УДК 519.863:623.618(477)

А. І. СБІТНЄВ,
доктор технічних наук, професор,
дійсний член Міжнародної академії комп'ютерних наук та систем,
заслужений діяч науки і техніки України,
професор Національного університету оборони України
В. В. КОЗЛОВ,
кандидат технічних наук, доцент,
завідувач кафедри інформаційних систем і технологій,
Національна академія статистики, обліку та аудиту

Формулювання задачі розподілу взаємопов'язаних ресурсів як задачі математичного програмування

Показано, що у загальній постановці задача про розподіл взаємопов'язаних ресурсів є повною задачею нелінійного програмування, а частковим випадком задачі розподілу взаємопов'язаних ресурсів є задача про максимальний потік, яка успішно вирішується наведеним у роботі алгоритмом.

Ключові слова: *алгоритм, пов'язані ресурси, розподіл ресурсів, локальний екстремум.*

Постановка проблеми. В роботі [1] сформульовано задачу розподілу взаємопов'язаних ресурсів (далі – задача РВР) і наведено основні елементи побудови алгоритму її розв'язання. Постановку цієї задачі здійснено у рамках загальної моделі математичного програмування. Її зміст ми наводимо нижче зі зміною термінів "ціна" й "вага" для узгодження використання останнього терміна з джерелами [2–8].

Задано множину ресурсів $x_i \in X$, де $i = 1, 2, ..., n$. На множині X задано функцію F, яка відображує множину на числову вісь, тобто кожному ресурсу x_i ставиться у відповідність число $F(x_i) = f_i$ – "ціна" ресурсу. Зв'язок між ресурсами визначається відношенням $\Gamma \subseteq X \times X$, на якому вводиться функція W, що ставить у відповідність пов'язаній парі (x_i, x_j) певне число $W(x_i, x_j) = w_{ij}$ – "вагу" зв'язку. *Необхідно розподілити ресурси між К об'єктами O_j з приймальними спроможностями p_j, де $j = 1, K$, так, щоб сумарна ціна ресурсів, що потрапили до будь-якого об'єкта, не перевищувала його приймальної спроможності, а сумарна вага зв'язків між ресурсами, що потрапили до різних об'єктів, була мінімальною.*

Мета дослідження – показати, що загальна постановка задачі про розподіл взаємопов'язаних ресурсів є повною задачею нелінійного програмування, а частковим випадком задачі розподілу взаємопов'язаних ресурсів є задача про максимальний потік, яка успішно вирішується наведеним у роботі алгоритмом.

Виклад основного матеріалу. Як і в [1], розглянемо випадок $K = 2$, тобто елементи множини X треба розподілити між двома множинами A і B, $p_1 = p(A)$, $p_2 = p(B)$.

Введемо на X характеристичну функцію y, що відображає належність елемента x_i до множини A або B:

$$y_i = \begin{cases} 0 & \text{при } x_i \in A \\ 1 & \text{при } x_i \in B. \end{cases}$$

Виконання обмежень щодо приймальних спроможностей множин A і B призводить до системи нерівностей:

$$\sum_{i=1}^{n} f_i y_i \le p(B) \qquad \sum_{i=1}^{n} f_i (1 - y_i) \le p(A).$$

Критерій мінімальності "розірваних" зв'язків (між ресурсами, що потрапили до різних множин) має вигляд:

$$\sum_{i=1}^{n}\sum_{j<i} w_{ij}\left|y_i - y_j\right| \to \min$$

або

$$\sum_{i=1}^{n}\sum_{j<i} w_{ij}\, y_i(1-y_j) + \sum_{i=1}^{n}\sum_{j<i} w_{ij}(1-y_i)\,y_j \to \min.$$

Таким чином, маємо задачу нелінійного програмування з булівськими змінними.

Число варіантів при використанні повного перебору дорівнює двійковому числу $y_1, y_2, ..., y_{n-1}, y_n$, тобто дорівнює 2^n. Тобто ми маємо так звану повну задачу.

