УДК 681.3

О.М. ПОДОЛЯКА

Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Україна

ПРО ОДНЕ УЗАГАЛЬНЕННЯ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

Розглянуто ефективний поліноміальний алгоритм пошуку множини розв'язків задачі про призначення та її узагальнення, у функціоналі якого присутня операція добутку.

задача про призначення, довершене паросполучення, нормалізаційний алгоритм, теорема Форда-Фалкерсона, поліноміальний алгоритм, оптимальний розв'язок

Вступ

Практична цінність представлених результатів міститься в можливості їх використання при розробці систем календарного планування та оперативного управління на транспорті та при розробці систем управління гнучкими автоматизованими системами для підприємств з дискретним характером виробництва.

Постановка проблеми у загальному вигляді

Необхідно знайти довершене паросполучення (перестановку) $\pi^* \in P$ з мінімальною вагою для заданої матриці β^{MM}, де M – порядок матриці:

Функціонал класичної задачі про призначення.

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

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

Функціонал задачі, що вирішується.

$$\rho\left(\pi^*\right) = \min_{\pi} \prod_{i,j \in \pi} \beta_{ij}, \;\; \beta_{ij} > 0 \,, \qquad (2)$$

де: $\rho(\pi) = \prod_{i,j \in \pi} \beta_{ij}$ – вага паросполучення π; β_{ij} – елемент матриці, що належить паросполученню.

Прикладна інтерпретація даної задачі може бути наступною. Як і у класичній задачі про призначення, в задачі (2) є M паралельно функціонуючих неідентичних машин та M робіт, які необхідно закріпити за певними машинами. Кожна машина може виконувати лише одну роботу. Загальне завдання (побудова довершеного паросполучення) полягає у виконанні M підзадач на M різних машинах.

У прикладних задачах, що відповідають функціоналу (2), елемент матриці β_{ij} може означати імовірність виконання/невиконання загального завдання при призначенні i-ї роботи на j-ту машину. Тоді розв'язок задачі представлятиме собою добуток відповідних ймовірностей β_{ij}.

Слід зазначити, що в основі більшості алгоритмів розв'язання класичної задачі (1) лежить теорема Форда-Фалкерсона про максимальний потік і мінімальний розріз у мережі дводольного графа [1, 2, 3]. Фундаментальним законом теорії потоків є закон збереження потоку (що надійшло, те і відійшло). Задача (2) не може бути розв'язана алгоритмом пошуку максимального потоку, тому що в ній не виконується закон збереження потоку, оскільки у функціоналі присутня операція добутку (потік у мережі не може ділитися або множитися). Отже, з точки зору теорії мереж, задачі (1) і (2), по своїй сутності, є принципово різними. Задача про призначення у постановці (2) раніше не розглядалася.

У даній роботі представлено вперше запропонований поліноміальний алгоритм, за допомогою якого можна розв'язати як задачу (1), так і задачу (2). Ідея алгоритму полягає у пошуку множини елементів матриці β^{MM}, які належать множині оптимальних розв'язків. Ці елементи утворюють *оптимальну множину*.

В основі алгоритму покладено теореми заборон і нормалізації, які будуть розглянуті далі.

Обчислювальна складність алгоритму становить $M^3 Log(M)$.

Очевидні твердження

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.

Твердження 1. Заміна двох елементів розв'язку

Розглянемо рисунок, на якому схематично представлена матриця β.

рядок \ стовпець	*b*		*d*
a	$\beta_{a,b}$	…	$\beta_{a,d}$
…	…		…
c	$\beta_{c,b}$	…	$\beta_{c,d}$

Рис. 1. Перехресна заміна

Якщо в оптимальному розв'язку матриці β 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. Вона давно відома і зазвичай використовується при розв'язанні задачі комівояжера і задачі про призначення [1, 4, 5].

Слідство твердження 1

Якщо $\beta_{a,d} \wedge \beta_{c,b}$ – елементи оптимального розв'язку, а $\beta_{a,b} \wedge \beta_{c,d}$ – елементи, що замінюються, то

$\beta_{a,b} \wedge \beta_{c,d} \le \beta_{a,d} \wedge \beta_{c,b}$;

$\beta_{a,b} + \beta_{c,d} \le \beta_{a,d} + \beta_{c,b}$, для задачі (1);

$\beta_{a,b} * \beta_{c,d} \le \beta_{a,d} * \beta_{c,b}$, для задачі (2);

або для задачі (1):

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

або для задачі (2):

$$\frac{\beta_{a,b}}{\beta_{c,b}} \le \frac{\beta_{a,d}}{\beta_{c,d}}. \tag{4}$$

Зауваження. В матриці β^{MM} можуть існувати підматриці порядку $K \in [3, M]$, в яких можуть існувати заміни елементів по K [5].

