МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАЧИЯ

Р С Ф С Р

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени академика С.П.Королева

АВТОМАТИЗАЦИЯ ТИПОВЫХ ЗАДАЧ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ РЭА. ИССЛЕДОВАНИЕ АЛГОРИТМОВ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ РЭА

(Венгерский метод)

Лабораторная работа

Куйбышев 1986

УДК 621.396.6

В настоящих методических указаниях даются теоретические сведения и практические рекомендации, необходимые для успешного выполнения лабораторной работы "Автоматизация типовых задач конструкторского проектирования РЭА. Исследование алгоритмов размещения элементов РЭА. (Венгерский метод)".

Методические указания могут найти применение при выполнении курсовых проектов и на практических занятиях по курсу "Математическое обеспечение автоматизации конструирования и технологии с применением САПР" для специальности 0705.

I. Общие понятия транспортной задачи

Среди задач математического программирования особое место занимают задачи, математическая модель которых может быть записана с помощью системы линейных соотношений вида:

$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b_0$$

Типичным практическим примером подобных задач служит транспортная задача. Она в общем виде формируется так.

Пусть имеется m пунктов отправления $A_1, A_2, \ldots, A_m$ и n пунктов назначения $B_1, B_2, \ldots, B_n$. Ресурсы груза в i-м пункте отправления обозначим через a_i $(i = 1,2,\ldots,m)$, а потребность каждого j-го пункта потребления – через b_j ($j = 1,2,\ldots,n$). Заданы стоимости перевозки единицы груза от каждого i-го пункта отправления до каждого j-го пункта назначения. Требуется определить, какое количество груза $x_{ij} \geqslant 0$ необходимо перевезти из каждого i-го пункта отправления к каждому j-му пункту назначения, чтобы:

1) вывезти грузы всех поставщиков;

2) удовлетворить всех потребителей;

3) достигнуть экстремума целевой функции.

С математической точки зрения транспортная задача линейного программирования формулируется следующим образом: отыскать набор переменных $\{x_{ij}\}$, минимизирующий линейную целевую функцию

$$L(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \qquad (1)$$

и удовлетворяющий системе линейных ограничений

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

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,\ldots,n; \qquad (3)$$

$$x_{ij} \geq 0, \quad i = 1,2,\ldots,m; \; j = 1,2,\ldots,n. \qquad (4)$$

В силу линейности целевой функции (I) и ограничений (2) и (3), для решения транспортной задачи может быть в принципе использован симплекс-метод. Реализацией этого метода, учитывающей конкретные особенности структуры транспортной задачи, является так называемый метод последовательного улучшения плана. Метод позволяет проверить любой опорный план на оптимальность и в случае необходимости обеспечить переход к новому опорному плану, на котором целевая функция имеет лучшее значение, чем на предыдущем. Искомый опорный план для удобства записывается в виде таблицы (матрицы).

П. Проблема выбора. Венгерский метод

Важным частным случаем транспортной задачи линейного программирования является проблема выбора (или, как часто ее называют, проблема назначения). Особенности этой задачи следующие:

- решение должно быть целочисленным;
- каждый из элементов множества чисел $\{a_i\}$ и $\{b_j\}$ в ограничениях (2) и (3) равен единице;
- количество строк и столбцов матрицы равны, т.е. $m = n$.

Проблема выбора возникает при решении следующей типовой задачи конструирования РЭА.

Пусть необходимо разместить n функциональных модулей $A_1, A_2, \ldots A_i, \ldots A_n$ на n монтажных местах $B_1, B_2, \ldots B_j, \ldots, B_n$. Введем параметр c_{ij}, характеризующий эффективность размещения модуля A_i на месте B_j. Набор параметров c_{ij}, $i = 1,2,\ldots,n$; $j = 1,2,\ldots,n$, сведенный в квадратную матрицу

$$C = \begin{vmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \hdashline c_{n1} & c_{n2} & \ldots & c_{nn} \end{vmatrix},$$

характеризует степень целесообразности назначения различных функциональных модулей на монтажные места. Задача сводится к отысканию набора монтажных мест, обеспечивающих максимальный суммарный эффект.

