УДК 519.68

О.В.Демидович, Г.Г.Цегелик

Узагальнена задача про призначення та методи її розв'язування

Розглянемо узагальнену задачу про призначення в постановці [3,6]

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} Q_{ij} x_{ij} \to \min \tag{1}$$

за умов

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1,2,...,m); \tag{2}$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \le b_j \quad (j = 1,2,...,n); \tag{3}$$

$$x_{ij} \in \{0 \cup 1\} \quad (i = 1,2,...,m; j = 1,2,...,n), \tag{4}$$

де Q_{ij}, a_{ij}, b_j — невід'ємні цілі константи; x_{ij} — шукані змінні.

У праці [5] доведено, що узагальнена задача про призначення належить до класу NP-повних задач і для її розв'язування невідомий ніякий поліноміальний алгоритм. Дана задача виникає при математичному моделюванні оптимального розподілу інформаційних ресурсів серед вузлів інформаційно-обчислювальних мереж [4]. Як відомо, при практичному застосуванні точний розв'язок задачі не завжди є необхідним. Тому залишається актуальною розробка алгоритмів, які дають змогу знайти наближений розв'язок за обмежену кількість кроків. Нижче запропонований евристичний алгоритм розв'язування задачі (1)-(4), за допомогою якого можна знайти наближений розв'язок за кількість ітерацій не більш ніж (n-1)m. Крім цього, пропонується модифікація даного алгоритму, яка забезпечує в усіх випадках знаходження кращого розв'язку, але потребує більшого обсягу обчислень.

Зазначимо, що задача (1)-(4) не завжди є розв'язною. Однак у деяких випадках відсутність розв'язку задачі можна визначити наперед.

Нехай

$$k_i = \min_{1 \le j \le n} a_{ij} \quad (i=1,2,...,m).$$

Визначимо функцію

$$n(k_1,\dots,k_m;b_2,\dots,b_n)=\max_{1\le i\le m}\Big(\sum_{j=1}^{i}k_j+\sum_{l=2}^{n}\min(b_l,k_i-1)\Big).$$

Тоді, якщо

$$n(k_1,\dots,k_m;b_2,\dots,b_n)>\sum_{i=1}^{n}b_i\ge\max_{2\le j\le n}n(k_1,\dots,k_m;b_2,\dots,b_j-1,\dots,b_n),$$

то розв'язок задачі не існує. Цей факт випливає з наслідку (2.3) теореми доведеної в [1].

Опишемо евристичний алгоритм знаходження наближеного розв'язку задачі (1)-(4).

Алгоритм. Він складається з двох етапів. На першому етапі знаходимо початковий розподіл $X=\left\{x_{ij}\right\}_{m,n}$, який завжди буде оптимальним, якщо не враховувати умови (3). Цей етап складається з таких кроків:

1. Для всіх індексів i ($i=1,2,\dots,m$) визначаємо $\min\limits_{1\le s\le n}Q_{is}$. Нехай

$$\min_{1\le s\le n}Q_{is}=Q_{is_i}\qquad(i=1,2,\dots,m).$$

2. Знаходимо початковий розподіл, тобто визначаємо матрицю $X=\left\{x_{ij}\right\}_{m,n}$, де

$$x_{is_i}=1\quad(i=1,2,\dots,m),$$

$$x_{ij}=0\ \ (i=1,2,\dots,m;\ j=1,2,\dots,n;\ (i,j)\ne(i,s_i)).$$

На другому етапі переходимо до нового розподілу, якщо для початкового розподілу існує хоча б один індекс $j=r$ такий, що умова (3) не виконується. На кожній ітерації другого етапу одному з елементів x_{ij} матриці X, який має значення 1 і міститься в j-му стовпчику такому що для індекса j не виконується умова (3), присвоюється значення 0. Натомість, одному з елементів i-го рядка, який має значення 0, присвоюється значення 1.

Другий етап складається з таких кроків:

1. Формуємо вектор ознак $E=(\varepsilon_1,\varepsilon_2,\dots,\varepsilon_n)$, де $\varepsilon_j=0$ ($j=1,2,\dots,n$).

