M. Sh. Levin, E. V. Levner, On an effective solution of the Bellman — Johnson problem on a tree-like network, *Avtomat. i Telemekh.*, 1978, Issue 10, 110–118

УДК 519.14

ОБ ЭФФЕКТИВНОМ РЕШЕНИИ ЗАДАЧИ БЕЛЛМАНА — ДЖОНСОНА НА СЕТИ, ИМЕЮЩЕЙ ВИД ДЕРЕВА

М. Ш. ЛЕВИН, Е. В. ЛЕВНЕР

(Москва)

Рассматривается задача Беллмана – Джонсона на сети в форме дерева (в случае двух машин). Опровергается существующее мнение, что для нее не существует алгоритма со степенной оценкой трудоемкости. Предлагается эффективный алгоритм ее решения с трудоемкостью $O(n \log n)$, где n – число работ.

Во многих практических задачах составления расписаний возникают ситуации, когда определенные последовательности выполнения работ оказываются недопустимыми по тем или иным соображениям. В частности, такие ситуации имеют место, когда работы связаны между собой отношениями предшествования: одни работы должны быть закончены прежде, чем начнутся некоторые другие. Обычно эти отношения задаются с помощью сети [1–5]. Одной из наиболее известных задач этого класса является задача Беллмана — Джонсона на сети. Она возникает, например, при составлении расписания выполнения программ в двухмашинном вычислительном комплексе, в котором каждая работа (программа) выполняется последовательно на обеих ЭВМ и программы информационно связаны: выходные результаты одних программ являются входными данными для других.

До последнего времени существовали предположения, что задача Беллмана — Джонсона (для двух машин) на сети, имеющей вид дерева, обязательно требует экспоненциального перебора вариантов [6]. В настоящей работе показано, что это мнение является ошибочным. Предлагается эффективный точный алгоритм ее решения с трудоемкостью $O(n \log n)$, где n — число работ.

1. Постановка задачи и основные определения

Имеются две машины и множество работ $R=\{1, 2, \ldots, n\}$, каждая из которых должна выполняться сначала на первой машине, а затем — на второй. Одновременно на каждой машине может выполняться только одна работа. Заданы $2n$ неотрицательных чисел a_i, b_i $(i=\overline{1,n})$, где $a_i(b_i)$ — продолжительность выполнения i-й работы на первой (на второй) машине. Работы связаны между собой отношениями предшествования, идентичными для обеих машин. Отношения предшествования задаются с помощью сети Г, представляющей собой совокупность нескольких ориентированных деревьев: вершины деревьев взаимно-однозначно соответствуют работам, а дуги, направленные от корней к висячим вершинам (или наоборот), указывают допустимый порядок выполнения работ.

Требуется установить оптимальный порядок s^* выполнения работ на машинах (расписание), который является допустимым относительно сети

Γ (т. е. удовлетворяет ограничениям предшествования, заданным сетью Γ) и при этом дает минимальное общее время выполнения всех работ: $f(s^*)=\min_s f(s)$.

Справедливо следующее утверждение: если отношения предшествования на обеих машинах одинаковы, то существует оптимальное расписание s^*, в котором на обеих машинах работы выполняются в одинаковом порядке. (Доказательство проводится так же, как и в случае, когда ограничения предшествования отсутствуют [7].) Далее будем рассматривать только такие расписания, т. е. расписание s — это перестановка n чисел 1, 2, ..., n: $s=\langle s[1], s[2], \ldots, s[n]\rangle$, где $s[r]$ — номер работы, находящейся на r-м месте (слева).

Рассмотрим следующую вещественную функцию от целочисленного аргумента (приоритеты):

$$(1) \qquad \omega(k)=\begin{cases} a_k-\max a_i, & \text{если} \quad a_k\leqslant b_k,\\ \max b_i-b_k, & \text{если} \quad a_k>b_k.\end{cases}$$

Для получения оптимального расписания в задаче Беллмана — Джонсона (с двумя машинами) без ограничений предшествования достаточно упорядочить числа 1, 2, ..., n в порядке неубывания приоритетов $\omega(k)$. (Эта процедура называется алгоритмом Джонсона [2, 7].)