Это типичная экстремальная задача комбинаторного вида. Ее решение путем прямого перебора практически невозможно при сколько-нибудь больших n, так как число перестановок $N = n! = 1 \times 2 \times 3 \times \ldots \times (n-1) \times n$. Поэтому наибольшее распространение для решения данной задачи нашел венгерский метод. Основной принцип его - оптимальность решения задачи о назначении - не нарушается при уменьшении (увеличении) элементов строки (столбца) на одну и туже величину d_i (d_j). Решение считается оптимальным, если все измененные искуственно элементы $c_{ij} \geq 0$ ($i, j = 1, 2, \ldots, n$) и можно отыскать такой набор x_{ij}, что $L(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c'_{ij} \cdot x_{ij} = 0$. Таким образом, общая идея алгоритма венгерского метода состоит в том, чтобы, сохраняя неотрицательность элементов матрицы, путем использования эквивалентных преобразований получить в результате матрицу C^*, содержащую n нулевых элементов, никакие два из которых не лежат в одной и той же строке и в одном и том же столбце матрицы. (Систему нулевых элементов матрицы, обладающую указанным свойством, будем называть системой независимых нулей). Тогда, если система независимых нулей содержит ровно n элементов, а остальные элементы матрицы C неотрицательны, то решение задачи $L_{C^*}(X)$ при удовлетворении (2) и (3) состоит в выборе плана назначений X^*, единичные компоненты которого соответствуют нулевым элементам матрицы C^*.

Универсальность венгерского метода применительно к задачам о назначениях заключается в том, что если при заданных ограничениях линейной модели можно найти максимальное значение целевой функции, которому соответствует некоторый оптимальный план $C^* = \{C'_{ij}\}$, то этот же план является одновременно и решением задачи минимизации целевой функции

$$L'(C^*) = \sum_{i=1}^{n}\sum_{j=1}^{n} C'_{ij} \cdot x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} C^{*\prime}_{ij}, \quad (5)$$

где $C'_{ij} = -C_{ij}$ при тех же линейных ограничениях. Это утверждение вытекает из принципа двойственности задач линейного программирования. Таким образом, венгерский метод можно применять как для поиска максимума, так и минимума целевой функции.

Ш. Алгоритм венгерского метода

Алгоритм венгерского метода состоит из предварительного этапа и не более чем ($n-2$) последовательно повторяющихся итераций. На подготовительном этапе образуется начальная система независимых нулей. В дальнейшем на каждой итерации число независимых нулей увеличивается не менее чем на единицу. Поэтому через конечное число шагов число независимых нулей достигает n, что означает окончание решения: оптимальный план однозначно определяется позициями независимых нулей в матрице, полученной на последней итерации. Рассмотрим детально работу основных этапов алгоритма венгерского метода, иллюстрируя конкретным примером.

Пример. Решить проблему выбора, определяемую матрицей C :

$$C = \begin{vmatrix} 3 & 7 & 5 & 8 \\ 2 & 4 & 4 & 5 \\ 4 & 7 & 2 & 8 \\ 9 & 7 & 3 & 8 \end{vmatrix}$$

Решение: А. Подготовительный этап. Образование первоначальной системы независимых нулей.

А-I. Получение нулей в каждой строке. Найти наименьший элемент в каждой строке матрицы C и вычесть его из всех ее элементов

$$d_i = \min_j \{c_{ij}\} \qquad (6)$$

$$c'_{ij} = c_{ij} - \min_j \{c_{ij}\} \geqslant 0$$

А-2. Получение нулей в каждом столбце. Найти наименьший элемент в каждом столбце матрицы C' и вычесть его из всех элементов столбца

$$d_j = \min_i \{c'_{ij}\} \qquad (7)$$

$$c''_{ij} = c'_{ij} - \min_i \{c'_{ij}\} \geqslant 0$$

$$C = \left|\begin{array}{cccc|c} 3 & 7 & 5 & 8 & 3 \\ 2 & 4 & 4 & 5 & 2 \\ 4 & 7 & 2 & 8 & 2 \\ 9 & 7 & 3 & 8 & 3 \end{array}\right| \quad (d_i) \longrightarrow C' = \begin{vmatrix} 0 & 4 & 2 & 5 \\ 0 & 2 & 2 & 3 \\ 2 & 5 & 0 & 6 \\ 6 & 4 & 0 & 5 \end{vmatrix}$$

$$\begin{array}{ccccc} d_j & 0 & 2 & 0 & 3 \end{array}$$

$$\downarrow$$

$$C'' = \begin{vmatrix} 0 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \\ 2 & 3 & 0 & 3 \\ 6 & 2 & 0 & 2 \end{vmatrix}$$

А-3. Образование первоначальной системы независимых нулей. Необходимо отыскать строку в матрице C'' с наименьшим числом нулей. Отметить один из нулей этой строки и вычеркнуть все остальные нули этой строки и того столбца, в которых находится этот нуль. Аналогичные операции последовательно проводят для всех строк

$$C'' = \begin{vmatrix} 0^* & 2 & 2 & 2 \\ \cancel{0} & 0^* & 2 & \cancel{0} \\ 2 & 3 & 0^* & 3 \\ 6 & 2 & \cancel{0} & 2 \end{vmatrix}$$

Если назначения, которые получены при всех отмеченных нулях, являются полными (т.е. число отмеченных нулей равно n), то решение является оптимальным, в противном случае следует переходить к следующему этапу.

