УДК 519.161

О.Б. МАЦИЙ, А.В. МОРОЗОВ, А.В. ПАНИШЕВ

РЕКУРРЕНТНЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Аннотация. Предложен новый метод решения задачи о назначениях, основанный на рекурсивном получении ее оптимального решения. Задача о назначениях формулируется в перестановочно-матричной форме, что позволяет использовать матричный подход к построению оптимального решения. Алгоритм состоит в нахождении взвешенного паросочетания минимального суммарного веса в двудольном графе с $2n$ вершинами. Вычислительная схема рекуррентного метода решения задачи о назначениях представлена в форме, удобной для реализации на ЭВМ.

Ключевые слова: задача о назначениях, паросочетание, двудольный граф, увеличивающий путь.

ВВЕДЕНИЕ

Широко известные методы решения задачи о назначениях (ЗН), такие как венгерский метод, метод Кана–Мункреса и метод потенциалов, построены с использованием различных подходов, применяемых в комбинаторной оптимизации, и характеризуются разной временной сложностью, не меньшей чем $O(n^3)$, где n — порядок матрицы стоимостей [1]. В [2] изложен алгоритм решения одного из вариантов ЗН, временная оценка которого понижена до $O(n^2)$, а также показано, что он выполняет функции процедуры, встроенной в метод ветвей и границ для быстрого вычисления более точных нижних оценок стоимости замкнутых маршрутов в задаче коммивояжера. Алгоритм состоит в нахождении взвешенного паросочетания минимального суммарного веса в двудольном графе с $2n$ вершинами с использованием принятых в [2] понятий кратчайшего увеличивающего пути. В данной статье описан рекуррентный метод решения ЗН, развивающий результаты работ [2, 3] и технически упрощающий наиболее распространенный венгерский метод.

ПЕРЕСТАНОВОЧНО-МАТРИЧНАЯ ПОСТАНОВКА ЗАДАЧИ

Опишем рассматриваемый алгоритм решения ЗН, исходя из следующей ее формулировки.

Для матрицы стоимостей (весов) $C = [c_{ij}]_n$, где $c_{ij} \in R_0^+$ или $c_{ij} = \infty$, R_0^+ — множество неотрицательных действительных чисел, найти

$$C(\sigma) = \min_{\pi} \sum_{i=1}^{n} c_{\pi[i]}. \tag{1}$$

Здесь $\pi = (\pi[1], \pi[2], \ldots, \pi[n])$ — перестановка множества $\{1, 2, \ldots, n\}$ номеров столбцов матрицы C; $\sigma = (\sigma[1], \sigma[2], \ldots, \sigma[n])$ — оптимальная перестановка стоимостью $C(\sigma) = \sum_{i=1}^{n} c_{\sigma[i]}$, $c_{\sigma[i]} \neq \infty$, $i = \overline{1, n}$, или решение ЗН. Перестановку $\pi = (\pi[1], \pi[2], \ldots, \pi[n])$, в которой $c_{\pi[i]} \neq \infty$, $i = \overline{1, n}$, назовем допустимым решением ЗН. Заметим, что ЗН с матрицей стоимостей, содержащей элементы $c_{ij} = \infty$, может не иметь решения. В этом случае необходимо установить, что множество допустимых решений задачи пусто. Будем искать σ, пошагово увеличивая на единицу число элементов k, $k = \overline{1, n}$, последовательности, образую-

© О.Б. Маций, А.В. Морозов, А.В. Панишев, 2015

щей определенную часть допустимого решения ЗН. Рассмотрим свойства (условия выполнения) этой последовательности и способ ее построения.

Любая часть допустимого решения ЗН, включающая k элементов, однозначно определяет подматрицу $[c_{i_s j_t}]_k$ матрицы C, $i_1 < i_2 < \ldots < i_s < \ldots < i_k$, $j_1 < j_2 < \ldots$ $\ldots < j_t < \ldots < j_k$. Пусть выполняются следующие условия:

а) последовательность $\pi_k = (\pi_k[i_1], \pi_k[i_2], \ldots, \pi_k[i_s], \ldots, \pi_k[i_k])$, $\pi_k[i_s] \in$ $\in \{j_1, j_2, \ldots, j_t, \ldots, j_k\}$, $k = \overline{1, n-1}$, является решением ЗН для подматрицы $[c_{i_s j_t}]_k$;

б) стоимость π_k не превышает стоимости решения ЗН для любой подматрицы порядка k матрицы C.

Если существует эффективная процедура преобразования последовательности π_k в последовательность π_{k+1}, $k = \overline{0, n-1}$, и задача (1) имеет решение, то для нахождения $\sigma = \pi_n$ требуется n шагов.

Матричный подход к построению оптимального назначения. Покажем, как строится последовательность π_k, $k = \overline{1, n}$.

