УДК 681.5

О.Г. Оксіюк, В.І. Вялкова

ПВНЗ «Європейський університет», Київ

РОЗПОДІЛ ФУНКЦІЙ МІЖ ЛОКАЛЬНИМИ СИСТЕМАМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

В статті розроблено евристичний алгоритм для рішення завдання розподілу функцій між локальними системами підтримки прийняття рішень, що базується на наближеному методі розрізування графів. Розподіл функцій між елементами ієрархічної системи є досить типовим завданням проектування складних технічних систем. З метою мінімізації часових витрат, пов'язаних з обміном інформацією між рівнями системи, і вартості апаратури звичайно прагнуть сконцентрувати на кожному рівні функції, які мають максимальний взаємозв'язок у процесі функціонування системи.

Ключові слова: *системи підтримки прийняття рішень, евристичний алгоритм, метод розрізування графів.*

Вступ

Постановка проблеми. Розподіл функцій між елементами ієрархічної системи є досить типовим завданням проектування складних технічних систем. З метою мінімізації часових витрат, пов'язаних з обміном інформацією між рівнями системи, і вартості апаратури звичайно прагнуть сконцентрувати на кожному рівні функції, які мають максимальний взаємозв'язок у процесі функціонування системи. Найпоширенішими методами рішення подібних завдань є методи, що базуються на розрізуванні графів [1].

Постановка завдання може варіюватися залежно від властивостей графів, які визначаються структурою й функціями системи, яка проектується.

Аналіз досліджень. У роботі [2] вирішується завдання розрізування простого лінійного неорієнтованого графа, ребра якого мають однакову вагу, що дорівнює одиниці; в [3] розглядається завдання про розрізування неорієнтованого графа у зваженими ребрами й вагами вершин, які рівні одиниці; в [4] описується алгоритм розрізування без зв'язкового орографу зі зваженими дугами, тощо.

Актуальність постановки завдання в різних областях техніки викликала велику кількість спроб її рішення. По точності рішення всі методи поділяються на точні й наближені. До точного ставиться метод повного перебору [1], а з методів упорядкованого перебору - метод "галузей і границь" [4]. Інші методи є, як правило, наближеними, евристичними.

Застосування того або іншого методу багато в чому залежить від числа вершин графа й насиченості матриці суміжності графа. Розмірність графа умовно визначається як мала частина (число вершин $n \le 6$), середня ($6 < n \le 30$) і більша ($n > 30$) [1].

Алгоритм повного перебору доцільно використати для графів з малою розмірністю. Метод "галузей і границь" застосуємо для графів із середньою розмірністю. Для графів з більше високою розмірністю ці методи вимагають занадто великих часових витрат. Тому на практиці для розрізування графів середньої й тим більше великої розмірності часто застосовуються наближені евристичні алгоритми.

Постановка завдання. Розробка алгоритму для рішення завдання розподілу функцій між локальними системами підтримки прийняття рішень.

Основна частина

Сформулюємо завдання розподілу функцій. Нехай задана множина функцій F, які підлягають розподілу між 1 елементами: $F = \{F_{i,j} = 1, I\}$; $F_i \cap F_j = \varnothing$; $i, j = \overline{1, j}$; $i \neq j$. Кожну з функцій $F_i \in F \times F^*$ можна реалізувати на одному з 1 елементів; F^* – множина уже розподілених функцій.

Введемо параметр розподілу X_{ik}:

$$X_{ik} = \begin{cases} 1, & \text{якщо i-я функція реализується на k-м елементі, } k = \overline{1,l}, \\ 0 - & \text{в протилежному випадку.} \end{cases}$$

Обов'язкове виконання умови $\sum_{i=1}^{I} X_{ik} = 1$, $k = \overline{1,l}$, $i = \overline{1,I}$, тобто кожну функцію можна закріпити тільки за одним елементом.

Далі через a_{ij} позначимо алгоритмічну зв'язність i-ї функції з j-ю (відносна частота виконання функції F_j після функції F_i), B_i – обсяг пам'яті, необхідний для виконання i-ї функції, $B^{(k)}$ – розташовуваний обсяг пам'яті обчислювальних засобів k-го елемента.

