

19180

Кибернетика

ОТДЕЛЬНЫЙ ОТТИСК

ОБ ЭФФЕКТИВНОМ РЕШЕНИИ НЕКОТОРЫХ ЗАДАЧ ТЕОРИИ РАСПИСАНИЙ НА СЕТЯХ

УДК 519.14

В последнее время большое внимание уделяется задачам теории расписаний с ограничением предшествования, т. е., когда одни работы могут выполняться только после выполнения некоторых других. В общем случае ограничение предшествования задается направленным графом, не содержащим контуров, в котором вершины соответствуют упорядочиваемым работам, а направленные дуги задают предшествование. Ограничение предшествования значительно усложняет задачи теории расписаний и для многих таких задач доказано, что они относятся к классу универсальных переборных задач [7].

В случае, когда ограничение предшествования задается графом простого вида, для некоторых задач существуют эффективные алгоритмы. Например, в [2] решена задача при ограничении предшествования, заданном в виде параллельных цепочек. При древовидном ограничении предшествования найдены эффективные алгоритмы для двух задач [3], [4]. Трудоемкость этих алгоритмов будет $O(n \log n)$, где n — число работ [5]. Попытка решить задачу из [3] при ограничениях предшествования общего вида принята в [6], но предлагаемый алгоритм не является эффективным.

В настоящей работе предлагается эффективный подход с трудоемкостью $O(n^2)$ к решению шести задач теории расписаний на сетях, сводящихся к параллельно-последовательным. Они обобщают сети, рассмотренные в работах [2]—[4].

1. ПОСТАНОВКА ЗАДАЧИ И ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

В рассматриваемых задачах имеется множество работ $R = \{1, 2, \dots, n\}$ и ограничение предшествования, заданное направленным графом $G = (R, \Gamma)$. Требуется установить оптимальный допустимый порядок s^* выполнения работ (расписание), удовлетворяющий заданным ограничениям предшествования, на котором значение функционала минимально: $f(s^*) = \min f(s)$.

Задача 1. Имеется одна машина. Каждая работа $i \in R$ характеризуется временем выполнения τ_i и функцией штрафа $\varphi_i(t) = a_i t + b_i$. Требуется минимизировать суммарный штраф

$$f(s) = \sum_{i=1}^n \varphi_i(C_i), \text{ где } C_i \text{ — момент окончания}$$

выполнения работы i (все работы поступают в момент времени $t = 0$).

Задача 2. Она отличается от задачи 1 тем, что функция штрафа имеет вид $\varphi_i(t) = a_i \exp(\lambda t) + b_i$.

Задача 3. Это задача Беллмана — Джонсона в случае двух машин. Каждая работа $i \in R$ должна выполняться сначала на первой машине, затем на второй со временем выполнения соответственно a_i, b_i . Требуется минимизировать суммарное время выполнения всех работ.

Задача 4. Имеется одна машина. Каждая работа $i \in R$ характеризуется временем выполнения τ_i и функцией штрафа $\varphi_i(t) = a_i t + b_i$. Требуется минимизировать суммарный штраф

$$f(s) = \sum_{i=1}^n \varphi_i(t_i), \text{ где } t_i = \sum_{l=1}^i \tau_{s[l]}, s[l] \text{ — номер работы, находящейся в расписании } s \text{ на } l\text{-м месте.}$$

Задача 5. Эта задача отличается от задачи 4 тем, что функция штрафа имеет вид $\varphi_i(t) = a_i \ln \lambda t + b_i$.

Задача 6. Имеется одна машина. Каждая работа $i \in R$ характеризуется стоимостью выполнения ρ_i и вероятностью срыва выполнения работы p_i . Требуется минимизировать среднюю стоимость выполнения всех работ $f(s) =$

$$= \sum_{i=1}^n \frac{\rho_{s[i]}}{D_{s[i]}}, \text{ где } D_{s[i]} = \prod_{k=1}^{s[i]} (1 - p_k).$$

Задача 6 при замене переменных сводится к задаче 1 [1], но ее можно рассматривать и независимо.