Исходная последовательность $\pi_1 = (\pi_1[i_1])$ определяется тривиально: в матрице C находится $c_{lr} = \min\{c_{ij} \mid 1 \le i, j \le n\}$ и, следовательно, $i_1 = l$, $\pi[i_1] = r$. Чтобы получить $\pi_2 = (\pi_2[i_1], \pi_2[i_2])$, определим $c_{ms} = \min\{c_{ij} \mid i \ne l, j \ne r\}$, $c_{lp} =$ $= \min\{c_{lj} \mid j \ne r\}$, $c_{vr} = \min\{c_{ir} \mid i \ne l\}$ (рис. 1).

		p	q		r		s	
m		c_{mp}					c_{ms}	
l		c_{lp}			c_{lr}			
v					c_{vr}		c_{vs}	
w			c_{wq}					

Рис. 1

Нетрудно видеть, что если $c_{lr} + c_{ms} \le c_{lp} + c_{vr}$, то условия а) и б) выполняются для последовательности π_2, в которой $i_1 = l, \pi_2[i_1] = r, i_2 = m, \pi_2[i_2] = s$. Ей соответствует подматрица

	r	s
l	c_{lr}	
m		c_{ms}

.

В противном случае этим условиям удовлетворяет последовательность π_2 с элементами $i_1 = l, \pi_2[i_1] = p, i_2 = v, \pi_2[i_2] = r$ и подматрицей

	p	r
l	c_{lp}	c_{lr}
v		c_{vr}

.

Преобразуем последовательность π_2 в последовательность $\pi_3 = (\pi_3[i_1],$ $\pi_3[i_2], \pi_3[i_3])$. Не теряя общности, предположим, что $\pi_2 = (\pi_2[l] = r, \pi_2[m] = s)$. Найдем $c_{wq} = \min\{c_{ij} \mid i \ne l, m;\ j \ne s, r\}$ и $MIN1 = c_{lr} + c_{ms} + c_{wq}$. Заметим, что $c_{wq} = c_{ms}$, если $\pi_2 = (\pi_2[l] = p, \pi_2[v] = r)$. Преобразование π_2 в π_3 является результатом решения следующей вспомогательной задачи.

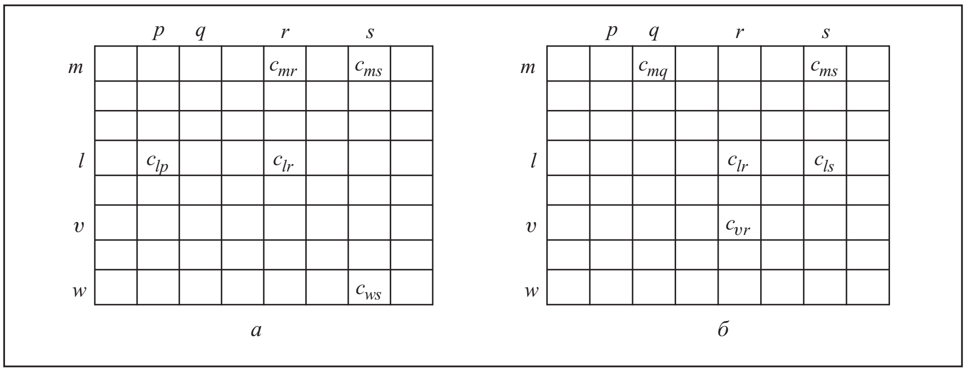

Рис. 2

Для строк l, m и столбцов r, s, определяемых числами c_{lr} и c_{ms} матрицы C, требуется найти тройку элементов с минимальной суммой их значений при условии, что любые два элемента из тройки должны находиться в трех разных строках, включающих l, m, и трех разных столбцах, включающих r, s.

Если искомая тройка не содержит c_{lr}, но содержит c_{ms}, то сумма значений ее элементов ограничена снизу величиной $S_1 = c_{vr} + c_{lp} + c_{ms}$, где

$$c_{vr} = \min\{c_{ir} \mid i \neq l, m\},\ c_{lp} = \min\{c_{lj} \mid j \neq r, s\}.$$

Пусть решение вспомогательной задачи является тройкой, в которую входит c_{lr} и не входит c_{ms}. Тогда оно формирует сумму $S_2 = c_{lr} + c_{mp} + c_{vs}$, где

$$c_{mp} = \min\{c_{mj} \mid j \neq s, r\},\ c_{vs} = \min\{c_{is} \mid i \neq l, m\}.$$

Элементы решения вспомогательной задачи, не содержащего c_{lr} и c_{ms}, определяют величину $S_3 = \min\{c_{lp} + c_{mr} + c_{ws}, c_{mq} + c_{ls} + c_{vr}\}$. Здесь

$$c_{ws} = \min\{c_{is} \mid i \neq l, m\},\ c_{mr} = \min\{c_{mj} \mid j \neq p, s\} \ \text{(рис. 2, } a\text{)},$$

$$c_{ls} = \min\{c_{is} \mid i \neq m, v\},\ c_{mq} = \min\{c_{mj} \mid j \neq r, s\} \ \text{(рис. 2, } б\text{)}.$$

Определим величину, равную $MIN2 = \min\{S_1, S_2, S_3\}$. Ясно, что она соответствует искомой последовательности π_3, если $MIN2 \leq MIN1$, иначе $\pi_3 = (\pi_3[l] = r, \pi_3[m] = s,\ \pi_3[w] = q)$.