Введені позначення завдання розподілу функцій можна представити у вигляді наступного завдання цілочисельного лінійного програмування: знайти

$$\min \sum_{k=1}^{l} \sum_{i=1}^{I} a_{ij} X_{ik} \qquad (1)$$

при обмеженнях

$$\begin{gathered} \sum_{i=1}^{I} B_i X_{ik} \le B^{(k)}, \ k = \overline{1,l}; \\ \sum_{i=1}^{I} X_{ik} = 1, \ X_{ik} = 0,1. \end{gathered} \qquad (2)$$

Рішенням завдання буде сукупність векторів $X_1^* = |X_{11}, ..., X_{i1}, ..., X_{I1}|$; $X_2^* = |X_{12}, ..., X_{i2}, ..., X_{I2}|$; $X_l^* = |X_{1l}, ..., X_{il}, ..., X_{Il}|$ зі складовими $X_{ik} = 0,1$, що забезпечують мінімум цільової функції.

Завдання можна інтерпретувати як завдання розрізування кінцевого орієнтованого зваженого графа $G = (Y, V)$, у якому вершинам множини Y ставиться у відповідність обсяг пам'яті, яка займана i-ю функцією B_i, а множини дуг – алгоритмічна зв'язаність i-ї функції з j-ю – a_{ij}. Рішення складається в розрізуванні графа G на k = l підграфів <G_k>, що задовольняють умовам $Y = \bigcup^{l} Y_k$; $Y_1 \cap Y_2 \cap Y_l = \varnothing$ і вимогам мінімуму цільової функції (1) при обмеженнях на інші параметри (2).

Розглянемо можливість застосування алгоритму, викладеного в роботі [5] для рішення поставленого завдання. Алгоритм послідовно закріплює вершини графа $G = (Y, V)$ за заданим числом підграфів $G_k = (Y_k, V_k)$, $k = 1 \div l$. У процесі роботи алгоритму всі вершини діляться на вже закріплені $\left(F_n^*\right)$ і ще не закріплені $\left(F_l = F / F_n^*\right)$, тобто вільні.

Нижня оцінка ступеня зв'язаності графа G при перенесенні деякої i-ї вершини з множини F_l в множину F_n^* визначається в такий спосіб:

$$Q(Y_{n+1}) = Q(Y_n) + \Delta Q_{ij}(Y_n) + q_{ij} + d_{ij}, \qquad (3)$$

де $Q(Y_n)$ – зв'язність k підграфів, частково сформованих закріпленням вершин (сума ваг ребер, що зв'язують закріплені вершини на різних частково сформованих підграфах); $\Delta Q_{ij}(Y_n)$ – збільшення $Q(Y_n)$ при включенні деякої i-ї вільної вершини $(Y_i \in F_l)$ у деякий підграф G_j; q_{ij} – нижня оцінка суми ваг ребер, що з'єднують вершини з множиною вже закріплених вершин F_n^* з вершинами з множиною ще вільних F_l, які утворять граф Γ_i^n; d_{ij} – нижня оцінка зв'язаності розрізування графа Γ_i^n, що складений з вільних вершин.

Нижня оцінка q_{ij} дорівнює оптимальному значенню цільової функції

$$q_{ij} = \min \sum_{l \in F_l} \sum_{k=1}^{l} a_{lk} X_{lk} \qquad (4)$$

при обмеженнях

$$\begin{gathered} \sum_{l \in F_l} X_{lk} = 1, \ k = \overline{1,l}; \\ X_{ik} \ge 0, \ i \in F_l; \\ \sum_{l \in F_l} B_l X_{lk} \le B^{(k)}, \end{gathered} \qquad (5)$$

де a_{lk} – сума ваг ребер, що з'єднують вершину $l \in F_l$ з усіма вершинами з F_n^*, крім вершин підграфа *G_k*;

$$X_{lk} = \begin{cases} 1, & \text{якщо l-а вершина закріплена за } G_k, \\ 0 - & \text{в протилежому випадку.} \end{cases}$$

Нехай на m-му кроці алгоритму n вершин, що утворюють F_n^*, закріплюються за підграфами G_k, $k = 1 \div l$. У графі $\Gamma_i^n(V^n, U^n)$, отриманому з вихідного графа G шляхом видалення всіх уже закріплених вершин, знаходиться вершина з найбільшою кількістю ребер, що виходять із неї. Виділяються групи підграфів G_k з найменшою сумою ваг вершин; із всіх груп вибирається по одному підграфу. Далі до кожного залишеного підграфу приєднується обрана вершина, при цьому по формулі (3) визначається нижня оцінка зв'язності.