Для задачи 3 просто доказывается, как и в [10], что оптимальное расписание можно искать на множестве одмаршрутных расписаний. Таким образом, расписание s — это перестановка n чисел $1, 2, \dots, n$; $s = \langle s[1], \dots, s[l], \dots, s[n] \rangle$.

Указанные задачи, несмотря на различие в их постановке, обладают общими свойствами. В случае отсутствия ограничений предшествования они могут быть решены упорядочением работ по неубыванию вещественных функций — приоритетов $\omega(i)$, вычисляемых для каждой работы $i \in R$. Для рассматриваемых задач [9], [1], [13], [8] приоритеты имеют такой вид:

$$\omega(i) = \frac{\tau_i}{a_i};$$

$$\omega(i) = a_i \exp(\lambda \tau_i) [1 - \exp(\lambda \tau_i)]^{-1};$$

$$\omega(i) = \text{sign}(a_i - b_i) [M - \min(a_i, b_i)],$$

$$M = \sum_{k \in R} (a_k + b_k);$$

$$\omega(i) = \frac{a_i \tau_i}{1 - \tau_i};$$

$$\omega(i) = \frac{a_i \ln \lambda \tau_i}{1 - \ln \lambda \tau_i};$$

$$\omega(i) = \frac{\rho_i}{p_i}.$$

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

Приоритеты обладают следующим свойством [1] [10]. Если $\omega(i) \leq \omega(j)$, то

$$f(s) \leq f(s'). \quad (1)$$

Другим общим свойством рассматриваемых задач является то, что в них выполняется условие «склеивания» работ, введенное автором и Е. В. Левнером, которое заключается в следующем.

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 $i = s[l]$, $j = s[l+1]$ ($1 \leq l < n$). Пусть имеется работа $J \in R$. Определим расписание $\bar{s}$ множества работ $(R \setminus \{i, j\}) \cup J$ следующим образом:

$$\bar{s}[r] = \begin{cases} s[r], & r = \overline{1, l-1}; \\ J, & r = l; \\ s[r+1], & r = \overline{l+1, n-1}. \end{cases} \quad (2)$$

Определение 2. Работу $J = J(i, j)$ назовем эквивалентной работам i, j , если выполняются следующие два условия:

- 1) параметры работы J в явном виде могут быть выражены через параметры работ i и j ;
- 2) для любого расписания $s \in N(i, j)$ выполняется соотношение

$$f(s) = f(\bar{s}) + c_{ij}, \quad (3)$$

где c_{ij} — константа, не зависящая от s .

Определение 3. Расписание $\bar{s}$, полученное из $s \in N(i, j)$ по формулам (2), называется эквивалентным расписанию s , если J — работа, эквивалентная работам i, j .

Параметры эквивалентной работы $J(i, j)$ и константа c_{ij} выражаются через параметры работ i, j так:

для задачи 1

$$a_J = a_i + a_j, \quad \tau_J = \tau_i + \tau_j,$$

$$b_J = b_i + b_j - a_i \tau_j, \quad c_{ij} = 0;$$

для задачи 2

$$a_J = a_i \exp(-\lambda \tau_j) + a_j, \quad \tau_J = \tau_i + \tau_j,$$

$$b_J = b_i + b_j, \quad c_{ij} = 0;$$

для задачи 3

$$a_J = a_i + a_j - \min(a_j, b_i), \quad b_J = b_i + b_j -$$

$$- \min(a_j, b_i), \quad c_{ij} = \min(a_j, b_i);$$

для задачи 4

$$a_J = \frac{a_i}{\tau_i} + a_j, \quad \tau_J = \tau_i \tau_j, \quad b_J = b_i + b_j, \quad c_{ij} = 0;$$

для задачи 5

$$a_J = a_i + a_j, \quad \tau_J = \tau_i \tau_j, \quad b_J = b_i + b_j - a_i \ln \tau_j,$$

$$c_{ij} = 0;$$

для задачи 6

$$\rho_J = \rho_i (1 - \rho_j) + \rho_j,$$

$$\rho_J = 1 - (1 - \rho_i)(1 - \rho_j), \quad c_{ij} = 0.$$

Пусть сеть Γ задает ограничение предшествования. Обозначим через $\bar{\Gamma}$ сеть, полученную из Γ «склеивкой» i и j (т. е. вместо работ i и j в $\bar{\Gamma}$ входит эквивалентная работа J и $\bar{\Gamma} = (\Gamma \setminus \{i, j\}) \cup J$). Обозначим через $\bar{N}(i, j)$ множество расписаний работ $(R \setminus \{i, j\}) \cup J$, допустимых на сети $\bar{\Gamma}$, т. е. $\bar{N}(i, j)$ — 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 $N(i, j) \rightarrow \bar{N}(i, j)$ по формуле (2) обозначим через Φ . Очевидна следующая лемма.