Елементи вектора ознак, що дорівнюють 0, визначають ті стовпці матриці X, елементи яких можна змінювати, а елементи вектора ознак, що дорівнюють 1, визначають ті стовпці матриці X, які далі змінюватись не будуть.

2. Шукаємо такий індекс j, для якого умова (3) не виконується. Якщо такий індекс не знайдений, переходимо до пункту 7. Якщо ж такий індекс існує, позначаємо його через r і переходимо до пункту 3.

3. Визначаємо множину $\Phi_r = \{s: s \neq r, \varepsilon_s = 0\}$. Якщо множина $\Phi_r = \varnothing$, то розв'язок знайти не вдалось, і переходимо до пункту 8. Якщо ж $\Phi_r \neq \varnothing$, то для кожного i, для якого $x_{ir} = 1$, знаходимо $\min\limits_{s \in \Phi_r} (Q_{is} - Q_{ir})$. Нехай $\min\limits_{s \in \Phi_r} (Q_{is} - Q_{ir}) = Q_{is_i} - Q_{ir}$. Переходимо до пункту 4.

4. Знаходимо $\min\limits_{i} (Q_{is_i} - Q_{ir})$, де мінімум береться за тими індексами i, для яких $x_{ir} = 1$. Нехай $\min\limits_{i} (Q_{is_i} - Q_{ir}) = Q_{ls_l} - Q_{lr}$.

5. У матриці X приймаємо $x_{lr} = 0$, $x_{ls_l} = 1$.

6. Перевіряємо умову (3) для $j=r$. Якщо вона не виконується, то переходимо до пункту 3, інакше присвоюємо елементові ε_r вектора E значення 1 (тобто r-й стовпчик матриці X вилучаємо з подальшого розгляду) і переходимо до пункту 2.

7. X — знайдений розподіл.

8. Кінець.

Таким чином, алгоритм дає змогу за обмежену кількість кроків знайти наближений розв'язок задачі (1)-(4).

Визначимо максимально можливу кількість кроків. Оскільки для кожного знайденого r існує не більш ніж m елементів x_{lr}, яким можна присвоїти нульове значення (виконуючи пункт 5 алгоритму), а всіх індексів r, для яких не виконується умова (3), може бути не більш ніж n (причому після перегляду (n-1)-го індексу множина $\Phi_r = \varnothing$), то всього ітерацій може бути не більш ніж $m(n-1)$.

Проаналізуємо запропонований алгоритм з погляду отримання оптимального розв'язку. Пункти 2-6 для фіксованого $j=r$ здійснюють перехід до нового розподілу X, для якого умова (3) виконується. При цьому бажано, щоб значення цільової функції було якнайменшим. У запропонованому вище алгоритмі це досягається шляхом послідовного виконання при фіксованому $j=r$ пунктів 3-6 для кожного i, для якого $x_{ir} = 1$.

Розглянемо окрему задачу переходу від існуючого розподілу змінних x_{ij} для фіксованого $j=r$ до нового розподілу X, для якого умова (3) виконується. У запропонованому нижче алгоритмі цей перехід здійснюється за один крок, тобто всі i, для яких $x_{ir}=1$ розглядаються одночасно. Далі буде показано, що така задача еквівалента задачі про рюкзак, а тому послідовність ітерацій 3-6 алгоритму 1 можуть дати лише її наближений розв'язок.

Отже, можна запропонувати таку модифікацію алгоритму, а саме його другого етапу.

1. Формуємо вектор ознак $E=(\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n)$, де $\varepsilon_j=0$ (j=1,2,...,n).

2. Шукаємо такий індекс j, для якого умова (3) не виконується. Якщо такий індекс не знайдений, переходимо до пункту 7. Якщо ж такий індекс існує, позначаємо його через r і переходимо до пункту 3.

3. Для знайденого індексу $j=r$ визначаємо набір $i_1,\ldots,i_l$, такий, що $x_{i_1r}=\ldots=x_{i_lr}=1$.

