}$ |

*Рис. 1. Перехресна заміна*

Якщо в оптимальному розв'язку матриці $\beta$ замінити будь-яку пару елементів за правилом $\beta_{a,b} \wedge \beta_{c,d} \Rightarrow \beta_{a,d} \wedge \beta_{c,b}$, то отриманий розв'язок може бути:

1. Еквівалентним оптимальним розв'язком, що відрізняється від початкового двома елементами.
2. Допустимим (не оптимальним) розв'язком.

Ця заміна називається *перехресною*, або заміною по два. Зазвичай вона використовується при розв'язанні задачі комівояжера і задачі про призначення [3, 4].

*Зауваження*. В матриці $\beta^{MM}$ можуть існувати підматриці порядку $K \in [3, M]$, в яких можуть існувати заміни елементів по $K$ [4].

Нехай: $\beta_{a,d} \wedge \beta_{c,b}$ – елементи оптимального розв'язку, а $\beta_{a,b} \wedge \beta_{c,d}$ – елементи, що замінюються, то $\beta_{a,b} + \beta_{c,d} \le \beta_{a,d} + \beta_{c,b}$.

Тоді

$$\left(\beta_{a,b} - \beta_{c,b}\right) \le \left(\beta_{a,d} - \beta_{c,d}\right). \tag{3}$$

**Лінійні пари.** Лінійною парою *(ЛП) $i$-го та $k$-го рядків матриці називатимемо елемент, який обчислюється за наступною формулою:*

$$D_j = \beta_{ij} - \beta_{kj},\ j \in [0, M-1]. \tag{4}$$

Елемент $\beta_{ij}$ називатимемо клієнтським значенням лінійної пари, а $\beta_{kj}$ – серверним значенням лінійної пари. *Мультимножина лінійних пар* (МмЛП) – це упорядкована послідовність лінійних пар. Як основний критерій упорядкування виступає величина лінійної пари. Всю мультимножину упорядковуватимемо за незростанням значень елементів.

Лінійну пару називатимемо *найбільшою*, якщо їй відповідає найбільший елемент МмЛП. Клієнтське значення називатимемо *найбільшим*, якщо воно формує найбільшу лінійну пару МмЛП.

Лінійну пару називатимемо *максимальною*, якщо їй відповідає один з рівних максимальних елементів, і при цьому в МмЛП існує відмінний від максимального елемент. Клієнтське значення називатимемо *максимальним*, якщо йому відповідає максимальний елемент МмЛП.

У загальному випадку найбільший елемент є максимальним, але не навпаки.

Визначення найменшого і мінімального значення вводиться так само.

**Теорема 1. Заборони**

Якщо в МмЛП рядків $r$ та $i$ є найбільший елемент $D_j$, то елемент $\beta_{rj}$ не може формувати оптимальний розв'язок задачі, а, отже, може бути заборонений.

Треба відзначити, що ця теорема є окремим випадком загальної теореми, яка може бути сформульована наступним чином.

Якщо у матриці $\beta^{MM}$ існує рядок й підмножина $C$ з $n$ елементів цього рядка ( $n < M$ ) формує максимальні клієнтські значення у $n$ МмЛП, то всі елементи підмножини $C$ у матриці $\beta^{MM}$ можна заборонити.

**Ролі лінійних пар.** Зрозуміло, що після застосування теореми 1 деякі елементи матриці будуть заборонені. І в МмЛП з'являться лінійні пари, які матимуть заборонені елементи. Класифікуємо ці пари і визначимо критерії їх упорядкування в МмЛП.

Нехай кожний елемент матриці містить:

1. Значення.
2. Дозвіл.
3. Індекс стовпця елемента в матриці $\beta^{MM}$ (щоб після упорядкування елементів МмЛП можна було визначити місцеположення елемента у початковій матриці).

Якщо дозвіл елемента $a$ дорівнює false (або 0), то елемент матриці є забороненим і позначається $\bar{a}$, отже він не може утворювати оптимальний розв'язок.

Нехай $a$ і $c$ – пара елементів, що знаходяться в різних рядках і в одному стовпці матриці. Тоді можливі наступні комбінації дозволів.