Заметим, что приоритеты $\omega(k)$ могут быть вычислены следующим образом [2]:

$$(2) \qquad \omega(k)=\operatorname{sign}(a_k-b_k)\,[M-\min(a_k, b_k)],$$

где $M=\sum_{i\in R}(a_i+b_i)$.

Определение 1. Допустимые расписания s и s' называются сопряженными по работам i, j, если $s=\langle\ldots i, j\ldots\rangle$, $s'=\langle\ldots j, i\ldots\rangle$ и $s[r]=s'[r]$ при $s[r]\neq i, j$.

Функция $\omega(k)$ обладает следующим свойством:

$$(3) \qquad f(s)\leqslant f(s'), \text{ если } \omega(i)\leqslant\omega(j)$$

(доказательство проводится так же, как и в случае, когда ограничения предшествования отсутствуют, см. [2, 7]).

Определение 2. Работы $i, j\in R$ называются независимыми относительно сети Γ, если допустимость относительно сети Γ произвольного расписания не зависит от порядка следования элементов i и j, т. е. существует допустимое расписание, в котором i предшествует j, и допустимое расписание, в котором j предшествует i.

Будем рассматривать случай, когда сеть Γ, задающая отношение предшествования, представляет собой совокупность конечного числа ориентированных деревьев с ориентацией дуг от корней к висячим вершинам. (Предлагаемый ниже алгоритм может быть использован и в случае противоположной ориентации.)

Определение 3. Пусть X — множество вершин сети Γ и $x, y\in X$, а $y\in\Gamma x$. Тогда вершина y называется последователем x.

Определение 4. Будем говорить, что задача Беллмана — Джонсона на сети Γ приведена к нормальному виду, если для любых $x, y\in R$, таких, что $y\in\Gamma x$, выполняется условие $\omega(x)<\omega(y)$.

Определение 5. Поддерево T_x с корнем x в сети Γ будем называть нормальным, если:

1) в множестве вершин поддерева T_x кроме x входят все вершины из Γ, которым предшествует x;

2) для любых вершин y, z из T_x, таких, что $z\in\Gamma y$, выполняется условие $\omega(y)<\omega(z)$.

Пусть $N(i, j)$ — множество допустимых расписаний работ R, в которых две произвольные фиксированные работы $i, j \in R$ стоят рядом, и работа i предшествует работе j. Рассмотрим произвольное расписание $s \in N(i, j)$, в котором $i = s[l]$, $j = s[l+1]$ $(1 \leqslant l \leqslant n-1)$. Пусть имеется работа $J \in R$. Определим расписанием $\bar{s}$ множества работ $(R \setminus \{i, j\}) \cup J$ следующим образом:

$$(4) \qquad \bar{s}[r] = \begin{cases} s[r] & \text{при } r = \overline{1, l-1}, \\ J & \text{при } r = l, \\ s[r+1] & \text{при } r = \overline{l+1, n-1}. \end{cases}$$

Определение 6. Работу $J = J(i, j)$ назовем эквивалентной работам i, j, если выполняются следующие два условия:

1) параметры a_J и b_J работы J могут быть в явном виде выражены через параметры a_i, b_i, a_j, b_j работ i и j,

2) для любого расписания $s \in N(i, j)$ выполняется соотношение

$$(5) \qquad f(s) - f(\bar{s}) = c,$$

где c — константа, не зависящая от s.

Заметим, что величина $f(\bar{s})$ имеет смысл, так как целевая функция f может быть подсчитана на перестановках длины m, $m \leqslant n$.

Определение 7. Расписание $\bar{s}$, полученное из $s \in N(i, j)$ по формулам (4), называется эквивалентным расписанию s, если J — работа, эквивалентная работам i, j.

В случае существования эквивалентной работы для любых $i, j \in R$ будем говорить, что выполняется условие «склеивания» работ.