Якщо вилучити з умови задачі необхідність обліку "ціни" ресурсів при припущенні $p_1=p_2=\infty$, тобто при знятті обмежень на приймальні можливості об'єктів (множин A й B), то зведена задача відповідатиме класу задач теорії графів, відомих як **задача про максимальний потік** [2–8]:

Задано граф $G = (X, U)$, дугам якого приписані дійсні числа, що називаються пропускними здатностями. У графі виділено дві вершини, одна з яких називається *джерелом*, а друга — *стоком*. Уздовж дуг графа можуть протікати потоки. Пропускну спроможність дуги (x_i, x_j) позначимо через q_{ij}, величину потоку уздовж її — через z_{ij}. Позначимо також джерело буквою s, а стік — буквою t.

Нехай вершини графа розбиті на дві непересічні множини X^1 і X^2. Множина дуг, кожна з яких бере початок у X^1, а закінчується в X^2, називається *розрізом* графа С. *Пропускною спроможністю* розрізу будемо називати суму пропускних спроможностей дуг, що утворять цей розріз. *Мінімальний розріз* — це розріз з мінімальною пропускною спроможністю. Говорять, що розріз *відокремлює* дві вершини x_i, і x_j, якщо $x_i \in X^1$, а $x_j \in X^2$.

Загальноприйнятий алгоритм находження максимального потоку слідує з доказу "теореми про максимальний потік і мінімальний розріз", що формулюється в такий спосіб [9]:

Теорема. Величина максимального потоку від джерела s до стоку t дорівнює мінімальному значенню величини розрізів, що відокремлюють s від t.

Спростимо цю задачу. Розглянемо випадок, коли має місце $\forall ij(w_{ij}=1)$, тобто "вага" зв'язку для всіх пар $\langle x_i, x_2\rangle$ є однаковою і дорівнює 1.

Можливою фізичною інтерпретацією наведеної математичної моделі буде така задача: *перервати зв'язок між споживачами інформації у інформаційній мережі противника шляхом виведення з ладу мінімальної кількості елементарних ланок зв'язку за умови, що відомі початкові і кінцеві пункти зв'язку.*

Цю задачу можна представити як задачу про максимальний потік на неорієнтованому графі, вершинами якого є пункти проміжного зв'язку противника, а ребрами – канали передачі інформації (рис. 1).

На цьому рисунку: x_1 (перший пункт) відповідає вершині 1, а x_k (кінцевий пункт) – вершині 8.

Необхідно знайти мінімальне число ребер, розсічення яких перериває всі шляхи з вершини 1 у вершину 8.

Перш ніж приступити до розв'язання задачі, виберемо найбільш зручне подання графа.

1. На рис. 1 показано так звану діаграму графа [9]. Це перше і найбільш поширене подання графа.

2. Можливе подання графа перерахуванням його вершин (множина X) і ребер (перерахуванням пар вершин, між якими є ребро – відношенням $\Gamma \subseteq X \times X$. Тут $X \times X$ – декартовий добуток X самого на себе).