*Таблиця 1*

Типи лінійний пар

| № пари / Дозвіл | 1 (агент) | 2 (клієнт) | 3 (сервер) | 4 (спонжер) |
|---|---|---|---|---|
| $a$ | 1 | 1 | 0 | 0 |
| $c$ | 1 | 0 | 1 | 0 |

Ролі пар елементів визначимо відносно рядка, в якому знаходиться елемент $a$, на основі відомої моделі клієнт-сервер.

Тоді вважатимемо, що елементи $a_i = 1 \wedge c_i = 0$ виступають у ролі клієнтів, оскільки для того, щоб елемент $a_i$ належав розв'язку, необхідний відповідний елемент $c_j = 1, j \neq i$. Можна сказати, що клієнт $a$ запитує послугу (допустимий шлях) у сервера $c$.

Тоді пара, що відповідає випадку 1, виступає у ролі клієнта і сервера. Таку пару називатимемо *агентом*. Відповідно пару 2 – *клієнтом*, пару 3 – *сервером*. Пара 4 не надає елементів розв'язку і тому є паразитичною. Таку пару називатимемо *спонжером*.

*Зауваження*. Мультимножину будь-яких двох рядків має сенс розбити на мультимножину спонжерів та мультимножину агентів, клієнтів та серверів, оскільки спонжери не надають елементів оптимальним розв'язкам. Таким чином, теорему заборон слід застосовувати для мультимножини агентів, клієнтів та серверів.

**Упорядкування елементів МмЛП.** Побудова МмЛП відбувається на основі двокритеріального упорядкування. Основним критерієм є значення лінійної пари. Другий критерій використовується, якщо значення лінійних пар рівні, але вони виступають у різних ролях. Цей критерій будується з урахуванням того, щоб елементи рівних за значенням лінійних пар, які створюють дозволені і заборонені шляхи, формували кластери. Під кластером розуміємо групу елементів одного типу (наприклад, агенти, рівні за значенням лінійних пар), які зберігаються у мультимножині послідовно.

Тоді, якщо лінійні пари рівні, то: **спонжер > сервер > агент > клієнт**.

**Алгоритм побудови всіх мультимножин лінійних пар.**

Нехай $M$ – розмір матриці $\beta$.

Щоб побудувати всі МмЛП, необхідно для кожного зі сполучень $C_M^2$ пар індексів рядків $(s_1, s_2) \in C_M^2$ сформувати мультимножину лінійних пар.

Код алгоритму має вигляд:

```
for (s1 = 0; s1 < M-1; ++s1)
    for (s2 = s1+1; s2 < M; ++s2)
        normalize_stirng_s(s1,s2)
```

*Зауваження.* Метод normalize_stirng_s(s1,s2) можна реалізувати так, що він будуватиме одну мультимножину для обох пар $(s_1, s_2)$ і $(s_2, s_1)$. У цьому випадку алгоритм пошуку необхідних елементів у МмЛП ускладнюється, оскільки при упорядкуванні елементів враховуються їх ролі.

Обчислювальна складність наведеного алгоритму дорівнює $M^3 \, log(M)$ [1, 2].

**Теорема 2. Нормалізація найбільшого елемента МмЛП**

*Нормалізація* – процедура зменшення значень елементів матриці, при якому оптимальна множина не змінюється. Нормалізація допускається лише для заборонених елементів матриці. Тому теореми заборони і нормалізації тісно пов'язані. По суті, нормалізація має на увазі дві дії: заборону елемента та зменшення його значення.

Нехай:

$\beta$ – початкова матриця;

$Ms$ – мультимножина рядків $r$ та $i$;

$Ms_{max^1} = \beta_{rk} - \beta_{ik}$ – найбільший елемент $Ms$ (найбільша лінійна пара), $k$ – індекс стовпця елемента $Ms_{max^1}$;

$Ms_{max^2}$ – наступний за найбільшим елемент.

Тоді

$$\overline{\beta_{rk}} = \beta_{rk}^{Norm} = \beta_{ik} + Ms_{max^2} + \varepsilon \, . \tag{5}$$

Після застосування теореми 2 значення найбільшого елемента буде зменшене до значення наступного елемента мультимножини. Тому в кожному МмЛП будуть, як мінімум, дві лінійні пари, що відрізняються на нескінченно малу величину $\varepsilon$.

Алгоритм нормалізації найбільшого елемента МмЛП має постійну обчислювальну складність.

