УДК 519.852

РЕШЕНИЕ ОТКРЫТОЙ ЗАДАЧИ НАЗНАЧЕНИЯ СТАНДАРТНЫМ СИМПЛЕКС-МЕТОДОМ

В.С. МУХА

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь

Поступила в редакцию 13 ноября 2013

Разработаны подход и алгоритм, позволяющие автоматизировать решение открытой задачи назначения произвольного размера стандартным симплекс-методом.

Ключевые слова: задача назначения, линейное программирование, целочисленное программирование, многомерные матрицы.

Постановка задачи

Закрытая (сбалансированная) задача назначения является бинарной целочисленной задачей линейного программирования [1]. Переменные, описывающие задачу, определяются следующим образом:

$$u_{i,j} = \begin{cases} 1, & \text{если работа } i \text{ назначена исполнителю } j, \\ 0, & \text{иначе}, \end{cases} \tag{1}$$

причем число работ равно числу исполнителей, т.е. $i, j = \overline{1,n}$. Пусть $\alpha_{i,j}$ – стоимость соответствующего назначения. Задача состоит в минимизации суммарной стоимости назначения $f = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j} u_{i,j}$ при ограничениях

$$\sum_{j=1}^{n} u_{i,j} = 1 = s_i, \; i = \overline{1,n}, \tag{2}$$

$$\sum_{i=1}^{n} u_{i,j} = 1 = b_j, \; j = \overline{1,n}. \tag{3}$$

Ограничения (2), (3) означают, что каждому исполнителю назначается только одна работа. Открытая (несбалансированная) задача возникает тогда, когда число работ не совпадает с числом исполнителей. Для решения такой задачи обычно предлагается приводить прямоугольную матрицу стоимостей назначений $\alpha = (\alpha_{i,j})$ к виду квадратной матрицы путем добавления нулевых строк или столбцов и решать затем закрытую задачу назначения. Однако такая процедура приводит к увеличению необходимой памяти и неудобна как в использовании, так и в интерпретации результатов оптимизации. Целесообразно иметь алгоритм решения открытой задачи назначения, свободный от указанной процедуры. Целесообразно также для ее решения использовать существующие широко распространенные и хорошо отлаженные программные средства линейного программирования.

Для разработки подобного алгоритма сформулируем открытую задачу назначения следующим образом. Минимизируется функция

$$f = \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{i,j} u_{i,j} \to \min_{u_{i,j}} \tag{4}$$

по переменным $u_{i,j}$ вида (1) при ограничениях

$$\sum_{j=1}^{n} u_{i,j} = s = 1,\ i = \overline{1,m}, \tag{5}$$

$$\sum_{i=1}^{m} u_{i,j} \le b = 1,\ j = \overline{1,n}, \tag{6}$$

если $m \le n$, и ограничениях

$$\sum_{j=1}^{n} u_{i,j} \le s = 1,\ i = \overline{1,m}, \tag{7}$$

$$\sum_{i=1}^{m} u_{i,j} = b = 1,\ j = \overline{1,n}, \tag{8}$$

если $m > n$. Кроме того, будем учитывать ограничения

$$u_{i,j} \ge 0,\ i = \overline{1,m},\ j = \overline{1,n}. \tag{9}$$

Задача (4)–(9) может быть решена с помощью программы linprog.m Matlab [2], предназначенной для решения задачи линейного программирования следующего вида:

$$f = a^T x \to \min_{x}, \tag{10}$$

$$A_{eq} x = b_{eq}, \tag{11}$$

$$A_{le} x \le b_{le}, \tag{12}$$

$$x \ge 0. \tag{13}$$

Предлагаемый в данной работе алгоритм основывается на компьютерном приведении задачи (4)–(9) к виду (10)–(13) и последующем использовании программы linprog.m.

Разработка алгоритма

Необходимым шагом для разработки алгоритма является формулировка задачи (4)–(9) в многомерно-матричной форме. Для этого введем двухмерные матрицы $\alpha = (\alpha_{i,j})$, $u = (u_{i,j})$, $i = \overline{1,m}$, $j = \overline{1,n}$.

Тогда целевую установку (4) задачи (4)–(9) можно записать следующим образом:

$$f = {}^{0,2}(\alpha u) \to \min_{u}. \tag{14}$$

Здесь и далее используется многомерно-матричный математический аппарат [3]. В частности, ${}^{0,2}(\alpha u)$ означает $(0,2)$-свернутое произведение матриц α и u.

Для многомерно-матричной записи ограничений (6), (8) сформируем трехмерную матрицу $c = (c_{k,i,j})$, $k = \overline{1,n}$, $i = \overline{1,m}$, $j = \overline{1,n}$, элементы которой определим формулой

$$c_{k,i,j} = \begin{cases} 1,\ k = j, \\ 0,\ k \ne j, \end{cases} \tag{15}$$

и вектор длины n с единичными компонентами $b = (b_j) = (1_j)$, $j = \overline{1,n}$. В этих обозначениях ограничения (6) и (8) запишутся соответственно в виде

$${}^{0,2}(cu) \le b,\ {}^{0,2}(cu) = b. \tag{16}$$

Для многомерно-матричной записи ограничений (5) и (7) сформируем трехмерную матрицу $d=(d_{k,i,j})$, $k=\overline{1,m}$, $i=\overline{1,m}$, $j=\overline{1,n}$, элементы которой определим формулой

$$d_{k,i,j}=\begin{cases}1,\ k=i,\\ 0,\ k\neq i,\end{cases} \qquad (17)$$

и вектор длины m с единичными компонентами $s=(s_i)=(1_i)$, $i=\overline{1,m}$.