Лемма 1. 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 $f(s_0) \leq f(s)$, необходимо и достаточно, чтобы эквивалентное расписание $\bar{s}_0 = \Phi(s_0) \in \bar{N}(i, j)$ было оптимальным на $\bar{N}(i, j)$, т. е. для любого $\bar{s} \in \bar{N}(i, j)$ выполнялось $f(\bar{s}_0) \leq f(\bar{s})$.

Если известно, что оптимальное расписание s_0 исходной задачи P_0 можно искать на множестве $N(i, j)$, то по лемме 1 достаточно найти решение $\bar{s}_0$ задачи P_0 на «склеенной» сети $\bar{\Gamma}$, а затем по $\bar{s}_0$ восстановить оптимальное расписание $s_0 = \Phi^{-1}(\bar{s}_0)$. Будем говорить, что задачи P_0 и $\bar{P}_0$ — эквивалентны.

Аналогично можно ввести склеивание конечного числа работ. Пусть работы множества $Q \subset R$ ($|Q|=k$) в исходном расписании s стоят рядом, начиная с l -го места слева. Тогда эквивалентное расписание $\bar{s}$ определяется формулой

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

При этом

$$f(s) = f(\bar{s}) + c_Q. \quad (5)$$

Вводя две фиктивные работы, всегда можно перейти от общего случая ограничения существования к двухполюсной сети [11]. Далее, рассмотрим двухполюсные сети, в которых каждая вершина достижима из входного полюса, а выходной полюс достижим из каждой вершины.

Определение 4. Сеть, полученная из однореберных сетей в результате конечного числа параллельных и последовательных соединений, называется параллельно-последовательной (П-сеть) [11].

Определение 5. Работы $i, j \in R$ называются независимыми, если допустимость произвольного расписания не зависит от порядка следования элементов i и j , т. е. существуют допустимые расписания, в которых i предшествует j и j предшествует i .

Определение 6. Будем говорить, что задача имеет нормальный вид, если для любых $i, j \in R$, таких, что $j \in \Gamma i$, выполняется $\omega(i) < \omega(j)$.

Очевидно, что если задача имеет нормальный вид, то для получения оптимального расписания достаточно упорядочить множество работ в порядке неубывания их приоритетов.

Определение 7. Множество $X \subseteq R$ назовем модулем, если для любого $i \in X$ выполняется одно из трех условий:

- 1) любая работа $j \in X$ образует с работой i независимую пару;
- 2) работа i должна предшествовать любой работе $j \in X$;
- 3) любая работа $j \in X$ должна предшествовать работе i .

Определение 8. Будем говорить, что модуль X имеет нормальный вид, если для любых работ $x_1, x_2 \in X$ таких, что $x_2 \in \Gamma x_1$ выполняется $\omega(x_1) < \omega(x_2)$.

2. ЭЛЕМЕНТАРНЫЕ МОДУЛИ И ПРИВЕДЕНИЕ ИХ К НОРМАЛЬНОМУ ВИДУ

Последовательный модуль (рис. 1) составляют работы множества $\{r_i\}$, $i = \overline{1, k}$.

Лемма 2. Любой последовательный модуль за счет склеивания работ и перехода к эквива-

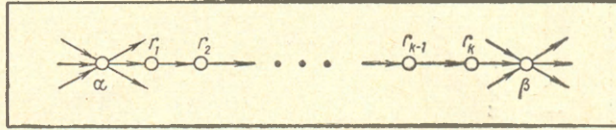


Рис. 1

лентным задачам можно привести к нормальному виду с трудоемкостью $O(k)$, где k — число работ модуля.