**Теорема 3. Нормалізація спонжерів**

Нехай:

$\beta$ – вхідна матриця;

$SpMs$ – мультимножина спонжерів рядків $r$ та $i$ матриці $\beta$;

$SpMs_k = \overline{\beta_{rk}} - \overline{\beta_{ik}}$ – $k$-ий елемент $SpMs$;

$Ms$ – мультимножина серверів, агентів і клієнтів рядків $r$ та $i$;

$Ms_{max}$ – максимальний елемент $Ms$;

Якщо $SpMs_k \geq Ms_{max}$, то:

$$\overline{\beta_{rk}} = \overline{\beta_{ik}} + Ms_{max} + \varepsilon \, . \tag{6}$$

Дана теорема може бути сформульована наступним чином: якщо мультимножина рядків $r$ та $i$ матриці $\beta$ представлена мультимножиною спонжерів та мультимножиною серверів, агентів і клієнтів, то значення всіх спонжерів, які більше за максимальний елемент другої мультимножини, можна зменшити за наведеними формулами.

Обчислювальна складність алгоритму нормалізації спонжерів на підставі теореми 3, у загальному випадку, є лінійною і, таким чином, вона не перевищує обчислювальної складності упорядкування елементів МмЛП. Тому обчислювальна складність загального алгоритму побудови МмЛП та їх нормалізації складає $M^3 \, log(M)$. Цей загальний алгоритм названо нормалізаційним.

**Застосування нормалізаційного алгоритму для розв'язання багатокритеріальної задачі про призначення.** У попередньому розділі статті було показано, що реалізація нормалізаційного алгоритму заснована на побудові МмЛП та нормалізації найбільших елементів МмЛП і спонжерів. Для побудови МмЛП та нормалізації заборонених елементів матриці $\beta^{MM}$ необхідно, щоб для елементів матриці $\beta_{ij}$ були реалізовані наступні операції: арифметичні операції (+,–), оператор присвоєння (=) та оператори порівняння (==,!=,>,>=,<,<=). Реалізація нормалізаційного алгоритму для розв'язання багатокритеріальної задачі потребує математично визначити всі ці операції для елементів $\beta_{ij}$, які у загальному випадку є кортежами, бо можуть містити у собі критерії різних типів. При програмній реалізації елемент $\beta_{ij}$ (вартість призначення) представляється як клас, який містить необхідний набір перевантажених операцій.

**Математичні операції елементів матриці багатокритеріальної задачі про призначення, яка має множину упорядкованих критеріїв.**

Нехай: $a = (a_1, ..., a_i, ..., a_n)$, $b = (b_1, ..., b_i, ...., b_n)$, $c = (c_1, ..., c_i, ...., c_n)$, де $a$, $b$ та $c$ – вектори, $n$ – кількість критеріїв, $a_i$ – $i$-й критерій вектора $a$.

**Додавання** $c = a + b$**.**

$$c_i = a_i + b_i\ ,\ i = \overline{1,n}\ .$$

**Віднімання** $c = a - b$ **.**

$$c_i = a_i - b_i\ ,\ i = \overline{1,n}\ .$$

**Множення** $c = a * b$ **.**

$$c_i = a_i * b_i\ ,\ i = \overline{1,n}\ .$$

**Ділення** $c = a / b$ **.**

$$c_i = \frac{a_i}{b_i}\ ,\ i = \overline{1,n}\ .$$

**Присвоєння** $a = b$ **.**

$$a_i = b_i\ ,\ i = \overline{1,n}\ .$$

**Рівність** $a == b$ **.**

$i = 1$;
while( $i <= n$ AND $a_i == b_i$ ) do $++i$ ;
if( $i == n + 1$ ) then true
else false;

**Менше** $a < b$ **.**

$i = 1$;
while( $i <= n$ AND $a_i == b_i$ ) do $++i$ ;
if( $i == n + 1$ ) then false
else if( $a_i < b_i$ ) then true
else false;

Інші оператори порівняння (!=, <=, >=) будуються за допомогою наведених операцій й відповідних логічних операцій NOT та AND.

**Приклад розв'язання багатокритеріальної задачі про призначення.**

Розглянемо приклад розв'язання трьохкритеріальної задачі про призначення, яка має множину упорядкованих за важливістю критеріїв $\beta_{ij}(c_k)$, $k = \overline{1,3}$. При цьому:

1. $\beta_{ij}(c_1) > \beta_{ij}(c_2) > \beta_{ij}(c_3)$;
2. $\beta_{ij}(c_1) \to min$ , $\beta_{ij}(c_2) \to max$ , $\beta_{ij}(c_3) \to min$ .