Обозначим через $\bar{\Gamma}$ сеть, полученную из Γ «склейкой» i и j (т. е. в $\bar{\Gamma}$ вместо вершин i и j входит вершина J и $\bar{\Gamma}J = \Gamma i \cup \Gamma j \setminus j$). Обозначим через $\bar{N}(i, j)$ множество расписаний работ $(R \setminus \{i, j\}) \cup J$, допустимых на сети $\bar{\Gamma}$. Другими словами, $\bar{N}(i, j)$ — это множество всех перестановок элементов из $(R \setminus \{i, j\}) \cup J$, которые удовлетворяют ограничениям предшествования, заданным сетью $\bar{\Gamma}$.

Очевидно, что $\bar{s} \in \bar{N}(i, j)$.

Отображение $N(i, j) \to \bar{N}(i, j)$, задаваемое формулами (4), (5), обозначим через Φ.

2. Свойства эквивалентных расписаний

Лемма. Для того чтобы расписание $s_0 \in N(i, j)$ было оптимальным на $N(i, j)$, т. е. для любого $s \in N(i, j)$ выполнялось условие

$$(6) \qquad f(s_0) \leqslant f(s),$$

необходимо и достаточно, чтобы эквивалентное расписание $\bar{s}_0 = \Phi(s_0) \in \bar{N}(i, j)$ было оптимально на $\bar{N}(i, j)$, т. е. для любого $\bar{s} \in \bar{N}(i, j)$ выполнялось $f(\bar{s}_0) \leqslant f(\bar{s})$.

Доказательство леммы приводится в приложении.

Пусть известно, что оптимальное расписание s_0 исходной задачи P_0 можно искать на множестве $N(i, j)$. Тогда в силу леммы достаточно найти решение $\bar{s}_0$ задачи P_1 на «склеенной» сети $\bar{\Gamma}$, а затем по $\bar{s}_0$ восстановить оптимальное расписание s_0: $s_0 = \Phi^{-1}(\bar{s}_0)$. Будем говорить, что задачи P_0 и P_1 эквивалентны. Далее, пусть известно, что оптимальное расписание в задаче P_1 можно искать на некотором множестве расписаний $N_1(k, t)$, в которых $k, t \in (R \setminus \{i, j\}) \cup J$, а k, t стоят рядом и $t \in \bar{\Gamma}k$. В силу леммы для нахождения $\bar{s}_0$ достаточно найти решение $\bar{s}_1$ задачи P_2, сформулированной на графе Γ_1, где Γ_1 получается из $\bar{\Gamma}$ «склейкой» вершин k и t.

Обозначим через J_1 работу, эквивалентную k и t, через $\bar{N}(k, t)$ — множество расписаний работ $(R \setminus \{i, j, k, t\}) \cup J \cup J_1$, допустимых на сети Γ_1. Отображение $\bar{N}(i, j) \to \bar{N}_1(k, t)$ обозначим через Φ_1. Тогда $\bar{s}_0 - \Phi_1^{-1}(\bar{s}_1)$ и $s_0 =$

$=\Phi^{-1}(\bar{s}_0)$. Будем говорить, что задача P_2 эквивалентна задачам P_1 и P_0. Процесс «склеивания» вершин и получения эквивалентных задач дальше может быть продолжен аналогично.

Перейдем к изучению свойств эквивалентных расписаний в задаче Беллмана — Джонсона. Оказывается, в случае двух машин параметры эквивалентной работы J могут быть определены следующим образом:

$$a_J=a_i+a_j-\min(a_j, b_i),\qquad(7)$$
$$b_J=b_i+b_j-\min(a_j, b_i).$$

Не изменяя значений целевой функции, для любого расписания работ можно сделать так, чтобы момент окончания выполнения работы k на машине m совпадал с моментом начала выполнения следующей работы на той же машине $(k=\overline{1, n-1})$. Для этого достаточно на диаграмме Ганта (см. [2, 7]) все работы на первой машине максимально «сдвинуть влево», а на второй машине — «сдвинуть вправо». Тогда легко видеть, что при выполнении (7) выполняется (5), причем $c=\min(a_j; b_i)$.