В общем случае последовательность $\pi_k = (\pi_k[i_1], \pi_k[i_2], \ldots, \pi_k[i_s], \ldots, \pi_k[i_k])$ со свойствами а) и б) преобразуется в последовательность $\pi_{k+1} = (\pi_{k+1}[i_1],\ \pi_{k+1}[i_2], \ldots, \pi_{k+1}[i_r], \ldots, \pi_{k+1}[i_{k+1}])$ с этими же свойствами следующим образом.

В матрице C определяется

$$c_{\pi_k[i_{k+1}]} = \min\{c_{ij} \mid i \neq i_1, i_2, \ldots, i_s, \ldots, i_k,\ j \neq \pi_k[i_1], \pi_k[i_2], \ldots, \pi_k[i_s], \ldots, \pi_k[i_k]\},$$

формируется последовательность $\pi^1_{k+1} = (\pi_k, \pi_k[i_{k+1}])$ и вычисляется

$$MIN1 = \sum_{s=1}^{k} c_{\pi_k[i_s]} + c_{\pi_k[i_{k+1}]}.$$

Далее решается задача поиска $k+1$ элементов, которые в матрице C формируют минимальную сумму своих значений и располагаются в разных строках и столбцах, включая все строки и столбцы с номерами, заданными величинами $c_{\pi_k[i_1]}, c_{\pi_k[i_2]}, \ldots, c_{\pi_k[i_s]}, \ldots, c_{\pi_k[i_k]}$. Обозначим эту сумму $MIN2$. Найденные элементы образуют искомую последовательность $\pi_{k+1} = \pi^2_{k+1}$, если $MIN2 \leq MIN1$. Иначе $\pi_{k+1} = \pi^1_{k+1}$.

ОБОСНОВАНИЕ И ОПИСАНИЕ РЕКУРРЕНТНОГО АЛГОРИТМА

Предложенная схема поиска оптимального назначения является основой алгоритма, в котором решение задачи (1) находится исключительно средствами теории паросочетаний для двудольных графов [1].

Матрице стоимостей ЗН C взаимно-однозначно соответствует двудольный граф (X, Y, U), в котором $|X| = |Y| = n$ и вершина $i \in X$ соединена с вершиной $j \in Y$ ребром $(i, j) \in U$ с весом $c_{ij} \neq \infty$.

Паросочетание графа образует такое множество ребер, в котором никакие два из них не имеют общей вершины. Ребра, не входящие в паросочетание, называются свободными. Максимальное паросочетание имеет наибольшее число ребер. Вершина, принадлежащая ребру паросочетания, называется насыщенной, остальные вершины графа — свободными. Ребро (i, j), входящее в паросочетание, обозначим $[i, j]$. Вершина j ребра $[i, j]$ определяется как напарник i. Паросочетание, насыщающее все вершины графа, является совершенным. В двудольном графе (X, Y, U), где $|X| = |Y| = n$, мощность совершенного паросочетания, если оно существует, равна n. Решение ЗН σ состоит в построении в двудольном взвешенном графе (X, Y, U), $|X| = |Y| = n$, совершенного паросочетания с минимальным суммарным весом ребер [1].

Пусть в графе имеется паросочетание M. Простой путь называется чередующимся относительно паросочетания M, если в него входит каждое второе ребро. Чередующийся путь, который начинается и заканчивается ребрами, не принадлежащими паросочетанию M, называется увеличивающим относительно паросочетания M. Следовательно, если $(i_0, j_1, i_1, j_2, i_2, j_3, \ldots, i_{k-1}, j_k, i_k, j_{k+1})$ — увеличивающий путь в двудольном графе (X, Y, U), то в нем свободны вершины i_0, j_{k+1} и $k + 1$ ребер $(i_0, j_1), (i_1, j_2), (i_2, j_3), \ldots, (i_{k-1}, j_k), (i_k, j_{k+1})$. Остальные k ребер пути образуют паросочетание $\pi_k = ([i_1, j_1], [i_2, j_2], [i_3, j_3], \ldots, [i_k, j_k])$. Длина пути определяется числом встречающихся в нем ребер.

Предположим, что во взвешенном графе (X, Y, U), соответствующем матрице стоимостей ЗН C, не имеется паросочетания. Тогда каждое ребро в (X, Y, U) представляет собой увеличивающийся путь длины 1, а ребро с минимальным весом образует паросочетание или исходную последовательность π_1.

Пусть в графе (X, Y, U) построено паросочетание $\pi_k = ([i_1, j_1], [i_2, j_2], \ldots$ $\ldots, [i_l, j_l], \ldots, [i_k, j_k])$ с минимальной суммой $C(\pi_k)$ весов ребер среди всех паросочетаний мощностью k. Преобразуем π_k в паросочетание $\pi_{k+1} = ([i'_1, j'_1],$ $[i'_2, j'_2], \ldots, [i'_l, j'_l], \ldots, [i'_k, j'_k], [i'_{k+1}, j'_{k+1}])$, у которого величина $C(\pi_{k+1})$ достигает минимума на множестве Π_{k+1} всех паросочетаний мощностью $k+1$.