Доказательство. Рассмотрим последовательный модуль. Если $\omega(r_i) < \omega(r_{i+1})$ для любого i ($i = \overline{1, k-1}$), то последовательный модуль имеет нормальный вид. Пусть это не так, т. е. существует r_i такой, что

$$\omega(r_i) \geq \omega(r_{i+1}). \quad (6)$$

Пусть имеется некоторое допустимое расписание s работ множества R . Между работами r_i и r_{i+1} в расписании s могут находиться элементы некоторого множества $A \subseteq R \setminus \{r_i\}$. Работы множества A можно склеить и перейти к эквивалентному расписанию $\bar{s}$ по формуле (4), где вместо работ множества A имеется эквивалентная работа $J(A)$. Возможны два случая: 1) $\omega(r_i) > \omega(J(A))$; 2) $\omega(r_i) \leq \omega(J(A))$.

В случае работы r_i и $J(A)$ можно поменять местами и перейти к расписанию s_1 . Тогда, согласно (1), $f(s_1) \leq f(s)$. Если заменить в s_1 работу $J(A)$ на исходные работы множества A , получим расписание s_1 . При этом $f(s_1) \leq f(s)$.

Второй случай аналогичен первому. При $A = \emptyset$ все очевидно. Отметим, что в силу допустимости s и определения 7 элементы множества A образуют независимые пары с r_i, r_{i+1} .

Таким образом, от исходного допустимого расписания, не нарушая допустимости и не ухудшая значения критерия, всегда можно перейти к некоторому расписанию, в котором r_i и r_{i+1} находятся рядом, т. е. оптимальное допустимое расписание работ множества R можно искать на множестве $N(r_i, r_{i+1})$. По лемме 1 можно перейти к множеству $\bar{N}(i, j)$. Так как $|\{r_i\}| = k$ — конечно, то за конечное число проверок $\omega(r_i) \geq \omega(r_{i+1})$ и, если требуется, за конечное число склеиваний можно перейти к эквивалентной задаче, где исходный модуль имеет нормальный вид. Трудоемкость этого процесса будет $O(k)$.

Параллельный модуль (рис. 2) составляют два последовательных модуля, состоящих из работ множеств $\{r_i\} = \{r_1, r_2, \dots, r_k\}$ и $\{q_j\} = \{q_1, q_2, \dots, q_l\}$. Обозначим число работ параллельного модуля $m = k + l$.

Лемма 3. Любой параллельный модуль за счет склеивания работ и перехода к эквивалентным задачам можно свести к последовательному модулю нормального вида с трудоемкостью $O(m)$, где m — число работ модуля.

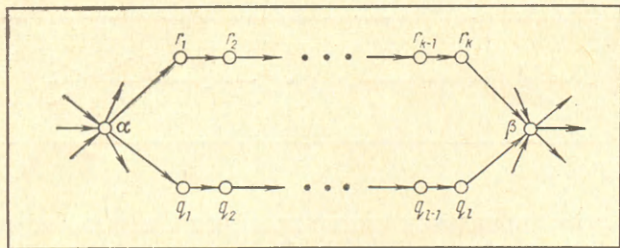


Рис. 2

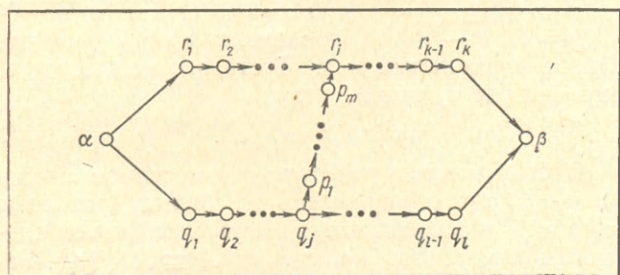


Рис. 3

Доказательство. По лемме 2 за $O(k) + O(l) = O(m)$ действий последовательные модули, составляющие параллельный модуль, можно привести к нормальному виду. Таким образом, можно считать