Аналогично можно ввести «склеивание» конечного числа работ. Пусть, например, в расписании s работы $q_1, q_2, \ldots, q_k$ стоят рядом и q_i предшествует q_{i+1} $(1\leqslant i\leqslant k-1)$. Тогда параметры эквивалентной работы можно определить следующим образом:

$$a_J=\sum_{j=1}^{k} a_{q_j}-\min\left(\sum_{j=2}^{k} a_{q_j}, \sum_{j=1}^{k-1} b_{q_j}\right),\qquad(8)$$
$$b_J=\sum_{j=1}^{k} b_{q_j}-\min\left(\sum_{j=2}^{k} a_{q_j}, \sum_{j=1}^{k-1} b_{q_j}\right),$$

и константа из формулы (5) имеет вид

$$c=\min\left(\sum_{j=2}^{k} a_{q_j}, \sum_{j=1}^{k-1} b_{q_j}\right).\qquad(9)$$

Следует заметить, что условие «склеивания» работ выполняется и в задачах, рассмотренных Хорном [4] и Гордоном, Танаевым [2, 3].

3. Приведение задачи к нормальному виду

Теорема. Пусть $x\in R$, а M — множество последователей x, причем для любого $y\in M$ поддерево с корнем y является нормальным. Пусть корень $y_0\in M$ такой, что для любого $y\in M$

$$\omega(y_0)\leqslant\omega(y),\qquad(10)$$
$$\omega(x)\geqslant\omega(y_0).\qquad(11)$$

Тогда от любого допустимого расписания работ множества R можно без ухудшения критерия перейти к некоторому допустимому расписанию, в котором работы x и y_0 находятся рядом.

Доказательство теоремы приводится в приложении.

Теорема позволяет любую исходную задачу с помощью «склеивания» работ приводить к эквивалентной задаче нормального вида. Рассмотрим этот процесс. Пусть задана задача Беллмана — Джонсона в случае двух машин на сети Г. Висячие вершины составляют множество M^0. Каждый элемент этого множества представляет собой нормальное поддерево. Рассмотрим M^1 — множество вершин, все последовательности которых принадлежат M^0. Пусть x — произвольный элемент множества M^1 и $M_x=$

$=\{i \in R \mid i \in \Gamma x\}$. Пусть элемент $i_0 \in M_x$, такой, что $\omega(i_0) \leqslant \omega(i)$ для всех $i \in M_x$. Если $\omega(x) < \omega(i_0)$, то поддерево с корнем x является нормальным. Если $\omega(x) \geqslant \omega(i_0)$, то по теореме в любом допустимом расписании, в том числе и в оптимальном, перестановками, не нарушающими допустимость и не ухудшающими значение критерия, можно перейти к расписанию, в котором работы x и i_0 находятся рядом. Таким образом, можно искать оптимальное расписание на множестве тех расписаний $N(x, i_0)$, в которых работы x и i_0 находятся рядом. По лемме можно перейти к эквивалентной задаче, т. е. искать оптимальное расписание на множестве эквивалентных расписаний $\bar{N}(x, i_0)$, в которых вместо работ x и i_0 рассматривается эквивалентная им работа J.

Проводя описанную выше процедуру для эквивалентной задачи и осуществляя в случае необходимости «склеивание» и переход к эквивалентным задачам, за конечное число шагов (множество R конечно), получим эквивалентную задачу, в которой все вершины множества M^1 являются корнями нормальных поддеревьев.

Затем перейдем к множеству вершин M^2, все последователи которых принадлежат множеству $M^0 \cup M^1$, и т. д.

Продолжая этот процесс, получим эквивалентную задачу нормального вида.

Алгоритм решения нормальной эквивалентной задачи очевиден. Он состоит в упорядочении работ в порядке неубывания приоритетов $\omega(k)$. Действительно, эта процедура совпадает с алгоритмом Джонсона и дает расписание $\bar{s}_0$ с наименьшим значением целевой функции, а так как задача нормального вида, то расписание $\bar{s}_0$ является допустимым.