Лінійні пари

Лінійною парою (ЛП) i-го та k-го рядків матриці називатимемо елемент, який обчислюється за наступними формулами.

Для задачі (1):

$$D_j = \beta_{ij} - \beta_{kj},\ j \in [0, M-1]. \tag{5}$$

Для задачі (2):

$$D_j = \frac{\beta_{ij}}{\beta_{kj}},\ j \in [0, M-1], \beta_{kj} > 0. \tag{6}$$

Елемент β_{ij} називатимемо клієнтським значенням лінійної пари, а β_{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 по незростанню значень елементів.

Зауваження. З формули (4) випливає, що для розв'язання задачі (2) необхідно, щоб елементи матриці β були більше нуля. Задача (1) не має обмежень на значення 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.

Визначення найменшого і мінімального значення вводиться так само.

Теорема 1. Заборон

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

Теорема 2. Заборон

Якщо клієнтське значення β_{rj} є максимальним в M різних МмЛП матриці β^{MM}, то елемент β_{rj} не може формувати оптимальний розв'язок задачі і, отже, може бути заборонений.

Зауваження. Дану теорему має сенс застосовувати, якщо є багато МмЛП, в яких є множина однакових максимальних елементів. Такому випадку відповідає матриця з множиною елементів, що повторюються, які знаходяться в прямокутних підматрицях.

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 β^{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 лінійних 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, використовуючи відому архітектуру 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, теореми 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 різній ролі. 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 – кількість рядків, а N – кількість стовпців матриці.

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

Код алгоритму має такий вигляд:

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

Зауваження: Метод test_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 ураховуються їх ролі.

Обчислювальна складність алгоритму побудови мультимножин лінійних пар

Обчислювальна складність побудови сполучень всіх рядків матриці по два становить

$$C_M^2 = \frac{1}{2} M(M-1) = O(M^2) \text{ [2].}$$

Обчислювальна складність побудови окремого МмЛП становить $M \log(M)$.

Обчислювальна складність алгоритму дорівнює $M^3 \log(M)$.

Теорема 3. Нормалізації забороненого елемента матриці

Нормалізація – процес 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:

β – вхідна матриця і $\overline{\beta_{rc}}$ – її заборонений елемент, який знаходиться в рядку r та в стовпці c;

β_{rj} – дозволені елементи рядка r, $j \in [0, N-1]$;

β_{ic} – дозволені елементи стовпця c, $i \in [0, M-1]$;

ε – нескінченно мала величина.

Тоді для задачі (1):

$$\begin{aligned}\overline{\beta_{rc}} &= \beta_{rc}^{Norm} = \\ &= \max_{\substack{i,j \\ \beta_{rj}+\beta_{ic}<\beta_{rc}+\beta_{ij}}} \left[\beta_{rj} + \beta_{ic} - \beta_{ij}\right] + \varepsilon = \qquad (7)\\ &= \beta'_{rj} + \beta'_{ic} - \beta'_{ij} + \varepsilon \quad .\end{aligned}$$

Для задачі (2):

$$\begin{aligned}\overline{\beta_{rc}} &= \beta_{rc}^{Norm} = \\ &= \max_{\substack{i,j \\ \beta_{rj}+\beta_{ic}<\beta_{rc}+\beta_{ij}}} \left[\beta_{rj} * \frac{\beta_{ic}}{\beta_{ij}}\right] + \varepsilon = \qquad (8)\\ &= \beta'_{rj} * \beta'_{ic} / \beta'_{ij} + \varepsilon \quad .\end{aligned}$$

Елемент β'_{ij} назвемо *нормалізатором*. За індексами елемента (r, c) і його нормализатора (i, j) можна визначити нормалізоване значення β_{rc}^{Norm}. Елемент $\overline{\beta_{rc}}$ може мати кілька нормалізаторів, оскільки у матриці може бути декілька рівних максимальних значень, що задовольняють наведеним вище формулам.