$$\omega(r_i) < \omega(r_{i+1}), \quad 1 \leq i < k; \quad (7)$$

$$\omega(q_j) < \omega(q_{j+1}), \quad 1 \leq j < l. \quad (8)$$

Рассмотрим произвольное допустимое расписание s работ множества R . Если для любых $x, y \in \{r_i\} \cup \{q_j\}$ таких, что работа x будет в s левее y , выполняется $\omega(x) < \omega(y)$, то модуль имеет нормальный вид. Пусть это не так, т. е. существуют $r_\lambda \in \{r_i\}$, $q_\nu \in \{q_j\}$ такие, что r_λ будет в s правее q_ν и

$$\omega(r_\lambda) < \omega(q_\nu). \quad (9)$$

Покажем, что в этом случае, не ухудшая значений критерия, можно перейти к допустимому расписанию, где работа r_λ предшествует работе q_ν . Пусть в расписании s между работами r_λ и q_ν находятся работы множества A . Выделим пять возможных случаев:

- 1) $A = \emptyset$;
- 2) $A \neq \emptyset$, $A \cap \{r_i\} = \emptyset$, $A \cap \{q_j\} = \emptyset$;
- 3) $A \cap \{r_i\} = \emptyset$, $A \cap \{q_j\} \neq \emptyset$;
- 4) $A \cap \{r_i\} \neq \emptyset$, $A \cap \{q_j\} = \emptyset$;
- 5) $A \cap \{r_i\} \neq \emptyset$, $A \cap \{q_j\} \neq \emptyset$.

Случаи 1 и 2 очевидны (см. доказательство леммы 2). В случае 3 множество A включает в себя

работы множества $Q = \{q_{\nu+1}, q_{\nu+2}, \dots, q_{\nu+\mu}\}$. В силу (8) и (9) имеем $\omega(r_\lambda) < \omega(z)$ для любого $z \in Q$.

Рассмотрим последовательно пары элементов r_λ и $q_{\nu+\eta}$, где η изменяется от μ до 0. Для каждой такой пары имеет место один из рассмотренных случаев. Таким образом, учитывая, что множество A конечно, за конечное число шагов, не ухудшая значения критерия, можно перейти к допустимому расписанию, где работа r_λ предшествует работе q_ν . Случай 4 симметричен случаю 3. Случай 5 сводится к одному из рассмотренных.

Пусть $A \cap \{r_i\} = \{x_1, x_2, \dots, x_\mu\}$, при этом x_η в s будет левее $x_{\eta+1}$, $1 \leq \eta < \mu$. Согласно (8), $\omega(x_\eta) < \omega(x_{\eta+1})$, а, согласно (9), $\omega(x_\eta) < \omega(q_\nu)$, $1 \leq \eta \leq \mu$. Обозначим элемент r_λ через $x_{\mu+1}$. Рассмотрим последовательно элементы x_η , где η изменяется от 0 до $\mu+1$. Для работ q_ν и x_η имеет место один из первых трех случаев. Так как множество A — конечно, то за конечное число шагов, не ухудшая значение критерия, можно перейти к допустимому расписанию, где работа r_λ предшествует работе q_ν .

Таким образом, оптимальное допустимое расписание работ множества R можно искать на множестве расписаний, где элементы множеств $\{r_i\}$ и $\{q_j\}$ упорядочены по неубыванию приоритетов, т. е. параллельный модуль можно заменить последовательным, в котором работы упорядочены по неубыванию приоритетов. Слияние двух упорядоченных последовательностей с количеством чисел k и l требует $O(k+l)$ действий [12]. Поскольку могут существовать x, y , такие, что $x \in \{r_i\}$, $y \in \{q_j\}$ и $\omega(x) = \omega(y)$, получившийся последовательный модуль следует привести к нормальному виду (по лемме 2 это требует $O(m)$ действий).

3. АЛГОРИТМ ДЛЯ П-СЕТИ

На основе доказанных выше лемм строится алгоритм решения рассматриваемых в работе задач.

1-й шаг. Приведение к нормальному виду всех последовательных модулей за счет склеивания работ и перехода к эквивалентным задачам.

2-й шаг. Если вся сеть приведена к последовательному модулю нормального вида, то переход к шагу 4.

3-й шаг. Проведение всех возможных слияний последовательных модулей нормального вида в параллельных модулях. Переход к шагу 1.