4. Визначаємо множину $\Phi_r=\{s: s\neq r, \varepsilon_s=0\}$. Якщо $\Phi_r=\varnothing$, то розв'язок знайти не вдалося, і переходимо на пункт 8. Якщо ж $\Phi_r\neq\varnothing$, то для кожного індексу i_k (k=1,2,...,l), визначаємо індекс p_k такий, що

$$Q_{i_kp_k}-Q_{i_kr}=\min_{s\in\Phi_r}\left(Q_{i_ks}-Q_{i_kr}\right).$$

Позначаємо $c_{i_k}=Q_{i_kp_k}-Q_{i_kr}$, ($k$=1,2,...,$l$) і вводимо змінні $y_k\in\{0\cup1\}$, де $y_k=1$, якщо x_{i_kr}=1 не зміниться при переході до нового розподілу, і $y_k=0$, якщо в новому розподілі приймемо, що x_{i_kr}=0.

Тоді умову мінімального збільшення значення цільової функції (1) при переході до нового розподілу, в якому для даного $j=r$ умова (3) виконується, можна записати у вигляді

$$\min\left\{\sum_{k=1}^{l}c_{i_k}(1-y_k)\left|\sum_{k=1}^{l}a_{i_k}y_k\le b_r;\, y_k\in\{0\cup1\}, k=1,2,\ldots,l\right.\right\},$$

або

$$\max\left\{\sum_{k=1}^{l}c_{i_k}y_k\left|\sum_{k=1}^{l}a_{i_k}y_k\le b_r;\, y_k\in\{0\cup1\}, k=1,2,\ldots,l\right.\right\}. \quad (5)$$

Задача (5) є лінійною задачею про рюкзак. Для її розв'язування можна використати метод динамічного програмування з попереднім зменшенням розмірності задачі, запропонований у [2].

5. Розв'язуємо задачу (5) і здійснюємо перехід до нового розподілу. Для цього, для кожного елемента $y_k = 0$ розв'язку задачі (5) приймаємо $x_{i_k r}$=0 і $x_{i_k p_k}$ =1. У такий спосіб отримуємо новий розподіл X, при якому умова (3) виконується для індексу $j=r$.

6. Приймаємо $\varepsilon_r = 1$ і переходимо до пункту 2.

7. X — знайдений розподіл.

8. Кінець.

1. Баранов В. И., Стечкин Б. С. Экстремальные комбинаторные задачии их приложения. М.,1989. 2. Марданов С.С. Сокращение размерности целочисельной задачи о ранце и ее решение паралельным алгоритом динамического програмрования.//Дискрет. математика. 1992. Т4. № 2. С.32-38. 3. Мину М. Мамематическое програмирование. Теория и алгоритмы. М., 1990. 4. Цегелик Г. Г. Системы распределенных баз данных. Львов, 1990. 5. Nemhauser G. L., Wosley L. A. Combinatorial and Integer Programming. John Wiley, New York, 1988. 6. Legedre J. P., Minoux M. Une application de dualite rn programmation en nobers entiers-selection et affections optimalis d'une flotte d'avions.// R.A.I.R.O. 1977. Vol.11, №2, P 201-222.

Стаття надійшла до редколегії 04.03.96

УДК 681.3.06

М.Ю. Щербина

Реалізація Outline з використанням Microsoft Visual C++ 1.51 і бібліотеки класів Microsoft Foundation Class*

Компілятор Microsoft Visual C++ 1.51 є потужним засобом для розробки програм під Microsoft Windows. Бібліотека класів Microsoft Foundation Class (MFC) і наявні засоби візуального програмування дають змогу спростити і прискорити процес створення програм. Проте MFC орієнтована на використання стандартних Windows-компонент, таких, як Button, Edit, ListBox і т.п. Фактично MFC можна вважати заміною Windows SDK.

---

* Ця робота була частково підтримана Міжнародною Соросівською програмою підтримки освіти в галузі точних наук (ISSEP), грант № U051215.