Слід зазначити, що за ознаку наявності нескінченно малої складової у елемента матриці може виступати логічне значення, яке використовується для позначення дозволу цього елемента.

Твердження 2. Допустимий діапазон значень забороненого елемента

Нехай елемент $\overline{\beta_{rc}} = a$ є забороненим. Якщо значення $\overline{\beta_{rc}}$ належатиме інтервалу $[a, \infty]$, то оптимальна множина не зміниться. Інакше кажучи, значення забороненого елемента можна збільшити у разі необхідності.

Значення забороненого елемента матриці є сенс збільшувати, якщо необхідно зменшити значення лінійної пари МмЛП, в якій він є серверним значенням. Такі дії дозволяють побудувати еквівалентну МмЛП з меншим значенням найбільшого або максимального елемента.

Твердження 3. Дозвіл забороненого елемента

Якщо дозволити елемент $\overline{\beta_{rc}} = a + \varepsilon$ і зменшити на ε його значення, то довжина оптимального розв'язку задачі не зміниться.

Теорема 4. Нормалізація найбільшого елемента МмЛП

Нехай:

β – 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 Ms.

Тоді для задачі (1):

$$\overline{\beta_{rk}} = \beta_{rk}^{Norm} = \beta_{ik} + Ms_{\max^2} + \varepsilon . \tag{9}$$

Для задачі (2):

$$\overline{\beta_{rk}} = \beta_{rk}^{Norm} = \beta_{ik} * Ms_{\max^2} + \varepsilon . \tag{10}$$

Після застосування теореми 4 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 ε.

Алгоритм нормалізації найбільшого елемента МмЛП має постійну обчислювальну складність.

Теорема 5. Нормалізація спонжерів

Нехай:

β – вхідна матриця;

$SpMs$ – мультимножина спонжерів рядків r та i матриці β;

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

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

$Ms_{\max}$ – максимальний елемент Ms.

Якщо $SpMs_k \ge Ms_{\max}$, то

для задачі (1):

$$\overline{\beta_{rk}} = \overline{\beta_{ik}} + Ms_{\max} + \varepsilon . \tag{11}$$

для задачі (2):

$$\overline{\beta_{rk}} = \overline{\beta_{ik}} * Ms_{\max} + \varepsilon . \tag{12}$$

Дана теорема може бути сформульована наступним чином: якщо мультимножина рядків r та i матриці β 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 5, в загальному випадку, є лінійною, і, значить, 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 назвемо нормалізаційним.

Приклад розв'язання задачі про призначення (1)

Розглянемо матрицю β^{MM}, $M = 5$:

	0	1	2	3	4
0	12	22	17	5	18
1	22	6	20	6	21
2	17	16	23	9	24
3	11	12	11	19	21
4	24	8	21	10	10

Розглянемо побудову МмЛП задачі (1).

Виконаємо побудову МмЛП нульового і першого рядків. Лінійні пари цих рядків матимуть наступний вигляд:

індекс стовпця	0	1	2	3	4
знач. пари	–10	16	–3	–1	–3

Тепер відсортуємо по незростанню отримані лінійні пари. МмЛП матиме вигляд:

індекс стовпця	1	3	2	4	0
знач. пари	16	–1	–3	–3	–10

У мультимножині є один найбільший елемент в стовпці 1, і наступний за значенням елемент у стовпці 3. По теоремі 1 найбільший елемент слід заборонити, а його значення можна зменшити по теоремі 4.

Звідси $\overline{\beta_{01}} = \beta_{11} + (\beta_{03} - \beta_{13}) = 6 - 1 = 5$.

Операції побудови МмЛП, заборони і нормалізації найбільших елементів слід виконати для всіх сполучень рядків і стовпців матриці (див. алгоритм побудови всіх мультимножин лінійних різниць). Якщо у побудованій мультимножині з'являться максимальні спонжери, їх слід нормалізувати по теоремі 5.

Виконання перелічених операцій представляє нормалізаційний алгоритм розв'язання задачі.

Після розв'язання задачі нормалізаційним алгоритмом ми одержуємо наступну матрицю Mn_1.

12(1)	11(0)	17(0)	6(0)	18(0)
7(0)	6(1)	12(0)	1(0)	13(0)
15(0)	14(0)	20(0)	9(1)	21(0)
6(0)	5(0)	11(1)	0(0)	12(0)
4(0)	3(0)	9(0)	-2(0)	10(1)

В кожній комірці вказано значення елемента, а у круглих дужках – його дозвіл (0 – елемент заборонений, 1 – дозволений).