Паросочетание π_k разбивает множества X, Y соответственно на подмножества насыщенных вершин $I_k = \{i_1, i_2, \ldots, i_l, \ldots, i_k\}$, $J_k = \{j_1, j_2, \ldots, j_l, \ldots, j_k\}$ и на подмножество свободных вершин $X - I_k = \{i_{k+1}, i_{k+2}, \ldots, i_s, \ldots, i_n\}$, $Y - J_k = \{j_{k+1}, j_{k+2}, \ldots, j_q, \ldots, j_n\}$. Найдем свободное ребро с весом

$$c_{i_r j_p} = \min\{c_{i_s j_q} \mid i_s \in X - I_k, j_q \in Y - J_k\}, \tag{2}$$

и, присоединив его к паросочетанию π_k, получим паросочетание $\pi'_{k+1} = \pi_k \cup$ $\cup [i_r, j_p]$ стоимостью $MIN1 = C(\pi_k) + c_{i_r j_p}$. Если π'_{k+1} не является паросочетанием π_{k+1} с минимальной суммой весов на множестве Π_{k+1}, то $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$.

Лемма 1. Пусть $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$. Тогда $\pi_{k+1} = (P_{k+1} - \pi_k) \cup (\pi_k - P_{k+1})$, где P_{k+1} — кратчайший увеличивающий путь относительно паросочетания π_k.

Доказательство. Построим кратчайший увеличивающий путь P_{k+1} относительно паросочетания π_k, и, следовательно, паросочетания π^2_{k+1}. Покажем, что

$\pi_{k+1} = \pi^2_{k+1}$. Ясно, что $C(\pi_k) \le C(\pi^2_{k+1})$. В величинах $C(\pi_k)$ и $C(\pi^2_{k+1})$ выделим общее слагаемое $C_0 = \sum_{\pi_k[i_s] \in \pi_k \cap \pi^2_{k+1}} c_{\pi_k[i_s]}$ и получим $C(\pi_k) - C_0 \le C(\pi^2_{k+1}) - C_0$. Однако $\pi_k \cap \pi^2_{k+1} = \pi_k - P_{k+1}$, а $C(\pi^2_{k+1}) - C_0$ — сумма весов ребер множества $P_{k+1} - \pi_k$, причем она минимальна, когда P_{k+1} — кратчайший увеличивающий путь относительно паросочетания π_k. Следовательно, π^2_{k+1} — паросочетание с минимальной суммой весов ребер на множестве $\Pi_{k+1} - \{\pi'_{k+1}\}$, т.е. $\pi_{k+1} = \pi^2_{k+1}$. ■

Пусть вершина $j_l \in J_k$, $l \in \{1, 2, \ldots, k\}$, является концом свободного ребра (i_r, j_l) с весом

$$c_{i_r j_l} = \min \{c_{i_s j_l} \mid i_s \in X - I_k\}, \tag{3}$$

а вершина $i_f \in I_k$, $f \in \{1, 2, \ldots, k\}$, — началом свободного ребра (i_f, j_p) с весом

$$c_{i_f j_p} = \min \{c_{i_f j_q} \mid j_q \in Y - J_k\}. \tag{4}$$

Обозначим X_k, $|X_k| \le k$, множество свободных вершин i_r, которые инцидентны ребру (i_r, j_l) с весом, определенным в (3), а Y_k, $|Y_k| \le k$, — множество свободных вершин j_p, инцидентных ребру (i_f, j_p) с весом, определенным в (4). Ясно, что кратчайший увеличивающий путь относительно паросочетания π_k начинается в вершине множества X_k и заканчивается в вершине множества Y_k.

Построим вспомогательный взвешенный орграф (V, A), в котором множество вершин $V = \{i_0\} \cup X_k \cup I_k \cup Y_k$, для поиска кратчайшего увеличивающего пути P_{k+1} относительно паросочетания π_k.

Множество дуг A орграфа (V, A) представлено разбиением на подмножества A_0, A_1, A_2, A_3. Подмножество A_0 содержит $|X_k|$ дуг (i_0, i_r) с нулевым весом, $i_r \in X_k$. В подмножество A_1 входит дуга (i_r, i_l), $i_r \in X_k$, $i_l \in I_k$, если и только если, вершина j_l ребра (i_r, j_l) — напарник вершины i_l, $[i_l, j_l] \in \pi_k$. Дуга (i_r, i_l) имеет вес $c(i_r, i_l) = c_{i_r j_l} + c_{i_l j_l}$. Дуга $(i_d, i_l) \in A_2$, i_d, $i_l \in I_k$, тогда и только тогда, когда вершина j_l ребра (i_d, j_l), $j_l \in J_k$, является напарником i_l, $[i_l, j_l] \in \pi_k$. Дуга (i_d, i_l) имеет вес $c(i_d, i_l) = c_{i_d j_l} + c_{i_l j_l}$. Подмножество A_3 включает дугу (i_f, j_p), $i_f \in I_k$, $j_p \in Y_k$, если вершины i_f и j_p соединены в графе (X, Y, U) ребром (i_f, j_p). Дуга (i_f, j_p) имеет вес $c(i_f, j_p) = c_{i_f j_p}$.