Вершина закріплюється за тим підграфом, що буде мати найменше значення $Q(Y_{n+1})$, що обумовлено формулою 3. Процедура аналогічно виконується для інших вершин.

Точне рішення завдання можна одержати за конкретним числом кроків алгоритму, що реалізує процес спрямованого перебору з поверненнями й нагадує метод "галузей і границь". До того ж для обчислення нижньої оцінки суми ваг дуг потрібне рішення транспортного завдання.

При збільшенні розмірності графа підвищується обчислювальна складність алгоритму й застосування його для розподілу функцій стає неефективним.

У цьому випадку найбільш прийнятними є наближені евристичні алгоритми.

Відомі два евристичних алгоритми [6, 7], які принципово можуть бути застосовні для рішення завдання розрізування графа в постановці (1), (2): алгоритм розбивки зваженого графа на зв'язкові підграфи із заданим числом вершин і мінімальною сумарною вагою розрізаних ребер; алгоритм розбивки міченої мережі на задане число зв'язаних підграфів з обмеженнями на суми ваг вершин і мінімальною сумарною вагою розрізаних ребер.

У першому випадку заданий неорієнтований кінцевий зв'язний зважений граф (мережа) G_k = =<A_k, P_k>, де $A = \{a_{ij}\}$ – множина вершин, $|A| = n$; P – множина ребер; a_{ij} – вага ребра $[a_i, a_j]$; $a_{ij} = a_{ji}$.

Потрібно розрізати граф G на T зв'язаних підграфів G_k =<A_k, P_k>, k = 1, 2,..., T із заданим числом вершин $|A_k| = m_k$, $\sum_{k=1}^{T} m_k = n$ так, щоб була мінімальної величина Z – сума ваг розрізаних ребер.

Алгоритм будується, виходячи з матричного способу завдання зваженого графа, в такий спосіб.

Формується симетрична матриця суміжності вихідного зваженого графа *G*

$$W = (w_{ij}), \quad i, j = \overline{1, n};$$

де $w_{ij} = \begin{cases} a_{ij}, & \text{якщо } |a_i, a_j| \in P, \\ 0 - & \text{в протилежному випадку} \end{cases}$.

Основна ідея виділення вершин для шуканого підграфу укладається у визначенні базової вершини, що має найбільший локальний ступінь зв'язку й "притяганні" до неї вершин, найбільш зв'язаних із цією базовою вершиною. Недоліком цього алгоритму є те, що для визначення обсягу отриманих при розрізуванні підграфів використається кількість вершин, а не їхні ваги.

Даний недолік усувається іншим алгоритмом, що вирішує завдання розрізування графа в наступній постановці [7]. Є мічена мережа G =<A, P>, тобто неорієнтований кінцевий зв'язний граф, у якому вершини мають ваги; $A = \{a_i\}$ – множина вершин, $|A| = n$; P – множина ребер; a_{ij} – вага ребра $[a_i, a_j]$; $a_{ij} = a_{ji}$; β_i – вага вершини a_i. Потрібно розрізати граф G на T зв'язкових підграфів G_k =<A_k, P_k>, k = 1, 2,..., T, таких, щоб сума ваг вершин цих підграфів $B_k = \sum_{a_i \in A_k} \beta_i$ задовольняла оцінці $\underline{B} \le B_k \le \overline{B}$. При цьому повинна бути мінімальної величина Z – сума ваг розрізаних ребер у розбивці графа. Цей алгоритм, як і попередній, будується, виходячи з матричного способу завдання зваженого графа, і використовує ідею виділення вершин для шуканого підграфу. Відмінність укладається в тім, що замість кількості вершин застосовуються ваги вершин, на суму яких накладаються обмеження.