Тогда условия (5) и (7) будут соответственно иметь вид:

$$^{0,2}(du)=s\,,\ ^{0,2}(du)\leq s\,. \qquad (18)$$

Запишем выражения (14), (16), (18) в виде, отражающем структуру участвующих в умножении матриц [3]:

$$f={}^{0,2}(\alpha_{(0,0,2)}u_{(2,0,0)})\to\min_{u_{(2,0,0)}}, \qquad (19)$$

$$^{0,2}(c_{(1,0,2)}u_{(2,0,0)})\leq b\,,\ ^{0,2}(c_{(1,0,2)}u_{(2,0,0)})=b\,, \qquad (20)$$

$$^{0,2}(d_{(1,0,2)}\,u_{(2,0,0)})=s\,,\ ^{0,2}(d_{(1,0,2)}\,u_{(2,0,0)})\leq s\,. \qquad (21)$$

Вводя матрицы, соответствующим образом ассоциированные с матрицами в левых частях выражений (19)–(21), и используя теорему об ассоциированных матрицах [3], получим следующую задачу линейного программирования в стандартной (классической) постановке:

$$f={}^{0,1}(\tilde{\alpha}_{(0,0,2)}\tilde{u}_{(2,0,0)})\to\min_{\tilde{u}_{(2,0,0)}} \qquad (22)$$

при ограничениях

$$^{0,1}(\tilde{d}_{(1,0,2)}\,\tilde{u}_{(2,0,0)})=s\,,\ ^{0,1}(\tilde{c}_{(1,0,2)}\tilde{u}_{(2,0,0)})\leq b\,, \qquad (23)$$

если $m\leq n$, и ограничениях

$$^{0,1}(\tilde{d}_{(1,0,2)}\,\tilde{u}_{(2,0,0)})\leq s\,,\ ^{0,1}(\tilde{c}_{(1,0,2)}\tilde{u}_{(2,0,0)})=b\,, \qquad (24)$$

если $m>n$, а также

$$\tilde{u}_{(2,0,0)})\geq 0 \qquad (25)$$

при любых m, n. Здесь $\tilde{d}_{(1,0,2)}$, $\tilde{c}_{(1,0,2)}$ – двухмерные матрицы, $\tilde{\alpha}_{(0,0,2)}$, $\tilde{u}_{(2,0,0)})$ – одномерные матрицы (векторы).

Приведенные выкладки приводят к следующему алгоритму решения задачи назначения.

1. Формируем матрицу стоимостей $\alpha=(\alpha_{i,j})$, $i=\overline{1,m}$, $j=\overline{1,n}$, в ее естественной форме.

2. Формируем матрицу c по формуле (15) и матрицу d по формуле (17).

3. Формируем матрицу $\tilde{\alpha}_{(0,0,2)}$, $(0,0,2)$-ассоциированную с матрицей α, матрицу $\tilde{c}_{(1,0,2)}$, $(1,0,2)$-ассоциированную с матрицей c и матрицу $\tilde{d}_{(1,0,2)}$, $(1,0,2)$-ассоциированную с матрицей d [3].

4. Формируем векторы $b=(b_j)$, $b_j=1$, $j=\overline{1,n}$ и $s=(s_i)$, $s_i=1$, $i=\overline{1,m}$.

5. Формируем правую часть ограничения (25), т.е. вектор длины mn с нулевыми компонентами.

6. Используя сформированные выше матрицы в качестве параметров программы linprog.m, решаем задачу (22)–(25) как стандартную задачу линейного программирования (находим оптимальное решение $\tilde{u}^*_{(2,0,0)}$ и оптимальное значение f^* целевой функции).

7. Полученное решение преобразуем в естественную форму, т.е. по полученной ассоциированной матрице $\tilde{u}^{*}_{(2,0,0)}$ формируем исходную для нее матрицу $u^{*}=(u^{*}_{i,j})$, $i=\overline{1,m}$, $j=\overline{1,n}$.

Данный алгоритм был оформлен в виде m-файла-функции Matlab. В качестве примера работы алгоритма и программы получено решение задачи назначения со следующими матрицами стоимостей при $m<n$ и $m>n$:

$$\alpha_1=\begin{pmatrix} \bar{5}, & 76, & 81, & 61, & 74, & 56 \\ 58, & 74, & \bar{9}, & 26, & 43, & 30 \\ 71, & \overline{44}, & 95, & 88, & 97, & 86 \\ 97, & 64, & 92, & 52, & \bar{8}, & 34 \end{pmatrix},\ \alpha_2=\begin{pmatrix} 33, & 60, & 43, & 96 \\ 48, & 03, & 38, & \overline{15} \\ 60, & 82, & \overline{17}, & 87 \\ \overline{17}, & 62, & 84, & 77 \\ 83, & 71, & 84, & 45 \\ 96, & \overline{10}, & 46, & 63 \end{pmatrix}.$$

Оптимальные назначения соответствуют стоимостям, помеченным в матрицах α_1 и α_2 чертой сверху. Целочисленность решения $u^{*}=(u^{*}_{i,j})$ достигается простым округлением полученного оптимального решения.

DECISION OF THE OPEN ASSIGNMENT PROBLEM BY STANDARD SIMPLEX METHOD

V.S. MUKHA

Abstract

The algorithm for the automatic decision of open assignment problem of any size by standard simplex method is designed.

Список литературы

1. *Вагнер Г.* Основы исследования операций. Т. 1. М., 1972.
2. *Кетков Ю.Л., Кетков А.Ю., Шульц М.М.* MATLAB 6x: программирование численных методов. СПб, 2004.
3. *Муха В.С.* Анализ многомерных данных. Минск, 2004.