На рис. 3 представлен граф (X, Y, U), для которого получено паросочетание π_k, $k = 4$, его ребра изображены жирными линиями. Вершины i_5 и i_6 образуют множество X_4, а вершины j_5 и j_6 — множество Y_4. Ребра (i_5, j_1), (i_6, j_2), (i_6, j_3) имеют вес, определенный в (3), а веса ребер (i_2, j_5), (i_3, j_6), (i_4, j_6) определены в (4).

Вспомогательный орграф (V, A), построенный на графе (X, Y, U) и соответствующий паросочетанию π_4, изображен на рис. 4. Множество вершин V содержит вместе с вершиной i_0 подмножества $X_4 = \{i_5, i_6\}$, $I_4 = \{i_1, i_2, i_3, i_4\}$, $Y_4 = \{j_5, j_6\}$. Множество дуг A образуют подмножества $A_0 = \{(i_0, i_5), (i_0, i_6)\}$, $A_1 = \{(i_5, i_1), (i_6, i_2), (i_6, i_3)\}$, $A_2 = \{(i_1, i_3), (i_1, i_4), (i_3, i_1)\}$, $A_3 = \{(i_2, j_5), (i_3, j_6), (i_4, j_6)\}$.

Из способа построения орграфа (V, A) следует, что множество путей, достижимых из вершины i_0 во все вершины $j_p \in Y_k$, совпадает с множеством увеличивающих путей относительно паросочетания π_k, соединяющих в графе (X, Y, U) каждую вершину $i_r \in X_k$ с каждой вершиной $j_p \in Y_k$. Если в графе (V, A) построен кратчайший путь из вершины i_0 в вершину $j_p \in Y_k$, то он содержит одну из дуг $(i_r, i_l) \in A_1$ и одну из дуг $(i_f, j_p) \in A_3$. Пусть $j_m \in Y_k$ — вершина пути,

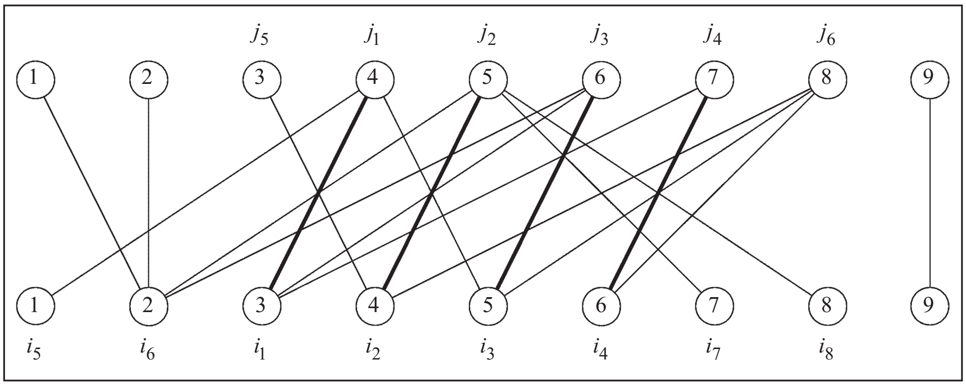

Рис. 3

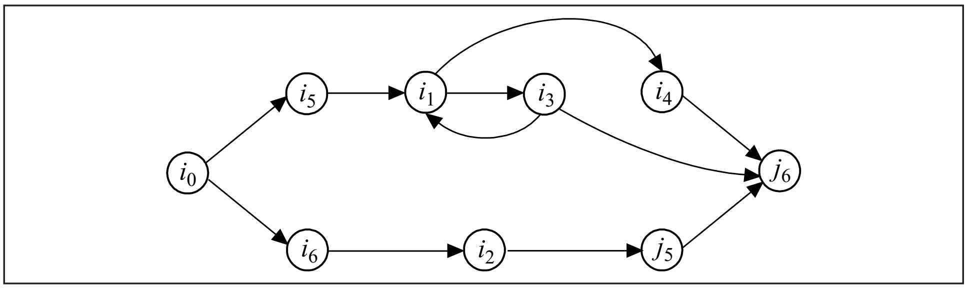

Рис. 4

кратчайшего среди всех путей из i_0 в остальные вершины множества Y_k. Очевидно, такой путь в графе (X, Y, U) определяет кратчайший увеличивающий путь P_{k+1} относительно π_k.