После составления оптимального расписания эквивалентной нормальной задачи для получения оптимального расписания исходной задачи s_0 нужно заменить все эквивалентные работы на исходные.

Если ограничения предшествования в задаче Беллмана — Джонсона в случае двух машин заданы в виде деревьев с ориентацией дуг от висячих вершин к корням, т. е. с ориентацией, противоположной ориентации дуг в сети Г, все приведенные результаты сохраняют силу с точностью до симметрии, в частности «склеивание» работ x, y_0 проводится в том случае, если $\omega(y_0) \geqslant \omega(x)$.

Заметим, что при числе машин $m \geqslant 3$ задача Беллмана — Джонсона является универсальной переборной задачей [6, 8].

4. Алгоритм

I. Приведение исходной задачи к эквивалентной задаче нормального вида.

0-й шаг. Выделение всех висячих вершин (множество M^0).

1-й шаг. Выделение множества M^1 (множество вершин, все последователи которых принадлежат множеству M^0). Для каждой вершины $k \in M^1$ сравнение $\omega(k)$ и $\omega(i_0)$, где i_0 такова, что $\omega(i_0) \leqslant \omega(i)$ для всех $i \in \Gamma k$. Если $\omega(k) \geqslant \omega(i_0)$, то «склеивание» работ k и i_0 по формулам (7) и переход к эквивалентной задаче. (Для эквивалентной задачи проводится аналогичное сравнение эквивалентной работы J с ее последователями).

k-й шаг. Выделение множества M^k (множество вершин, все последователи которых принадлежат множеству $\bigcup_{j=0}^{k-1} M^j$). Для каждой вершины $k \in M^k$ сравнение $\omega(k)$ и $\omega(i_0)$, где i_0 такова, что $\omega(i_0) \leqslant \omega(i)$ для всех $i \in \Gamma k$. Если $\omega(k) \geqslant \omega(i_0)$, то «склеивание» работ k и i_0 и переход к эквивалентной задаче. (Для эквивалентной задачи проводится аналогичное сравнение эквивалентной работы J с ее последователями.)

II. Решение эквивалентной нормальной задачи (упорядочение работ по неубыванию приоритетов $\omega(k)$).

III. Замена эквивалентных работ в решении эквивалентной нормальной задачи на работы исходного множества R.

Описанный алгоритм аналогичен алгоритмам Хорна [4] и Гордона — Танаева [3], отличаясь от них видом минимизируемой целевой функции и функции приоритетов $\omega(k)$. Для алгоритма Хорна известна оценка трудоемкости алгоритма — $O(n \log n)$, полученная в работе [9]. Эта же оценка справедлива и для рассмотренного здесь алгоритма.

Существование приоритета и выполнение условия «склеивания» позволяют построить эффективный алгоритм и для более общего, чем совокупность деревьев, ограничения предшествования, а именно параллельно-последовательной сети.

5. Пример

Пусть имеется множество работ $R=\{1, 2, \ldots, 10\}$, параметры и приоритеты $\omega(k)$ которых представлены в таблице. Пусть ограничения предшествования заданы в виде сети Г, представленной на рис. 1.

Номер работы k	Время выполнения работы		Приоритет $\omega(k)$	Место в оптимальном расписании
	a_k	b_k		
1	6	13	−14	4
2	3	13	−17	2
3	4	16	−16	3
4	20	8	12	8
5	7	18	−13	5
6	14	5	15	9
7	19	1	19	10
8	12	9	11	7
9	2	20	−18	1
10	9	15	−11	6

I. Приведение исходной задачи к эквивалентной задаче нормального вида.

0-й шаг. Выделяем множество висячих вершин:

$$M^0=\{1, 3, 6, 8, 9, 10\}.$$

1-й шаг. Выделяем множество вершин M^1, все последователи которых принадлежат множеству M^0: $M^1=\{5, 7\}$.