Розглянемо евристичний алгоритм, що дозволяє усунути зазначений недолік [6]. Нехай даний орієнтований кінцевий зважений граф G =<A, P>, що відповідає функціям, які виконуються обчислювальними засобами, де $A = \{a_i\}$ – множина вершин, $|A| = n$; P – множина ребер; β_i – вага вершини a_i; a_{ij} – вага ребра $[a_i, a_j]$, $a_{ij} = a_{ji}$; A_k^* – множина вже закріплених вершин за підграфами <G_k>. Необхідно розрізати граф G на T зв'язаних підграфів G_k =<A_k, P_k>, k = 1, 2,..., T, таких, щоб сума ваг розрізаних ребер Z була мінімальною, а сума ваг вершин підграфів $B_k = \sum_{a_i \in A_k} \beta_i$ задовольняла оцінці $B^k \le B_k$, k = 1, 2,..., T.

Алгоритм працює в такий спосіб.

1. Вводиться матриця суміжності вершин графа W_1 (операція 1).

2. Підсумовуються елементи матриці W_1 в рядках (у стовпцях), визначаються локальні ступені зв'язку вершин графа G_1. Шляхом приписування відповідних рядків і стовпців формується розширена матриця $W_1^{(1)}$ (операція 2).

3. У рядку a_i матриці $W_1^{(1)}$, що відповідає вершині a_i з найбільшим локальним ступенем зв'язку, визначається найбільший елемент a_{ij} (операції 3, 4). Якщо в рядку a_i є декілька найбільших елементів, то вибирається вершина стовпця з найбільшою сумою:

1) Якщо вершина a_i або вже закріплена за одним з підграфів, то переходимо до операції 4.

2) Якщо вершини a_i й a_j попередньо не закріплені, то в рядку a_i вибираємо наступний найбільший елемент a_{ik}, і якщо вершина a_k закріплена за одним з підграфів G_k, те переходимо до операції 4; у противному випадку пошук триває.

3) Якщо вершина не зв'язана з жодною вже закріпленою вершиною, то вибирається наступний рядок a_k з найбільшим локальним ступенем зв'язку й пошук триває.

4. Якщо виконується умова $\Delta B_k < \beta_i + \beta_j \le B_k$; $\Delta B_k = B_k - \min \beta_i$, то вершина a_i закріплюється за множиною $A_k^{(1)}$ (операція 9). Формування множини A_k закінчується. Підграф G_k сформований.

1) Якщо виконується умова $\beta_i + \beta_j \le B_k$, то вершина a_j (a_i) закріплюється за множиною $A_k^{(1)}$, до якої належить раніше закріплена вершина (операції 11, 12). Записується матриця суміжності $W_1^{(2)}$ графа, отриманого з вихідного графа шляхом конденсації вершин a_i і a_j. Матриця $W_1^{(2)}$ виходить із матриці $W_1^{(1)}$ шляхом викреслювання a_i рядків (стовпців) і приписування рядка (стовпця) в отриману матрицю $W_1^{(1)}$ по елементним додаванням a_i-ї й a_j-й рядків (стовпців).

2) Якщо виконується умова $\beta_i + \beta_j > B_k$, то a_i не включається в множину $A_k^{(1)}$. Множина A_k закінчується (закривається). Підграф G_k є сформований.

5. У сумарному рядку $a_i(j)$ визначаємо найбільший елемент.

1) Якщо найбільший елемент рядка $a_i(j)$, що стоїть в стовпці a_k, більше (або дорівнює) суми інших елементів стовпця a_k й виконується умова $\beta_{i(j)} + \beta_k \le B_k$, то множина $A_k^{(1)}$ поповнюється вершиною a_k (якщо вона не закріплена за іншим підграфом)., В четвертому пункті алгоритму замість a_i фігурує $a_i(j)$; а замість $a_j - a_k$. Процес конденсації триває до максимально припустимої суми ваг сконденсованих вершин. При останній конденсації потрібно віддати перевагу тій вершині a_k, для якої різниця між відповідним елементом сконцентрованого рядка й сумою інших елементів стовпця a_k найбільша. Знаходимо множину вершин A_k підграфа G_k.

2) Якщо найбільший елемент рядка $a_i(j)$, що стоїть у стовпці a_k, менше суми інших елементів стовпця a_k, то множину вершин зв'язаного підграфу G_k знайдено $\left(A_k = A_k^{(1)}\right)$.