Оптимальний розв'язок:

{ (0,0), (1,1), (2,3), (3,2), (4,4) }.

Сума елементів оптимального розв'язку:

$$S_1 = 12 + 6 + 9 + 11 + 10 = 48.$$

Добуток елементів оптимального розв'язку:

$$P_1 = 12 \cdot 6 \cdot 9 \cdot 11 \cdot 10 = 71280.$$

Приклад розв'язання узагальнення задачі про призначення (2)

Особливості програмної реалізації задачі

Лінійні пари задачі (2) є частками елементів матриці. Якщо для представлення часток у програмі використовувати стандартні дійсні типи даних, це може призвести до накопичення похибки множення/ділення і, як слідство, до заборони елементів, які можуть належати оптимальним розв'язкам.

Існує два способи вирішення вказаної проблеми.

Перший полягає у додаванні малого коефіцієнта ε, який компенсує втрату точності. У цьому разі алгоритм розв'язання задачі буде поліноміальним і дозволить знайти всі розв'язки задачі з точністю до ε. А, отже, має сенс його застосовувати для розв'язання практичних задач.

Другий спосіб полягає у використанні в програмі класу раціональних чисел, який ґрунтується на арифметиці великих чисел. Слід зазначити, що реалізація подібного класу є складною, і за рахунок використання довгої арифметики, алгоритм розв'язання задачі (2) не буде поліноміальним. Тим не менш, саме така реалізація алгоритму є суворою з математичної точки зору, а тому вона й була використана для розв'язання даної задачі.

Лінійні пари рядків 0 та 1:

стовпець	0	1	2	3	4
знач. ЛП	6/11	11/3	17/20	5/6	6/7

Побудуємо МмЛП.

стовпець	1	4	2	3	0
знач. ЛП	11/3	6/7	17/20	5/6	6/11

За теоремою 1 найбільший елемент слід заборонити, а його значення можна зменшити за теоремою 4.

Звідси

$$\overline{\beta_{01}} = \beta_{11} \cdot (\beta_{04} / \beta_{14}) = 6 \cdot 6/7 = 36/7 .$$

Після розв'язання задачі нормалізаційним алгоритмом ми одержуємо наступну матрицю Mn_2 .

$\frac{12390875}{1168032}(0)$	$\frac{8575000}{1156923}(0)$	$\frac{30625}{2187}(0)$	**5(1)**	$\frac{4287500}{385641}(0)$
$\frac{632043}{73600}(0)$	**6(1)**	$\frac{1587}{140}(0)$	$\frac{3470769}{857500}(0)$	9(0)
17(1)	$\frac{147200}{12393}(0)$	$\frac{1946720}{86751}(0)$	$\frac{1168032}{145775}(0)$	$\frac{73600}{4131}(0)$
$\frac{16222437}{1946720}(0)$	$\frac{3080}{529}(0)$	**11(1)**	$\frac{24057}{6125}(0)$	$\frac{4620}{529}(0)$
$\frac{70227}{7360}(0)$	$\frac{20}{3}(0)$	$\frac{529}{42}(0)$	$\frac{385641}{85750}(0)$	**10(1)**

Оптимальний розв'язок:

{ (0,3), (1,1), (2,0), (3,2), (4,4) }.

Сума елементів оптимального розв'язку:

$$S_2 = 5 + 6 + 17 + 11 + 10 = 49.$$

Добуток елементів оптимального розв'язку:

$$P_2 = 5 \cdot 6 \cdot 17 \cdot 11 \cdot 10 = 56100.$$

Порівняємо розв'язки задач (1) и (2).

$$S_2 - S_1 = 49 - 48 = 1 .$$

$$P_1 - P_2 = 71280\text{-}56100 = 15180 .$$

$$\frac{P_1}{P_2} = \frac{71280}{56100} \approx 1{,}270588 .$$

Аналіз матриць, побудованих алгоритмом

Введемо визначення.

Два рядка (стовпця) матриці лінійно рівні, якщо в МмЛП рівні всі лінійні пари, та нерівні в іншому випадку.

У матриць Mn_1 та Mn_2 всі рядки та стовпці лінійно рівні. Це означає, що всі перестановки елементів матриці еквівалентні, тобто мають однакову довжину з точки зору обраного функціоналу.