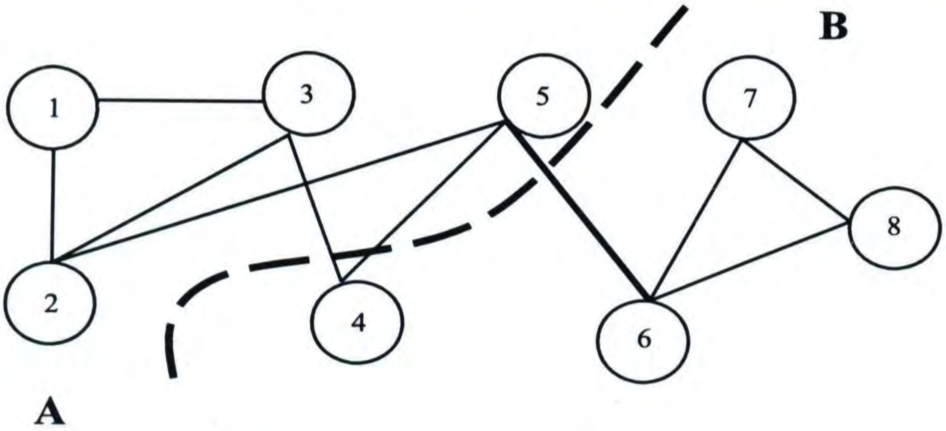

Рис. 1. Геометрична інтерпретація графа

Таке подання для нашого прикладу має вигляд:

X={1, 2, 3, 4, 5, 6, 7, 8};

Γ = {<1, 2>, <1, 3>, <2, 1>, <2, 3>, <2, 5>, <3, 1>, <3, 2>, <3, 4>, <4, 3>, <4, 5>, <5, 2>, <5, 4>, <5, 6>, <6, 5>, <6, 7>, <6, 8>, <7, 6>, <7, 8>, <8, 6>, <8, 7>}.

3. Можливе завдання графа матрицею суміжності, у якій на перетині рядка i й стовпця j знаходиться одиниця, якщо існує ребро <i, j>. Для неорієнтованого графа матриця суміжності є симетричною.

Нижче наведено матрицю суміжності для нашого прикладу:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	1	1	0	0	0	0	0
2	1	0	1	0	1	0	0	0
3	1	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	0	1	0	1	0	1	0	0
6	0	0	0	0	1	0	1	1
7	0	0	0	0	0	1	0	1
8	0	0	0	0	0	1	1	0

4. Найбільш привабливим для аналітичних маніпуляцій вважають завдання графа сукупністю тривимірних (або трикомпонентних) кортежів [9].

У цьому випадку наш приклад матиме вигляд:

<1, 2, {2, 3}>;
<2, 3, {1, 3, 5}>;
<3, 3, {1, 2, 4}>;
<4, 2, {3, 5}>;
<5, 3, {2, 4, 6}>;
<6, 3, {5, 7, 8}>;
<7, 2, {6, 8}>;
<8, 2, {6, 7}>.

У першій позиції тривимірного кортежу K_i, що описує вершину i, знаходиться ідентифікатор вершини i. Позначивши через Pr_j (j =1, 2, 3) відповідну проекцію кортежу (місце в кортежі), ми можемо попереднє речення записати формулою $Pr_1K_i = i$.

У третій позиції кортежу K_i знаходиться множина вершин, суміжних даній (у які входять ребра, що виходять з вершини i), тобто $Pr_3K_i = \Gamma(i)$ – згадаємо другий спосіб опису графа.

У другій позиції представлено число вершин, суміжних даній, тобто $Pr_2K_i = |Pr_3K_i|$.

Для розв'язання задачі скористаємося алгоритмом локального поліпшення розрізу, описаним у [1].

Вихідні дані:

$$X = \{1, 2, 3, 4, 5, 6, 7, 8\};\ x_1 = 1;\ x_k = 8.$$

Задамо довільно початкову розбивку X на дві непересічних підмножини A і B. Розбивка повинна задовольняти обмеженням:

$$A \cup B = X;\ A \cap B = \varnothing;\ x_1 \in A,\ x_k \in B.$$

Нехай це буде

$$A = \{1, 2, 3, 5\};\ B = \{4, 6, 7, 8\}.$$

Розв'язок.

Нагадаємо, що відповідно до алгоритму, описаному в [1], розв'язок задачі полягає в знаходженні серед вершин, що визначають розріз, тих, у яких число зв'язків з "чужою" множиною перевищує число "своїх" зв'язків. Такі вершини змінюють свою приналежність, тобто переносяться з однієї множини в іншу, формуючи тим самим новий розріз. Процедура повторюється, поки не буде знайдено стійке положення.