Сравниваем работу 5 с ее последователями: $M_5=\{8, 9\}$, $\omega(9)<\omega(8)$, $\omega(5)>\omega(9)$. Требуется рассчитать параметры эквивалентной работы $J=J(5, 9)$ по формулам (7) и перейти к эквивалентной задаче: $a_{J(5,9)}=7+2-2=7$, $b_{J(5,9)}=18+20-2=36$, $\omega(J(5,9))=-13$, $R=\{1, 2, 3, 4, J(5,9), 6, 7, 8, 10\}$.

Сравниваем работу $J(5,9)$ с ее последователями:

$$M_{J(5,9)}=\{8\}, \quad \omega(J(5,9))<\omega(8).$$

Сравниваем работу 7 с ее последователями: $M_7=\{10\}$, $\omega(7)>\omega(10)$. Требуется рассчитать параметры эквивалентной работы $J=J(7, 10)$ и перейти к эквивалентной задаче:

$$a_{J(7,10)}=19+9-1=27, \quad b_{J(7,10)}=1+15-1=15, \quad \omega(J(7,10))=5,$$
$$R=\{1, 2, 3, 4, J(5, 9), 6, J(7, 10), 8\}.$$

Новая сеть предшествования представлена на рис. 2.

2-й шаг. Выделяем множество вершин M^2, все последователи которых принадлежат множеству $(M^0 \cup M^1)$: $M^2=\{4\}$. Сравниваем работу 4 с ее последователями: $M_4=\{J(5,9), 6, J(7,10)\}$, $\omega(J(5,9))<\omega(J(7,10))$, $\omega(J(5,9))<\omega(6)$, $\omega(4)>\omega(J(5,9))$. Рассчитываем параметры эквивалентной работы $J=J(4, J(5, 9))$ и переходим к эквивалентной задаче:

$$a_{J(4,J(5,9))}=20+7-7=20, \quad b_{J(4,J(5,9))}=8+36-7=37, \quad \omega(J(4, J(5,9)))=0.$$

Сравниваем работу $J(4, J(5, 9))$ с ее последователями:

$$M_{J(4,J(5,9))}=\{8, 6, J(7,10)\}, \quad \omega(J(7,10))<\omega(8),$$
$$\omega(J(7,10))<\omega(6), \quad \omega(J(4, J(5,9)))<\omega(J(7,10)).$$

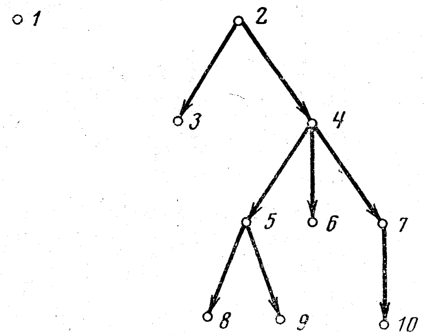

Рис. 1

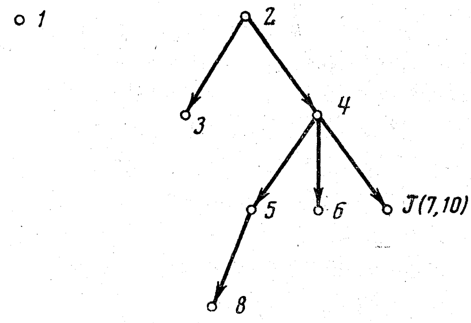

Рис. 2

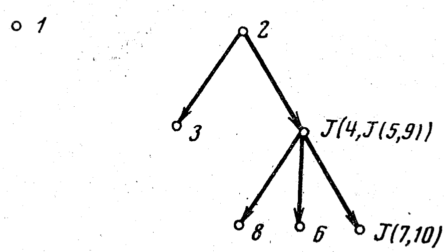

Рис. 3

Новое множество $R=\{1,\ 2,\ 3,\ J(5,9),\ 6,\ 8,\ J(7,10)\}$. Сеть предшествований представлена на рис. 3.