Твердження 4. Еквівалентне перетворення матриці задачі

Для задачі (1):

Якщо до кожного елемента довільного рядка/стовпця матриці додати константу, то оптимальний розв'язок задачі не зміниться.

Для задачі (2):

Якщо кожний елемент довільного рядка/стовпця матриці задачі (2) помножити/поділити на додатну константу, то оптимальний розв'язок задачі не зміниться.

Твердження 5. Симетрія задач на мінімум та максимум

Для задачі (1):

$$\rho(\pi^*) = \max_{\pi} \sum_{i,j \in \pi} \beta_{ij} = \min_{\pi} \sum_{i,j \in \pi} (-\beta_{ij}) . \qquad (13)$$

Для задачі (2):

$$\rho(\pi^*) = \max_{\pi} \prod_{i,j \in \pi} \beta_{ij} = \min_{\pi} \prod_{i,j \in \pi} \left(\frac{1}{\beta_{ij}} \right) . \qquad (14)$$

Приклади розв'язання задач (1) і (2) на максимум

Щоб розв'язати задачу на максимум за допомогою алгоритму пошуку розв'язку мінімальної довжини, скористаємося ствердженням 6.

Після завершення роботи алгоритму отримаємо наступні матриці та розв'язки.

Задача (1).

–31(0)	–22(1)	–29(0)	–17(0)	–30(0)
–22(0)	–13(0)	–20(1)	–8(0)	–21(1)
–25(0)	–16(0)	–23(1)	–11(0)	–24(1)
–33(0)	–24(0)	–31(0)	–19(1)	–32(0)
–24(1)	–15(0)	–22(0)	–10(0)	–23(0)

Оптимальні розв'язки:

{ (0,1), (1,2), (2,4), (3,3), (4,0) };

{ (0,1), (1,4), (2,2), (3,3), (4,0) };

Задача (2).

$\frac{7}{242}(0)$	$\frac{1}{22}(1)$	$\frac{8}{253}(0)$	$\frac{42}{605}(0)$	$\frac{1}{33}(0)$
$\frac{1}{22}(0)$	$\frac{1}{14}(0)$	$\frac{8}{161}(0)$	$\frac{6}{55}(0)$	$\frac{1}{21}(1)$
$\frac{7}{176}(0)$	$\frac{1}{16}(0)$	$\frac{1}{23}(1)$	$\frac{21}{220}(0)$	$\frac{1}{24}(0)$
$\frac{5}{228}(0)$	$\frac{55}{1596}(0)$	$\frac{220}{9177}(0)$	$\frac{1}{19}(1)$	$\frac{55}{2394}(0)$
$\frac{1}{24}(1)$	$\frac{11}{168}(0)$	$\frac{22}{483}(0)$	$\frac{1}{10}(0)$	$\frac{11}{252}(0)$

Оптимальний розв'язок:

{ (0,1), (1,4), (2,2), (3,3), (4,0) }.

Висновки

1. Алгоритм можна використовувати для розв'язання задачі про призначення, яка має множину упорядкованих за значимістю критеріїв або функціоналів.

У цьому випадку елемент матриці призначень являє собою складний тип, тому необхідно перевантажити оператори порівняння та присвоювання для цих елементів.

2. Алгоритм гарантує знаходження за полиноміальний 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.

3. Пошук одного розв'язку задачі на основі оптимальної множини може бути виконаний за поліноміальний час. Для знаходження одного розв'язку можна використати алгоритми побудови довершеного паросполучення на основі елементів оптимальної множини [1, 3, 5].

Література

1. Кофман А. Введение в прикладную комбинаторику. – М.: Наука, 1975. – 480 с.

2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.:МЦНМО, 2000. – 956 с.

3. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2001. – 304 с.

4. Панішев А.В., Подляка О.О. Одне з узагальнень задачі про призначення з обмеженнями // Вісник ЖІТІ. – Житомир: ЖІТІ, 1999. – №11. – С. 139-144.

5. Ахо Альфред В., Хопкрофт Джон Э., Ульман Джеффри Д. Структуры данных и алгоритмы. – М.: изд. дом "Вильямс".– 2000. – 384 с.

Надійшла до редакції 7.05.2007

Рецензент: д-р техн. наук, проф. О.Ю. Соколов, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАИ», Харків.