Следовательно, для нахождения в графе (X, Y, U) паросочетания π_{k+1}, стоимость которого минимальна на множестве всех паросочетаний π_{k+1} мощности $k+1$, достаточно:

— определить паросочетание $\pi'_{k+1} = \pi_k \cup [i_r, j_p]$ и суммарный вес его ребер $MIN1 = C(\pi_k) + c_{i_r j_p}$, где π_k — паросочетание минимальной стоимости на множестве всех паросочетаний π_k мощности k, $c_{i_r j_p}$ — вес ребра $[i_r, j_p]$, определенный в (2);

— найти паросочетание π^2_{k+1} и $MIN2 = C(\pi^2_{k+1})$, построив кратчайший увеличивающий путь P_{k+1} относительно паросочетания π_k;

— положить $\pi_{k+1} = \pi^1_{k+1}$, если $MIN1 \le MIN2$, и $\pi_{k+1} = \pi^2_{k+1}$ в противном случае.

Далее приведен алгоритм поиска решения ЗН, рекурсивно определяющий в двудольном графе (X, Y, U), $|X| = |Y| = n$, паросочетания π_k, которые содержат k $(k = \overline{1, n})$ ребер с минимальным суммарным весом. Предлагаемый алгоритм состоит из такого же числа этапов и имеет такую же временную сложность, что и наилучший из известных методов оптимального назначения — венгерский метод [1].

Алгоритм решения ЗН для матрицы стоимостей $\boldsymbol{C = [c_{ij}]_n}$, $n \ge 2$, элементы которой принимают значения из множества неотрицательных действительных чисел или равны ∞. Решение представлено совершенным паросочетанием $\pi = \pi_n$ с минимальной суммой $C(\pi)$ весов ребер $[i, j]$ в двудольном графе (X, Y, U), $|X| = |Y| = n$, $i \in X$, $j \in Y$, где $c_{ij} \in R_0^+$, если $(i, j) \in U$, иначе $c_{ij} = \infty$.

Шаг 0. Положить $k=1$ и найти $c_{i_k j_k}=\min\{c_{ij} \mid i,\, j=\overline{1,n}\}$, положить $I_k=\{i_k\}$, $J_k=\{j_k\}$, $\pi_k=\{[i_k, j_k]\}$, $C(\pi_k)=c_{i_k j_k}$.

Шаг 1. Присвоить $k=k+1$; если $k>n$, то завершение работы алгоритма с результатом: построено решение ЗН π.

Шаг 2. Найти $c_{i_k j_k}=\min\{c_{ij} \mid i\in X-I_{k-1},\, j\in Y-J_{k-1}\}$; если $c_{i_k j_k}=\infty$, то положить $MIN1=\infty$, иначе $\pi_k^1=\pi_{k-1}\cup[i_k, j_k]$, $MIN1=C(\pi_k^1)$.

Шаг 3. Найти все i_r такие, что для $j_l\in J_{k-1}$, $l\in\{1,2,\dots,k-1\}$, $c_{i_r j_l}=$ $=\min\{c_{ij_l} \mid i\in X-I_{k-1}\}\neq\infty$, и сформировать из них список X_k; если $X_k=\varnothing$, то положить $MIN2=\infty$ и перейти к шагу 6.

Шаг 4. Найти все j_p такие, что для $i_l\in I_{k-1}$, $l\in\{1,2,\dots,k-1\}$, $c_{i_l j_p}=$ $=\min\{c_{i_l j} \mid j\in J_{k-1}\}\neq\infty$, и сформировать из них список Y_k; если $Y_k=\varnothing$, то положить $MIN2=\infty$ и перейти к шагу 6.

Шаг 5. Построить взвешенный орграф (V, A), $V=\{i_0\}\cup X_k\cup I_{k-1}\cup Y_k$ и выполнить в нем поиск пути, кратчайшего на множестве всех путей в вершины Y_k, достигаемые из i_0; если построен такой путь, то в графе (X, Y, U) найти соответствующий ему кратчайший увеличивающий путь P_k относительно паросочетания π_{k-1} и определить $\pi_k^2=(P_k-\pi_{k-1})\cup(\pi_{k-1}-P_k)$, $MIN2=C(\pi_k^2)$, иначе положить $\pi_k^2=\varnothing$, $MIN2=\infty$.

Шаг 6. Если $MIN1=MIN2=\infty$, то завершение работы алгоритма с результатом: не существует для матрицы $[c_{ij}]_n$ решения ЗН; если $MIN1\neq\infty$ или $MIN2\neq\infty$, а также если $MIN1\le MIN2$, то $\pi_k=\pi_k^1$, $I_k=I_{k-1}\cup\{i_k\}$, $J_k=J_{k-1}\cup\{j_k\}$, иначе (если $MIN1>MIN2$) положить $\pi_k=\pi_k^2$, определить $I_k=\{i_l \mid l=\overline{1,k};\, [i_l, j_l]\in\pi_k^2\}$, $J_k=\{j_l \mid l=\overline{1,k},\, [i_l, j_l]\in\pi_k^2\}$ и перейти к шагу 1.

Теорема 1. Решение ЗН σ корректно находится построением в двудольном графе (X, Y, U), $|X|=|Y|=n$, соответствующем ее матрице стоимостей $[c_{ij}]_n$, последовательностей паросочетаний π_k, $k=\overline{1,n}$, где π_k — паросочетание, содержащее k ребер с минимальным весом, $\sigma=\pi_n$.