3-й шаг. Выделяем множество вершин M^3, все последователи которых принадлежат множеству $\bigcup_{j=0}^{2} M^j$: $M^3=2$. Сравниваем работу 2 с ее последователями:

$$M_2=\{3,\ J(4,\ J(5,9))\},\qquad \omega(3)<\omega(J(4,\ J(5,9))),\qquad \omega(2)<\omega(3).$$

Таким образом, получена эквивалентная нормальная задача на множестве работ $R=\{1,\ 2,\ 3,\ J(4,\ J(5,9)),\ 6,\ J(7,10),\ 8\}$ с ограничениями предшествования, заданными сетью, представленной на рис. 3.

II. Построение оптимального расписания $\bar{s}_0$ для эквивалентной нормальной задачи:

$$\bar{s}_0=\langle 2,\ 3,\ 1,\ J(4,\ J(5,9)),\ J(7,10),\ 8,\ 6\rangle.$$

III. Замена эквивалентных работ в решении эквивалентной нормальной задачи на работы исходного множества R. После замены эквивалентных работ $J(4,\ J(5,9))$ и $J(7,\ 10)$ на работы исходного множества получаем расписание $s_0=\langle 2,\ 3,\ 1,\ 4,\ 5,\ 9,\ 7,\ 10,\ 8,\ 6\rangle$ с $f(s_0)=121$.

ПРИЛОЖЕНИЕ

Доказательство леммы. Пусть s_0 оптимально, т. е. выполняется (6), а $\bar{s}_0=\Phi(s_0)$ не оптимально. Тогда существует $\bar{s}_1\in\bar{N}(i,\ j)$, такое, что

(П.1) $\qquad f(\bar{s}_1)<f(\bar{s}_0).$

В этом случае по (5) имеем

(П.2) $\qquad f(s_0)=f(\bar{s}_0)+c,\qquad f(s_1)=f(\bar{s}_1)+c,$

где $s_1=\Phi^{-1}(\bar{s}_1)$.

Очевидно, что такой обратный переход возможен. Из (П.1) и (П.2) получаем $f(s_1)<f(s_0)$, что противоречит предположению об оптимальности s_0. Доказательство в обратную сторону аналогично.

Доказательство теоремы. Пусть имеется произвольное допустимое расписание s элементов множества R, в котором между работами x и y_0 находятся элементы некоторого множества $A\subset R$. Если $A=\phi$, то это расписание является искомым. Пусть $A\neq\phi$. Очевидно, что любой элемент множества A образует с элементом y_0 независимую пару элементов, так как рассматривается допустимое расписание, и сеть Γ, задающая ограничения предшествования, является совокупностью деревьев, причем по условию теоремы $y\in\Gamma x$. Следовательно, возможны два случая:

1) когда любой элемент множества A образует с элементом x независимую пару элементов;

2) среди элементов множества A есть элементы, которые не могут стоять в допустимом расписании перед элементом x.

Рассмотрим первый случай. Как было показано, в задаче Беллмана – Джонсона (в случае двух машин) выполняется условие «склеивания» работ. Найдем по формуле (8) параметры работы J, эквивалентной работам множества A, и перейдем к эквивалентному расписанию $\bar{s}$, причем

$$f(s)-f(\bar{s})=c, \tag{П.3}$$

где c определяется формулой (9).

Подсчитаем по формуле (1) приоритет $\omega(J)$ эквивалентной работы J. Возможны два варианта:

а) если $\omega(J)\geqslant\omega(y_0)$, то поменяем эти работы местами, причем значение критерия вследствие (3) не ухудшится,

б) если $\omega(J)<\omega(y_0)$, то по (13) $\omega(J)<\omega(x)$. Тогда поменяем работы J и x местами, причем значение критерия вследствие (3) не ухудшится.

В обоих вариантах допустимость не нарушалась. В результате имеем допустимое эквивалентное расписание $\bar{s}_1$, причем

$$f(\bar{s}_1)\leqslant f(\bar{s}). \tag{П.4}$$

Заменив эквивалентную работу J в расписании $\bar{s}_1$ на исходные работы множества A, получим допустимое расписание s_1 работ множества R, в котором работы x и y_0 находятся рядом, причем, так как $f(s_1)-f(\bar{s}_1)=c$, то с учетом (П.3) и (П.4) имеем $f(s_1)\leqslant f(s)$.