6. Видаляємо із графа G вершини підмножини A_k з усіма інцидентними їм ребрами, тобто викреслюємо з матриці W_1 рядки й стовпці, що відповідають вершинам підмножини A_k (операція 14). Одержуємо матрицю суміжності W_2. Досліджуємо підграф, що залишився, на зв'язаність. Якщо він зв'язаний, то переходимо до операції 2.

Після знаходження (k – 1) підмножин A_1, $A_2, .., A_{k-1}$ та видалення вершин цих підмножин з множини вершин A, перевіряємо щоб сума ваг вершин, що залишилися, задовольняла умові $\sum_{i=1}^{A_k} \beta_i \le B_k$. Якщо умова виконується, то множина вершин, які залишились, утворюють останню шукану підмножину A_k; у противному випадку одне з підмножин A_1, $A_2, .., A_{k-1}$ варто поповнити за рахунок збільшення сумарної ваги розрізаних ребер.

Висновок

Розроблено евристичний алгоритм для рішення завдання розподілу функцій між локальними системами підтримки прийняття рішень, що базується на наближеному методі розрізування графів.

Список літератури

1. *Корбут А.А. Дискретное программирование / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.*
2. *Горинштейн Л.Л. О разрезании графов / Л.Л. Горинштейн // Изв. АН СССР. Техн.. кибернетика. – 1969. – № 1. – С. 79-85.*
3. *Бернштейн А.С. О линейном разрезании графов со взвешенными ребрами / А.С. Бернштейн, В.В. Семенкин // Электр. техника. – 1976. – Сер. 9, вып. 4(20). – С. 96-106.*
4. *Бурков В.Н. Решение задачи о линейном разрезе в бисвязном орографе алгоритмами типа «ветвей и границ» / В.Н. Бурков, В.О. Гроппен // Автоматика и телемеханика. – 1974. – № 9. – С. 104-110.*
5. *Кульба В.В. О формализованном распределении множества решаемых задач между различными узлами системы управления / В.В. Кульба, А.Д. Цвиркун // Автоматика и телемеханика. –1970. – № 9. – С. 28-35.*
6. *Фещенко В.П. Интерационный алгоритм разрезания графа на N подграфов / В.П. Фещенко, Л.П. Матюшков // Автом. проектирование сложных систем. – 1976. – Вып. 2. – С. 74-77.*
7. *Рыжков Д.П. Алгоритм решения графа на минимально-связанные подграфы / Д.П. Рыжков // Изв. АН СССР. Техн. кибернетика. – 1975. – № 6. – С. 122-128.*

Надійшла до редколегії 14.08.2012

Рецензент: д-р техн. наук, проф. О.М. Фоменко, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.

РАСПРЕДЕЛЕНИЕ ФУНКЦИЙ МЕЖДУ ЛОКАЛЬНЫМИ СИСТЕМАМИ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

О.Г. Оксиюк, В.И. Вялкова

В статье разработан эвристический алгоритм для решения задания распределению функций между локальными системами поддержки принятия решений, которое базируется на приближенном методе разрезания графов. Распределение функций между элементами иерархической системы является достаточно типичным заданием проектирования сложных технических систем. С целью минимизации часовых расходов, связанных с обменом информацией между уровнями системы и стоимости аппаратуры обычно стремятся сконцентрировать на каждом уровне функции, которые имеют максимальную взаимосвязь в процессе функционирования системы.

Ключевые слова: *системы поддержки принятия решений, эвристический алгоритм, метод разрезания графов.*

DISTRIBUTING OF FUNCTIONS BETWEEN IN-PLANT SYSTEMS OF SUPPORT OF MAKING A DECISION

O.G. Oksiyuk, V.I. Vyalkova

In the article a heuristic algorithm is developed for the decision of task distributing of functions between the in-plant systems of support of making a decision, which is based on the close method of scission of counts. Distributing of functions between the elements of the hierarchical system is the typical enough task of planning of the difficult technical systems. With the purpose of minimization of sentinel charges, related to the exchange information between the levels of the system and cost of apparatus usually aim to concentrate at every level functions which have maximal intercommunication in the process of functioning of the system.

Keywords: *systems of support of making a decision, heuristic algorithm, method of scission of counts.*