В. Последовательные итерации.

В-I. Поиск минимального набора строк и столбцов, содержащих нули.

Для проведения указанной процедуры необходимо осуществить следующие пометки строк и столбцов:

а) - пометить все строки, в которых не имеется ни одного отмеченного нуля (в рассматриваемом примере это строка 4 матрицы C'');

б) - пометить все столбцы, содержащие перечеркнутый нуль, хотя бы в одной из отмеченных строк (столбец 3 матрицы C'');

в) - пометить все строки, содержащие отмеченные нули, хотя бы в одном из отмеченных столбцов (строка 3 матрицы C'').

Действия б) и в) поочередно повторяются до тех пор, пока есть место отмечать.

$$C'' = \begin{array}{cc} \begin{array}{cccc} & & + & \end{array} & \\ \begin{vmatrix} 0^* & 2 & 2 & 2 \\ \cancel{0} & 0^* & 2 & \cancel{0} \\ 2 & 3 & 0^* & 3 \\ 6 & 2 & \cancel{0} & 2 \end{vmatrix} & \begin{array}{c} \\ \\ + \\ + \end{array} \end{array}$$

После этого необходимо зачеркнуть каждую непомеченную строку и каждый помеченный столбец (цель этих действий – провести минимальное число горизонтальных и вертикальных прямых, пересекающих, по крайней мере, один раз все нули).

$$C'' = \begin{array}{c c} \begin{array}{cccc} & & + & \end{array} & \\ \begin{vmatrix} 0^* & 2 & 2 & 2 \\ 0' & 0^* & 2 & 0' \\ 2 & 3 & 0^* & 3 \\ 6 & 2 & 0' & 2 \end{vmatrix} & \begin{array}{c} \\ \\ + \\ + \end{array} \end{array}$$

B-2. Перестановка нулей.

Необходимо взять наименьшее число из элементов невычеркнутых строк и стобцов (для нашего примера это число 2) и произвести следующие арифметические операции:

- вычесть его из каждого элемента принадлежащего невычеркнутым столбцам и строкам одновременно;
- прибавить к каждому числу принадлежащему вычеркнутым строкам и столбцам одновременно.

Математическая запись указанных процедур имеет вид:

$$h = \min_{\substack{i \in I' \\ j \in J'}} c''_{ij}$$

$$\left(c''_{ij}\right)_1 = \begin{cases} c_{ij} - h, \text{ если } (i \in I') \wedge (j \in J'), \\ c_{ij} + h, \text{ если } (i \notin I') \wedge (j \notin J'), \\ c_{ij}, \text{ если } [(i \in I') \wedge (j \notin J')] \vee [(i \notin I') \wedge (j \in J')] \end{cases} \quad (8)$$

Для рассматриваемого контрольного примера получим

$$C''_1 = \begin{vmatrix} 0^* & 2 & 4 & 2 \\ 0' & 0^* & 4 & 0' \\ 0 & 1 & 0^* & 1 \\ 4 & 0 & 0' & 0 \end{vmatrix}$$

Проделанные операции не изменят оптимального решения. Весь цикл расчета заново начинается с этапа В-I и продолжается до получения оптимального решения.

Для конкретно рассматриваемого примера имеем

$$C'' = \begin{vmatrix} 0^* & 2 & 4 & 2 \\ \cancel{0} & 0' & 4 & \cancel{0} \\ \cancel{0} & 1 & 0' & 1 \\ 4 & \cancel{0} & \cancel{0} & 0' \end{vmatrix}$$

Число отмеченных (независимых) нулей при этом равно 4, значит, назначение является полным, а решение оптимальным. Положения отмеченных нулей в матрице указывают размещение функциональных модулей на монтажном месте.