Рассмотрим второй случай. Если какой-то элемент $z\in A$ не может стоять в допустимом расписании перед элементом x, то элемент z входит в поддерево с корнем x. Тогда, учитывая (10) и то, что все поддеревья поддерева с корнем x являются нормальными, имеем

$$\omega(z)\geqslant\omega(y_0). \tag{П.5}$$

Обозначим множество всех таких элементов через B, $B\leqslant A$. Рассмотрим элемент $z_0\in B$, такой, что $s[z_0]\geqslant s[z]$ для всех $z\in B$. В расписании s между элементами z_0 и y_0 могут находиться работы некоторого множества $C\subset A$. Если $C=\phi$, то поменяем элементы z_0 и y_0 местами. Вследствие (3) и (П.5) получим допустимое расписание без ухудшения значения критерия. Пусть $C\neq\phi$. Очевидно, что по отношению к элементам z_0, y_0 и множеству C имеет место ситуация, аналогичная только что рассмотренному первому случаю, так как любой элемент множества образует с z_0 независимую пару, т. е. можно без ухудшения критерия перейти к допустимому расписанию, в котором элементы z_0 и y_0 находятся рядом. Затем можно поменять эти элементы местами без ухудшения значения критерия в силу (П.5).

Поскольку мощность множества B уменьшена на 1, а множество A конечно, то, продолжая описанный процесс, за конечное число шагов придем к допустимому расписанию, в котором работы x и y_0 находятся рядом и значение критерия не больше чем $f(s)$.

Поступила в редакцию
19 декабря 1977 г.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. *Бурков В. Н., Ловецкий С. Е.* Методы решения экстремальных задач комбинаторного типа (обзор). Автоматика и телемеханика, № 11, стр. 68–93, 1968.
2. *Танаев В. С., Шкурба В. В.* Введение в теорию расписаний. «Наука», 1975.
3. *Гордон В. С., Танаев В. С.* Детерминированные системы обслуживания с одним прибором, древовидным упорядочением требований и экспоненциальными функциями штрафа. В сб. «Вычислительная техника в машиностроении», стр. 3–10. Изд-во Ин-та технической кибернетики АН БССР, Минск, 1973.
4. *Horn W. A.* Single-Machine Job Sequencing with Treelike Precedence Ordering and Linear Delay Penalties. SIAM J. Appl. Math., v. 23, No. 2, pp. 189–202, 1972.
5. *Ullman J. D.* NP-Complete Scheduling Problems. J. Computer and System Sci., v. 10, pp. 384–393, 1973.
6. *Brucker P., Lenstra J. K., Rinooy Kan A. H. G.* Complexity of Machine Scheduling Problems. Mathematisch Centrum Afdeling Mathematische Besliskunde BV, No. 43, 1975.
7. *Джонсон С.* Оптимальное расписание для двух- и трехступенчатых процессов. Кибернетический сб., Новая серия, вып. 1, стр. 78–86. «Мир», 1965.
8. *Лившиц Э. М., Рублинецкий В. И.* О сравнительной сложности некоторых задач дискретной оптимизации. В сб. «Вычислительная математика и вычислительная

техника», вып. 3, стр. 78–85. Изд-во Физ.-техн. ин-та низких температур АН УССР, Харьков, 1975.
9. *Adolphson D., Hu T. C.* Optimal Linear Ordering. SIAM J. Appl. Math., v. 25, No. 3, pp. 403–423, 1973.

ON AN EFFECTIVE SOLUTION OF THE BELLMAN — JOHNSON PROBLEM ON A TREE-LIKE NETWORK

M. Sh. LEVIN, Ye. V. LEVNER

The paper is concerned with the Bellman — Johnson problem on a tree — like network (in the case of two machines). The view is refuted that no algorithm with exponential labor consumption estimate exists for it. An effective solution algorithm is proposed for a labor consumption of $O(n \log n)$ where n is the number of activities.