У практичних задачах критерії $\beta_{ij}(c_1)$, $\beta_{ij}(c_2)$ й $\beta_{ij}(c_3)$ можуть, наприклад, означати: собівартість роботи, кваліфікацію (чим вище, тим краще), тривалість виконання роботи.

Розглянемо матрицю $\beta^{MM}$, $M = 7$ :

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | {0;1;1} | {1;0;0} | {1;1;0} | {0;1;0} | {0;1;0} | {1;0;0} | {1;1;0} |
| 1 | {1;0;1} | {0;0;1} | {0;1;0} | {0;1;0} | {0;0;0} | {0;1;1} | {0;0;0} |
| 2 | {0;0;0} | {1;0;1} | {1;0;0} | {0;1;1} | {1;1;1} | {0;0;0} | {1;1;1} |
| 3 | {1;0;1} | {0;0;1} | {0;0;1} | {0;1;0} | {0;1;0} | {1;0;1} | {0;1;0} |
| 4 | {0;1;1} | {1;1;0} | {1;1;1} | {1;1;1} | {1;1;0} | {1;1;0} | {0;1;0} |
| 5 | {1;1;0} | {0;1;1} | {1;1;1} | {0;0;1} | {1;1;0} | {0;0;0} | {0;1;0} |
| 6 | {0;1;1} | {1;1;0} | {1;1;1} | {1;1;1} | {1;1;1} | {1;1;1} | {1;0;1} |

Скористаємось твердженням 1 для приведення функціоналу даної задачі до класичного виду (1). Тобто $\beta_{ij}(c_1) \to min$ , $\beta_{ij}(c_2) \to min$ , $\beta_{ij}(c_3) \to min$ . Отримаємо наступну матрицю:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | {0;-1;1} | {1;0;0} | {1;-1;0} | {0;-1;0} | {0;-1;0} | {1;0;0} | {1;-1;0} |
| 1 | {1;0;1} | {0;0;1} | {0;-1;0} | {0;-1;0} | {0;0;0} | {0;-1;1} | {0;0;0} |
| 2 | {0;0;0} | {1;0;1} | {1;0;0} | {0;-1;1} | {1;-1;1} | {0;0;0} | {1;-1;1} |
| 3 | {1;0;1} | {0;0;1} | {0;0;1} | {0;-1;0} | {0;-1;0} | {1;0;1} | {0;-1;0} |
| 4 | {0;-1;1} | {1;-1;0} | {1;-1;1} | {1;-1;1} | {1;-1;0} | {1;-1;0} | {0;-1;0} |
| 5 | {1;-1;0} | {0;-1;1} | {1;-1;1} | {0;0;1} | {1;-1;0} | {0;0;0} | {0;-1;0} |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | {0;-1;1} | {1;-1;0} | {1;-1;1} | {1;-1;1} | {1;-1;1} | {1;-1;1} | {1;0;1} |

Розглянемо приклад МмЛП третього та четвертого рядків.

| стовп. | 0 | 5 | 6 | 1 | 2 | 4 | 3 |
|---|---|---|---|---|---|---|---|
| знач. ЛП | {1;1;0} | {0;1;1} | {0;0;0} | {-1;1;1} | {-1;1;0} | {-1;0;0} | {-1;0;-1} |

У мультимножині є один найбільший елемент у стовпці 0 і наступний за значенням елемент у стовпці 5. За теоремою 1, найбільший елемент слід заборонити, а його значення можна зменшити за теоремою 4.

Звідси $\overline{\beta_{30}} = \beta_{40} + \left(\beta_{35} - \beta_{45}\right) = \{0;-1;1\} + \{0;1;1\} = \{0;0;2\}$.

Операції побудови МмЛП, заборони і нормалізації найбільших елементів слід виконати для всіх сполучень рядків і стовпців матриці (див. алгоритм побудови всіх мультимножин лінійних пар). Якщо у побудованій мультимножині з'являться максимальні спонжери, їх слід нормалізувати за теоремою 3. Виконання перелічених операцій представляє нормалізаційний алгоритм розв'язання задачі.

