АКАДЕМИЯ НАУК СССР

АВТОМАТИКА И ТЕЛЕМЕХАНИКА

3

МОСКВА · 1978

УДК 65.012.1.122.007

О МИНИМАЛЬНОМ ЧИСЛЕ ОПЕРАТОРОВ ДЛЯ ОБСЛУЖИВАНИЯ ОДНОРОДНОГО ЛИНЕЙНОГО ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

А. В. КАРЗАНОВ, Э. М. ЛИВШИЦ

(Москва)

Предлагается эффективный алгоритм нахождения минимального числа автооператоров, осуществляющих перенос однородных деталей между процессорами при последовательной непрерывной обработке.

1. Постановка задачи

Пусть с интервалом времени T на обработку поступают одинаковые детали. Каждая деталь в одном и том же порядке должна пройти n стадий автоматической обработки (условия однородности и линейности технологического процесса). Стадию обработки с номером i осуществляет процессор A_i за время $t_i \leqslant T$, $i=1, 2, \ldots, n$, причем в каждый момент он обрабатывает не более одной детали. Перенесение детали с одного процессора на следующий производится (авто)оператором. Время переноса оператором детали от процессора A_i к A_{i+1} равно ρ_i, $i=1, 2, \ldots, n-1$, а время перехода оператора от A_i к A_j равно ν_{ij}, $2 \leqslant i \leqslant n$, $1 \leqslant j \leqslant n-1$. Необходимо, чтобы после обработки детали i-м процессором она сразу же переносилась на A_{i+1} (условие непрерывности). Требуется определить минимальное число операторов для обслуживания описанного технологического процесса и составить алгоритм их работы (задача $\mathfrak{A}$). При этом предполагается, что каждый оператор является конечным автоматом, откуда следует, что движение оператора должно быть периодическим. Такая задача возникает в области планирования и управления производственными процессами, в частности при электролитической обработке деталей.

Задачи о минимальном числе операторов в теории расписаний рассматривались рядом авторов (обзор результатов приведен в [1]). Рассматриваемый процесс ввиду своей специфики моделируется на бесконечном графе «событий», имеющем регулярную структуру и допускающем «свертывание» к конечному графу весьма простого вида. Благодаря этому алгоритм решения задачи довольно прост и имеет низкую трудоемкость.

Ниже приводится алгоритм решения задачи $\mathfrak{A}$ и его обоснование. Алгоритм иллюстрируется на конкретном примере. Рассматривается также процесс с дополнительным условием физического характера.

2. Алгоритм

Опишем алгоритм решения задачи $\mathfrak{A}$.

1. Определяем числа $T_1, T_2, \ldots, T_{n-1}$ по следующей рекуррентной формуле:

$$T_1 = t_1, \quad T_{i+1} = (T_i + \rho_i + t_{i+1}) \pmod T.$$

2. Строим матрицу $D=\|d_{i,j}\|_{n-1}^{n-1}$: $d_{i,j}=T_j-T_i+k_{ij}T$, где k_{ij} — минимальное целое k, при котором

$$T_j-T_i+kT\geqslant\rho_i+\nu_{i+1,j}.$$

3. Решаем задачу о назначении [2] для матрицы D, т. е. определяем такой набор ее элементов $K=\{(i(q), j(q))\}$, $q=1, 2, \ldots, n-1$, что: а) в каждой строке, а также в каждом столбце ровно по одному элементу из этого набора, б) при этом сумма $\sum_{q=1}^{n-1}d_{i(q),j(q)}$ минимальна. Как будет показано при обосновании алгоритма, минимальное число операторов равно $\sum_{q=1}^{n-1} d_{i(q),j(q)}/T$.

4. Опишем работу операторов, определяемую набором K.

а. Выделяем циклы $C_1, C_2, \ldots, C_\gamma$ набора K $(K=U_{v=1}^{\gamma} C_v)$. Под циклом C понимается такая совокупность элементов

$$\{(i(q_1), j(q_1)), (i(q_2), j(q_2)), \ldots, (i(q_s), j(q_s))\}\leqslant K,$$

что $j(q_1)=i(q_2), j(q_2)=i(q_3), \ldots, j(q_s)=i(q_1)$.

Замечание 1. Частным случаем цикла является один элемент $(i(q), j(q))$ при $i(q)=j(q)$.

б. Каждому циклу $C_r=\{(i(q_1), j(q_1)), (i(q_2), j(q_2)), \ldots, (i(q_s), j(q_s))\}$ соответствует своя группа операторов $B_{r(1)}, B_{r(2)}, \ldots, B_{r(\chi)(C_r))}$. Здесь $\chi(C_r)=d(C_r)/T$, где $d(C_r)=\sum_{\alpha=1}^{s}d_{i(q_\alpha),j(q_\alpha)}$ — «вес» цикла (ниже будет показано, что величина $\chi(C_r)$ целая). Эти операторы будут осуществлять все операции переноса деталей от процессоров $A_{i(q_1)}, A_{i(q_2)}, \ldots, A_{i(q_s)}$. Пусть для определенности $A_{i(q_1)}$ — процессор с наименьшим номером $i(q_1)$ среди рассматриваемых. Оператор $B_{r(1)}$ начинает работу в момент окончания обработки первой детали процессором $A_{i(q_1)}$. Перенеся эту деталь к процессору $A_{i(q_1)+1}$, он переходит к $A_{i(q_2)}$ и дожидается конца обработки им очередной детали, после чего переносит эту деталь к $A_{i(q_2)+1}$ и переходит к $A_{i(q_3)}$ и т. д. В конце концов $B_{r(1)}$, «обслужив» процессор $A_{i(q_s)}$, переходит к $A_{i(q_1)}$ и продолжает работу по тем же правилам. Как будет показано в дальнейшем, время одного цикла оператора $B_{r(1)}$ равно $d(C_r)$.