Оптимальное решение может быть не единственным. Для рассмотренной задачи минимальное значение целевой функции будет равно

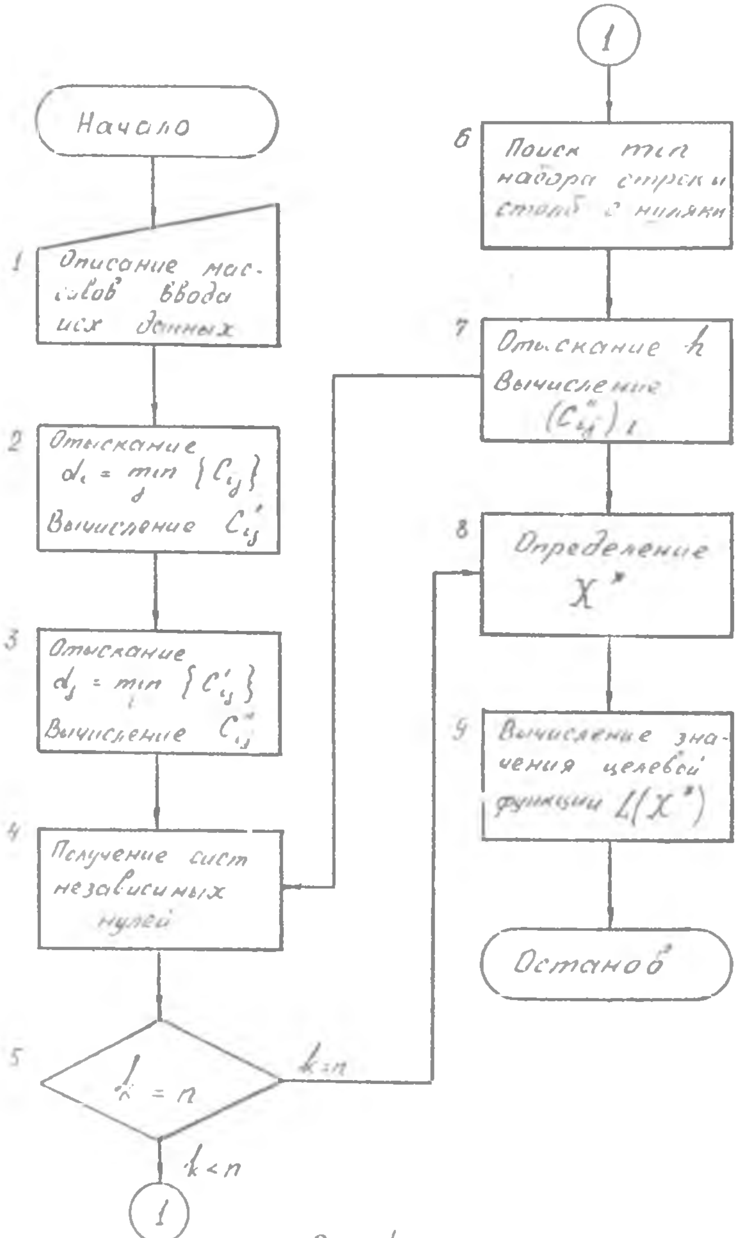
$$L = C_{11} + C_{22} + C_{33} + C_{44} = 3 + 4 + 2 + 8 = 17$$

Следует иметь в виду, что венгерский метод существенно использует возможность быстрого обращения к любому элементу матрицы и его буквальное использование при решении задач большого размера (требующих использование внешней памяти) может оказаться неэффективным.

IУ. Рекомендации по составлению программы

Блок-схема алгоритма изображена на рис. I.

Блок I. Для унификации программы венгерского метода размерность исходной матрицы (N) должна быть определена в первую очередь. Необходимо предусмотреть двумерные массивы под значения исходной матрицы $C(N,N)$, матрицы эквивалентных преобразователей $A(N,N)$. А также одномерные массивы для пометок столбцов $T(N)$ и строк $R(N)$. Для единообразия целесообразно ввести следующие обозначения:



Начало
1
Описание массивов ввода исх данных
2
Отыскание
$d_i = \min_j \{C_{ij}\}$
Вычисление C'_{ij}
3
Отыскание
$d_j = \min_i \{C'_{ij}\}$
Вычисление C''_{ij}
4
Получение сист независимых нулей
5
$k = n$
$k = n$
$k < n$
1
1
6
Поиск min набора строк и столб с нулями
7
Отыскание h
Вычисление $(C''_{ij})_1$
8
Определение X^*
9
Вычисление значения целевой функции $L(X^*)$
Останов

Рис. 1

- выделение строк и стобцов отмечать "- I;
- независимые нули матрицы через "- 2;
- перечеркнутые нули матрицы через "- 3".

Блоки 2 и 3. Осуществляется подготовительный этап согласно условиям (6) и (7). Преобразованная матрица заносится в $A(N,N)$.