Після розв'язання задачі нормалізаційним алгоритмом ми одержуємо наступну матрицю:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | {0;-1;-1} | {0;0;-1} | {0;0;1} | {0;-1;0} | {0;-1;0} | {0;0;2} | {0;-1;-2} |
| 1 | {0;-2;-2} | {0;-1;-2} | {0;-1;0} | {0;-2;-1} | {0;-2;-1} | {0;-1;1} | {0;-2;-3} |
| 2 | {0;-1;-3} | {0;0;-3} | {0;0;-1} | {0;-1;-2} | {0;-1;-2} | {0;0;0} | {0;-1;-4} |
| 3 | {0;-1;-1} | {0;0;-1} | {0;0;1} | {0;-1;0} | {0;-1;0} | {0;0;2} | {0;-1;-2} |
| 4 | {0;-1;1} | {0;0;1} | {0;0;3} | {0;-1;2} | {0;-1;2} | {0;0;4} | {0;-1;0} |
| 5 | {0;-2;1} | {0;-1;1} | {0;-1;3} | {0;-2;2} | {0;-2;2} | {0;-1;4} | {0;-2;0} |
| 6 | {0;-1;1} | {0;0;1} | {0;0;3} | {0;-1;2} | {0;-1;2} | {0;0;4} | {0;-1;0} |

Дозволені елементи матриці $\beta^{MM}$ виділені кольором, всі інші невиділені елементи заборонені.

Оптимальні розв'язки:

1. [0,3] [1,2] [2,5] [3,4] [4,6] [5,1] [6,0];
2. [0,4] [1,2] [2,5] [3,3] [4,6] [5,1] [6,0].

Довжина розв'язків:

$S$ ={0;1;0} + {0;1;0} + {0;0;0} + {0;1;0} + {0;1;0} + {0;1;1} + {0;1;1} = {0;6;2}.

**Аналіз побудованої матриці.** Введемо визначення. Два рядки (стовпця) матриці лінійно рівні, якщо в їх МмЛП рівні всі лінійні пари, та нерівні в іншому випадку. У побудованій нормалізаційним алгоритмом матриці всі рядки та стовпці лінійно рівні. Це означає, що всі довершені паросполучення матриці, з точки зору обраного функціоналу, мають однакову довжину {0;6;2}. Тобто не існує шляху поліпшення будь-якого розв'язку, а значить алгоритм знайшов оптимальні розв'язки.

**Висновки.**

1. Нормалізаційний алгоритм можна використовувати для розв'язання задачі про призначення, яка має множину упорядкованих за значимістю критеріїв або функціоналів. Ця задача є узагальненням класичної задачі про призначення.

2. Алгоритм гарантує знаходження за поліноміальний час всіх елементів матриці призначень (оптимальної множини), які формують оптимальні розв'язки. Процедура виводу всіх розв'язків на основі оптимальної множини, у загальному випадку, не є поліноміальною, але один розв'язок може бути отриманий за поліноміальний час $O\left(n^2\right)$ [3, 4].

3. Алгоритм можна використовувати для розв'язання задач, параметрами робіт яких виступають імовірності, тобто деякі властивості таких систем описуються математичними моделями теорії ймовірностей [1].

**ЛІТЕРАТУРА:**


1. *Подоляка О.М.* Про одне узагальнення задачі про призначення // Радіоелектронні і комп'ютерні системи. – Харків: ХАІ, 2007. – № 2(21). – С. 74–82.
2. *Панішев А.В. Подоляка О.М., Подоляка О.О.* Алгоритм пошуку множини всіх розв'язків задачі про призначення // Вісник ЖІТІ. – Житомир: ЖІТІ, 2007. – № 3(42). – С. 120–127.
3. *Новиков Ф.А.* Дискретная математика для программистов. – СПб.: Питер, 2001. – 304 с.
4. *Ахо Альфред В., Хопкрофт Джон Э., Ульман Джеффри Д.* Структуры данных и алгоритмы. –

М.: Изд. дом "Вильямс". – 2000. – 384 с.

ПОДОЛЯКА Олексій Миколайович – старший викладач кафедри інформатики Національного аерокосмічного університету ім. М.Є. Жуковського.

Наукові інтереси:
– дискретна математика і математичне моделювання;
– теорія програмування та алгоритмів.

ПОДОЛЯКА Оксана Олександрівна – кандидат технічних наук, доцент кафедри інформатики Національного автомобільно-дорожнього університету.

Наукові інтереси:
– математичне моделювання;
– теорія розкладів та її застосування.

Подано 05.09.2007