Нагадаємо, що для вершини 1 "своєю" множиною завжди є A, а для вершини 8 "своєю" є B.

Якщо $i \in A$, то умовою перенесення вершини i у множину B є виконання для неї нерівності

$$|Pr_3K_i \cap B| > |Pr_3K_i \cap A|. \quad (1)$$

І, навпаки, якщо $i \in B$, то умовою перенесення вершини i у множину A є виконання для неї нерівності

$$|Pr_3K_i \cap A| > |Pr_3K_i \cap B|. \quad (2)$$

Розглянемо опис графа трикомпонентними кортежами, супроводжуючи його коментарями

$\langle 1, 2, \{2, 3\}\rangle$ – вершина не підлягає перенесенню;
$\langle 2, 3, \{1, 3, 5\}\rangle$ – усі вершини з третьої проекції "свої";
$\langle 3, 3, \{1, 2, 4\}\rangle$ – "своїх" (1, 2) більше, ніж "чужих" (4);
$\langle 4, 2, \{3, 5\}\rangle$ – усі суміжні вершини є "чужими" – **вершину 4 потрібно перенести в множину *B*;**
$\langle 5, 3, \{2, 4, 6\}\rangle$ – "чужих" вершин більше, ніж "своїх" (вершина 5 можливий претендент на перенос);
$\langle 6, 3, \{5, 7, 8\}\rangle$ – "своїх" вершин (7, 8) більше, ніж "чужих" (5);
$\langle 7, 2, \{6, 8\}\rangle$ – усі вершини з третьої проекції є "своїми";
$\langle 8, 2, \{6, 7\}\rangle$ – вершина не підлягає перенесенню.

Переносимо вершину 4 у множину B (рис. 2) і переконуємося, що задача розв'язана.

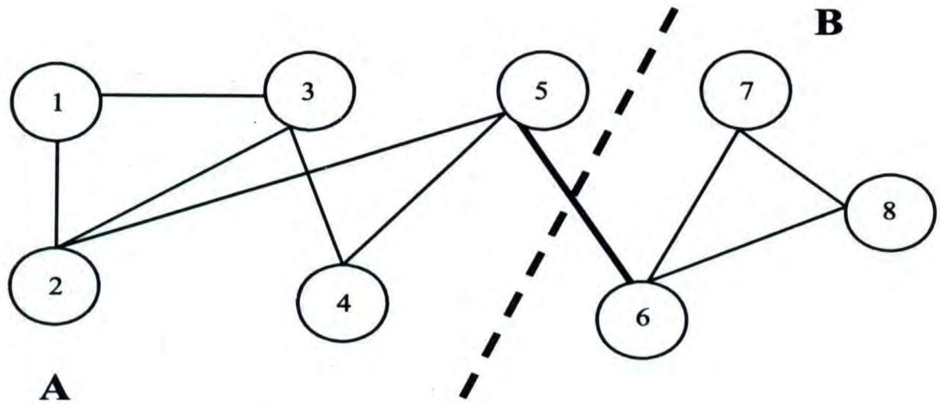

Рис. 2. Геометрична інтерпретація графа після переносу вершини 4

Висновки

1. У загальній постановці задача про розподіл взаємопов'язаних ресурсів є повною задачею нелінійного програмування.
2. Частковим випадком задачі розподілу взаємопов'язаних ресурсів є задача про максимальний потік, яка успішно вирішується алгоритмом, наведеним у роботі [1].