Доказательство. Пусть построено паросочетание π_{k-1}, $k=\overline{2,n}$. Выполнение алгоритма завершается, если, во-первых, не найдено ребра, объединение которого с π_{k-1} давало бы π_k^1, и во-вторых, во вспомогательном графе не существует пути из вершины i_0 в любую другую вершину множества Y_k, а следовательно — и увеличивающего пути относительно текущего паросочетания π_{k-1}, которое максимально [1].

Чтобы оценить трудоемкость решения ЗН, заметим, что оно строится в результате выполнения n этапов, каждый из которых увеличивает паросочетание на одно ребро.

На первом этапе определяется π_1 в худшем случае за время $O(n^2)$. На каждом следующем этапе строятся паросочетания π_k^1 и π_k^2, $k=\overline{2,n}$. Время построения π_k^1 равно $O((n-k+1)^2)$. Для определения π_k^2 требуется: $2(n-k+1)$ операций поиска вершин множеств X_k, Y_k; $k+1$ операций построения вспомогательного орграфа; $O(k^2)$ операций построения в нем алгоритмом Дейкстры кратчайшего пути; $O(k)$ операций с множествами. Поэтому время каждого k-го этапа ограничено величиной $O(n^2)$. ■

Рассмотим пример работы предложенного алгоритма. Исходная матрица $[c_{ij}]_n$ имеет вид

	1	2	3	4	5	6	7
1	∞	∞	2	8	9	0	∞
2	15	∞	3	0	24	0	24
3	5	∞	∞	5	0	2	∞
4	∞	10	2	15	23	∞	0
5	∞	0	0	∞	∞	∞	∞
6	0	15	10	3	17	∞	∞
7	14	0	24	2	2	15	21

Выполним алгоритм.

Шаг 0. Присвоим $X = Y = \{1, 2, 3, 4, 5, 6, 7\}$, $k = 1$, $C_{i_1 j_1} = \min\{c_{ij} \mid i, j = \overline{1, 7}\} = c_{16} = 0$, $I_1 = \{1\}$, $J_1 = \{6\}$, $X_1 = Y_1 = \varnothing$, $\pi_1 = \{[1, 6]\}$, $C(\pi_1) = c_{16} = 0$.

Шаг 1. Положим $k = 2$.

Шаг 2. Определяем $c_{i_2 j_2} = \min\{c_{ij} \mid i \neq 1, j \neq 6\} = c_{24} = 0$, $\pi_2^1 = \pi_1 \cup [2, 4] = \{[1, 6], [2, 4]\}$, $MIN1 = C(\pi_2^1) = c_{16} + c_{24} = 0 + 0 = 0$.

Шаг 3. Для $6 \in J_1$ определяем $c_{i_r 6} = \min\{c_{i6} \mid i \neq 1\} = c_{26} = 0$, $X_2 = \{2\}$.

Шаг 4. Для $1 \in I_1$ находим $c_{1 j_p} = \min\{c_{1j} \mid j \neq 6\} = c_{13} = 2$, $Y_2 = \{3\}$.

Шаг 5. Взвешенный граф (V, A) представляет собой путь $(i_0, 2, 1, 3)$, где $2 \in X_2$, $1 \in I_1$, $3 \in Y_2$, $C_{i_0 2} = 0$, $c(2, 1) = c_{26} + c_{16} = 0$, $c(1, 3) = 2$. Этому пути в подграфе графа (X, Y, U) соответствует кратчайший увеличивающий путь $P_2 = (2, 6, 1, 3)$ относительно паросочетания $\pi_1 = \{[1, 6]\}$. Путь P_2 образуют ребра $(2, 6)$, $[1, 6]$, $(1, 3)$, поэтому $\pi_2^2 = \{[2, 6], [1, 3]\}$, $MIN2 = c_{26} + c_{13} = 0 + 2 = 2$.

Шаг 6. Так как $MIN1 < MIN2$, то $\pi_2 = \pi_2^1 = \{[1, 6], [2, 4]\}$, $I_2 = \{1, 2\}$, $J_2 = \{6, 4\}$. Перейти к шагу 1.

Шаг 1. Положим $k = 3$.

Шаг 2. Определяем $c_{i_3 j_3} = \min\{c_{ij} \mid i \neq 1, 2;\ j \neq 6, 4\} = c_{35} = 0$, $\pi_3^1 = \pi_2 \cup [3, 5] = \{[1, 6], [2, 4], [3, 5]\}$, $MIN1 = C(\pi_3^1) = 0$.

Шаг 3. Так как $I_2 = \{1, 2\}$, $J_2 = \{6, 4\}$, то $c_{i_r 6} = \min\{c_{i6} \mid i \neq 1, 2\} = c_{36} = 2$, $c_{i_r 4} = \min\{c_{i4} \mid i \neq 1, 2\} = c_{74} = 2$ и $X_3 = \{3, 7\}$.