Блок 4. Осуществляется циклический просмотр и отыскание строк с минимальным числом нулей. При ее нахождении осуществляется в матрице $A(N,N)$ пометка независимого нуля и вычеркнутых нулей в соответствующих строке и столбце. После полного перебора N строк переход к блоку 5.

Блок 5. При сравнении числа независимых нулей с размерностью исходной матрицы $(K=N)$ необходимо идти к вычислению значения целевой функции. В противном случае осуществляется поиск минимального набора строк и столбцов с нулями (блок 6).

Блок 6. Проводятся процедуры пометок строк и столбцов (в массивах $R(N)$ и $T(N)$) в соответствии с указаниями П-I (пункты а, б, в).

Блок 7. Используя данные массивы $R(N)$ и $T(N)$, отыскивается наименьший элемент из числа неотмеченных строк и столбцов и осуществляется преобразование в соответствии с (8).

Блоки 8 и 9. В исходной матрице $C(N,N)$ отыскиваются элементы, соответствующие положениям независимых нулей X^*, и вычисляется экстремальное значение целевой функции.

Все преобразования над матрицами осуществляются посредством циклического просмотра элементов строк и столбцов, при этом особое внимание следует обратить на последовательность выполнения циклов по I (строкам) и J (столбцам). Для целей контроля выполнения алгоритма целесообразно предусмотреть печать промежуточных результатов на экране монитора и выдачу соответствующих коментариев. Имеет смысл организовать печать в виде подпрограммы, что существенно сокращает объем программы.

У. Порядок выполнения лабораторной работы

1. Ознакомиться с теоретической частью работы.
2. На соответствующем алгоритмическом языке написать программы решения задачи о назначении венгерским методом.
3. Ввести программу в ЭВМ и в случае необходимости отладить ее.
4. Получить у преподавателя контрольный пример и, решив его, убедиться в правильности составления программы.
5. При необходимости осуществить запись программы на магнитную ленту (для доработки программы на следующих занятиях).
6. Составить отчет о лабораторной работе по следующей форме:
   - титульный лист,
   - необходимые краткие теоретические сведения,
   - программа алгоритма и ее описание,
   - контрольный пример с полученными результатами,
   - выводы по работе.

Оформление листов отчета должно соответствовать требованиям ЕСКД и текстовым документа .

УI. Контрольные вопросы

I. Как формулируется транспортная задача?
2. Каковы особенности задачи о назначениях?
3. Из каких этапов состоит алгоритм венгерского метода?
4. В чем состоит основной принцип задачи о назначениях?
5. В чем суть идеи алгоритма венгерского метода?
6. Объясните смысл поиска минимального набора строк и столбцов с нулями?
7. Сформулируйте принцип двойственности задач линейного программирования?
8. Как осуществляются эквивалентные преобразования исходной матрицы?
9. Изобразите блок-схему алгоритма венгерского метода и дайте соответствующие пояснения.
I0. Охарактеризуйте достоинства и недостатки венгерского метода.
II. Как осуществить пометку строк, столбцов, независимых нулей матриц?
I2. Как организовать подпрограмму построчной печати матрицы?
I3. Изобразите блок-схему фрагментов программы:
   - подготовительный этап преобразования матриц;
   - получение первоначальной системы независимых нулей;
   - поиск минимального набора строк и столбцов с нулями;
   - определение матрицы с новыми нулями (перестановка нулей).

Литература

1. Деньдобренко Б.Н., Малика А.Г.Автоматизация конструирования РЭА. -М.: Высшая школа, 1980. -284с.
2. Пашкеев С.Д., Минязов Р.И., Могилевский В.Д. Машинные методы оптимизации в технике связи. -М.: Связь, 1976. -272с.
3. Петренко А.И. Основы автоматизации проектирования. -Киев: Техника, 1982. -295с.
4. Зеленский А.В., Матюнин С.А. Составление БЭЙСИК-программ для конструкторского и технологического проектирования РЭА для микро-ЭВМ Д3-28. Метод.указания к лаб.раб. -Куйбышев: КуАИ, 1985. -34с.
5. ГОСТ 19.002-80
6. ГОСТ 19.003-80

Подписано в печать 5.09.86.
Бумага оберточная белая. Формат 60x84 1/16.
Усл.печ.лист. 0,93. Уч.изд.л. 0,8. Тир. 50.
Зак. 566. Бесплатно.

г.Куйбышев. КуАИ, ул.Ульяновская, 18,
участок оперативной полиграфии.