Список використаних джерел

1. Сбітнєв А. І. Алгоритми розподілу взаємопов'язаних інформаційних ресурсів / А. І. Сбітнєв, В. В. Козлов // Науковий вісник Національної академії статистики, обліку та аудиту. – К. : НАСОА, 2013. – Вип. 4 (39). – С. 104–108.
2. Марчук Г. И. Методы вычислительной математики : [учеб. пособие] / Г. И. Марчук. – М. : Наука. 1989. – 808 с.
3. Березин И. С. Методы вычислений : [учеб. пособие] / И. С. Березин, Н. П. Жидков. – М. : Наука, 1966. – Т. 1. – 632 с.
4. Гольштейн Е. Г. Задачи линейного программирования транспортного типа / Е. Г. Гольштейн, Д. Б. Юдин. – М. : Наука, 1969. – 392 с.
5. Зуховицкий С. И. Линейное и выпуклое программирование : [справочное руководство] / С. И. Зуховицкий, Л. И. Авдеева. – М. : Наука, 1964. – 318 с.
6. Хедли Дж. Нелинейное и динамическое программирование / Дж. Хедли. – М. : Мир, 1967. – 506 с.
7. Солодов А. В. Теория информации и ее применение к задачам автоматического управления и контроля / А. В. Солодов. – М. : Наука, 1967. – 432 с.
8. Стар М. Управление производством / М. Стар. – М. : Прогресс, 1968. – 398 с.
9. Сбітнєв А. І. Систематизація та загальний аналіз сучасного стану прикладних задач розподілу зв'язаних ресурсів і методів їх розв'язку / А. І. Сбітнєв, В. В. Козлов // Прикладна статистика: проблеми теорії та практики : зб. наук. праць. – К. : НАСОА, 2011. – Вип. 9. – С. 96–103.

А. И. СБИТНЕВ,
доктор технических наук, профессор,
действительный член Международной академии компьютерных наук и систем,
заслуженный деятель науки и техники Украины,
профессор Национального университета обороны Украины
В. В. КОЗЛОВ,
кандидат технических наук, доцент,
заведующий кафедрой информационных систем и технологий,
Национальная академия статистики, учета и аудита

Формулировка задачи распределения взаимосвязанных ресурсов как задачи математического программирования

Показано, что в общей постановке задача о распределении взаимосвязанных ресурсов является полной задачей нелинейного программирования, а частным случаем задачи распределения взаимосвязанных ресурсов является задача о максимальном потоке, которая успешно решается приведенным в работе алгоритмом.

Ключевые слова: *алгоритм, связанные ресурсы, распределение ресурсов, локальный экстремум.*

A. I. SBITNEV,
Dr. Sc. (Engineering), Prof.,
Full Member of the International Academy of Computer Sciences and Systems,
Honored Worker of Science and Technology of Ukraine,
Professor of National Defense University of Ukraine
V. V. KOZLOV,
PhD (Engineering), Associate Professor,
Head of Information Systems and Technologies Department,
National Academy of Statistics, Accounting and Audit

Setting the Problem of Allocating Connected Resources as a Problem of Mathematical Programming

The problem of allocating connected resources was set in previous works, with basic elements for constructing its algorithm. The authors propose to fit this problem into the conventional model of mathematical programming, translating it into a problem of nonlinear programming with booleans.

Using the exhaustive search, the number of variants equals the binary number of y_1, y_2, ..., y_{n-1}, y_n, that is, 2^n. The NP problem (full nonlinear programming problem), therefore, occurs. This problem belongs to the class of problems of graph theory, known as the maximal flow problem.

The conventional algorithm of searching for maximal flow is based on the proof of "theorem about maximal flow and minimum cut", formulated as follows: the maximal flow from the source s to the source t equals the minimum size of cuts which separate s from t.

Physical interpretation of this mathematical model is given. The algorithm of local improvement of cut is used for the solution of the problem.

In accordance with the algorithm, the problem is solved by finding the tops from among the tops determining a cut, in which the number of connections with the "strange" set exceeds the number of "own" connections. Such tops change their belonging, by moving from one set to another one, forming a new cut in this way. The procedure recurs unless a stable position is found.

It is shown that basically the problem on allocating connected resources is a full nonlinear programming problem, whereas the problem of maximal flow is a special case of the problem of connected resources, which can be effectively solved by the algorithm set up in the article.

Keywords: *algorithm, connected resources, distribution of resources, local extreme.*