Шаг 4. Определяем $c_{1 j_p} = \min\{c_{1j} \mid j \neq 6, 4\} = c_{13} = 2$, $c_{2 j_p} = \min\{c_{2j} \mid j \neq 6, 4\} = c_{23} = 3$, $Y_3 = \{3\}$.

Шаг 5. Подграфу двудольного графа (X, Y, U), определенному на множестве вершин $X_3 \cup I_2 \cup J_2 \cup Y_3$ (рис. 5), соответствует вспомогательный граф (рис. 6) с весами дуг $c_{i_0 3} = c_{i_0 7} = 0$, $c(3, 1) = c_{36} + c_{16} = 2 + 0 = 2$, $c(1, 2) = c_{14} + c_{24} = 8 + 0 = 8$, $c(2, 1) = c_{26} + c_{16} = 0 + 0 = 0$, $c(7, 2) = c_{74} + c_{24} = 0 + 2 = 2$, $c(1, 3) = 2$, $c(2, 3) = 3$, $3 \in Y_3$.

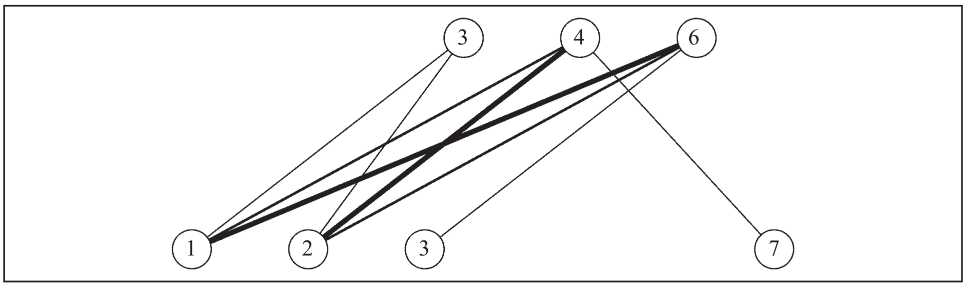

Рис. 5

Кратчайший путь $(3, 1, 3 \in Y)$, связывающий в орграфе (V, A) вершины из X_3 с вершинами из Y_3, состоит из ребер $(3, 6)$, $[1, 6]$, $(1, 3)$ подграфа графа (X, Y, U), которые образуют кратчайший увеличивающий путь P_3 относительно паросочетания $\pi_2 = \{[1, 6], [2, 4]\}$. Поэтому $\pi_3^2 = \{[2, 4], [3, 6], [1, 3]\}$, $MIN2 = c_{24} + c_{36} + c_{13} = 0 + 2 + 2 = 4$, а $MIN1 = c_{16} + c_{24} + c_{35} = 0$, следовательно, $\pi_3 = \pi_3^1$.

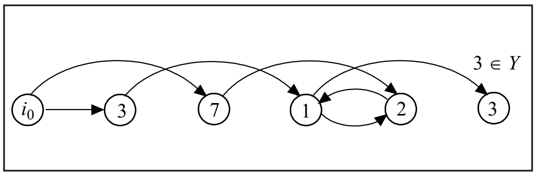

Рис. 6

После выполнений алгоритма для $k = 4, 5, 6, 7$ получаем решение ЗН $\sigma = \{[1, 6], [2, 4], [3, 5], [4, 7], [5, 3], [6, 1], [7, 2]\}$, $C(\sigma) = 0$.

ЗАКЛЮЧЕНИЕ

Предложенный метод решения ЗН основан на рекурсивном получении ее оптимального решения. Вычислительная схема рекуррентного метода решения ЗН представлена в форме, удобной для реализации на ЭВМ. Приведенная схема поиска оптимального назначения является основой алгоритма, в котором решение задачи находится исключительно средствами теории паросочетаний для двудольных графов, представленными в перестановочно-матричной форме.

Разработанный метод по сравнению с другими дает возможность сэкономить вычислительные ресурсы, в результате достигается выигрыш в скорости работы алгоритма на больших размерностях входных данных.

Для оценки времени решения ЗН предложенным методом проведен вычислительный эксперимент. Исследована зависимость времени решения ЗН различными методами от размерности входных данных. Сравнивались четыре метода решения ЗН: метод потенциалов, венгерский алгоритм, алгоритм Кана–Мункреса и рекуррентный метод. Наилучшие результаты показал рекуррентный метод, а наихудшие — метод потенциалов.

СПИСОК ЛИТЕРАТУРЫ

1. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация: Алгоритмы и сложность. — М.: Мир, 1985. — 510 с.
2. Левченко А.Ю., Морозов А.В., Панишев А.В. Быстрый алгоритм решения задачи о назначениях для нахождения нижней границы стоимости маршрута коммивояжера // Искусственный интеллект. — 2011. — Вып. 4. — С. 406–416.
3. Левченко А.Ю., Морозов А.В., Панишев А.В. Механизм ускорения вычислений в методе Литтла для решения задач класса коммивояжера // Искусственный интеллект. — 2012. — Вып. 2. — С. 95–110.

Поступила 20.02.2015