4-й шаг. Замена эквивалентных работ на работы исходного множества.

Трудоёмкость процесса нормализации последовательных модулей будет $O(n)$, так как если имеются последовательные модули с числом

вершин $k_1, k_2, \dots, k_l$, то $O(k_1) + O(k_2) + \dots + O(k_l) \leq O(n)$ в силу того, что $k_1 + k_2 + \dots + k_l \leq n$. Аналогично трудоемкость процесса слияний на шаге 3 будет $O(n)$. Каждое слияние последовательных модулей нормального вида в параллельных модулях уменьшает число последовательных модулей на единицу. Так как число последовательных модулей ограничено числом вершин n , то возможно не более n слияний, т. е. число циклов в алгоритме не превышает n и трудоемкость каждого из них — $O(n)$. Таким образом, трудоемкость алгоритма будет не более $O(n^2)$, где n — число работ исходного множества R .

Предложенный алгоритм позволяет эффективно решать задачи в тех случаях, когда ограничение задано более сложным образом, чем сеть, сводящаяся к П-сети, например, на некоторых сетях, содержащих Н-сети¹.

Пусть задано множество работ $R = \{r_\lambda\} \cup \{q_\eta\} \cup \{p_\mu\}$, где $\{r_\lambda\} = \{r_1, r_2, \dots, r_k\}$, $\{q_\eta\} = \{q_1, q_2, \dots, q_l\}$, $\{p_\mu\} = \{p_1, p_2, \dots, p_m\}$ и имеет место ограничение предшествования в виде, представленном на рис. 3. Введением двух фиктивных вершин — α и β — сведем исходную сеть к двухполюсной сети, представляющей собой Н-сеть. Очевидно, что в оптимальном допустимом расписании работа q_j непосредственно предшествует некоторой работе r_γ ($1 \leq \gamma \leq i$). Решив задачу для всех возможных значений γ (на i различных П-сетях) и выбрав решение с минимальным значением критерия, получим оптимальное допустимое расписание исходной задачи с трудоемкостью $O(k + l + m)^2 \leq O(n^3)$.

¹ Сеть, которую нельзя представить в виде последовательного или параллельного соединения сетей, называется Н-сетью [11].

Автор благодарен Е. В. Левнеру за постоянное внимание к работе.

ЛИТЕРАТУРА

1. Танаев В. С., Шкурба В. В. Введение в теорию расписаний. — М.: Наука, 1975.
2. Конвей Р. В., Максвелл В. Л., Миллер Л. В. Теория расписаний. — М.: Наука, 1975.
3. Horn W. A. Single-Machine Job Sequencing With Treelike Precedence Ordering and Linear Delay Penalties. — SIAM J. Appl. Math., 1972, 23, 1.
4. 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. — Минск: изд-во Института технической кибернетики АН БССР, 1973.
5. Adolphson D., Hu T. C. Optimal Linear Ordering. — SIAM J. Appl. Math., 1973, 25, 3.
6. Sidney J. B. Decomposition Algorithms for Single-Machine Sequencing with Precedence Relations and Deferral Costs. — Oper. Res., 1975, 23, No. 2.
7. Ullman J. D. NP-Complete Scheduling Problems. — Journ. of Computer and System Sciences, 1975, 10.
8. Лившиц Э. М. Последовательность операций при изготовлении сложной детали. — М.: Автоматика и телемеханика, 1968, № 11.
9. Smith W. E. Various Optimizers for Single Stage Production. — Naval Res. Logist. Quart., 1956, 3, No. 1.
10. Джонсон С. Оптимальное расписание для двух- и трехступенчатых процессов. — Кибернетический сборник. Новая серия. — М.: Мир, 1965, вып. 1.
11. Дискретная математика и математические вопросы кибернетики/Под ред. С. В. Яблонского и О. Б. Лупанова. — М.: Наука, 1974.
12. Knuth D. E. The Art of Computer Programming. — Sorting and Searching, Addison-Wesley, 1973, 3.
13. Symposium on the Theory of Scheduling and its Applications. — Berlin, Heidelberg, New York: Springer-Verlag, 1973.

Поступила в редакцию
16.VIII 1977