Оператор $B_{r(2)}$ в точности повторяет движение $B_{r(1)}$, будучи «сдвинутым» от него по времени на T (т. е. он начинает свою работу с того же процесора $A_{i(q_1)}$ и второй детали), и, вообще, $B_{r(p)}$ сдвинут по времени относительно $B_{r(1)}$ на $(p-1)T$.

Аналогичным образом определяется работа групп операторов, обслуживающих остальные циклы.

Оценка числа действий алгоритма. Число действия алгоритма на этапах 1 и 4 может быть оценено, очевидно, как $O(n)$, на этапе 2 — как $O(n^2)$. Решение задачи о назначении (этап 3) может быть получено при помощи алгоритма [3], имеющего оценку $O(n^3)$. Таким образом, общая оценка алгоритма $O(n^3)$.

3. Графовые модели и анализ задачи

Пусть k — номер детали в порядке поступления на обработку. Можно считать, что детали начали поступать как угодно давно, т. е. $-\infty<k<+\infty$. Обозначим $T_{k,i}^{\text{кон}}$ — календарное время конца обработки k-й детали i-м процессором. Условия однородности, линейности и непрерывности процесса можно записать в виде

$$A1.\ T_{k,i+1}^{\text{кон}}-T_{k,i}^{\text{кон}}=\rho_i+t_{i+1},\quad -\infty<k<+\infty,\quad 1\leqslant i\leqslant n-1;$$

$$A2.\ T^{\text{кон}}_{k+1,i}-T^{\text{кон}}_{k,i}=T,\quad -\infty<k<+\infty,\quad 1\leqslant i\leqslant n.$$

Построим следующий бесконечный граф G. Его вершины взаимно-однозначно соответствуют моментам $T^{\text{кон}}_{k,i}$, $-\infty<k<+\infty$, $i=1, 2, \ldots, n-1$, причем если моменты $T^{\text{кон}}_{k',i'}$ и $T^{\text{кон}}_{k'',i''}$ календарно одинаковые, им все же соответствуют разные вершины графа G. Сохраним за вершинами графа G те же обозначения $T^{\text{кон}}_{k,i}$. Из вершины $T^{\text{кон}}_{k',i'}$ ведет ориентированная дуга в вершину $T^{\text{кон}}_{k'',i''}$, если и только если $T^{\text{кон}}_{k'',i''}-T^{\text{кон}}_{k',i'}\geqslant \rho_{i'}+\nu_{i'+1,\,i''}$, т. е. если и только если оператор, перенеся деталь k' от процессора $A_{i'}$ к процессору $A_{i'+1}$, успевает подойти к процессору $A_{i''}$ не позже конца обработки им детали k''.

Лемма 1. Отображение $f^l:T^{\text{кон}}_{k,i}\to T^{\text{кон}}_{k+l,i}$ является автоморфизмом графа G.

Доказательство леммы вытекает из свойств A1 и A2.

Назовем действием перенесение некоторой детали от одного процессора к следующему. Из определения графа G следует, что каждому действию взаимно-однозначно соответствует вершина графа G.

Легко видеть, что последовательности действий каждого (m-го) оператора соответствует бесконечный путь в графе G, причем, как следует из условий задачи, этот путь периодический, т. е. существует такое целое s_m, что если $T^{\text{кон}}_{k,i}$ — вершина пути, то $T^{\text{кон}}_{k+s_m,i}$— так же вершина этого пути (минимальное такое число s_m называется периодом пути). Обратно, каждый периодический путь в G задает допустимое движение оператора.

Из того, что все действия должны быть выполнены и никакие два оператора не совершают одного и того же действия, вытекает, что исходная задача эквивалентна следующей: найти в графе G минимальное число периодических непересекающихся путей, содержащих все вершины графа.

Пусть $T_i=T^{\text{кон}}_{k,i}(\mathrm{mod}\,T)$ (определение корректно, так как $\forall k', k'', i$ $T^{\text{кон}}_{k',i}=T^{\text{кон}}_{k'',i}$ $(\mathrm{mod}\,T)$— следствие из A2). Построим «свернутый» граф $\overline{G}$. Его вершинами будут «моменты» T_i, $i=1, 2, \ldots, n-1$. Каждой дуге $(T^{\text{кон}}_{k',i'}-T^{\text{кон}}_{k'',i''})$ графа G взаимно-однозначно соответствует дуга в графе $\overline{G}$ с началом $T_{i'}$ и концом $T_{i''}$. Будем обозначать эту дугу $\{T_{k',\,i'},\ T_{k'',\,i''}\}$, а ее весом назовем величину $l\{T_{k',i'},\ T_{k'',i''}\}=T^{\text{кон}}_{k'',i''}-T^{\text{кон}}_{k',i'}$. Таким образом, граф $\overline{G}$ является взвешенным мультиграфом с $n-1$ вершинами и бесконечным множеством дуг.

Обозначим через U_{ij} — множество дуг графа $\overline{G}$ с началом в T_i и концом в T_j (быть может, $i=j$). Множество U_{ij} разбивается на классы — каждый класс образован дугами равного веса. Из леммы 1 следует, что если $\{T_{k',\,i},\ T_{k'',\,j}\}$ — дуга в U_{ij}, то $\{T_{k'+l,\,i},\ T_{k''+l,\,j}\}$ так же является дугой в U_{ij} и с учетом $A2$ принадлежит к тому же классу. Кроме того, если $\{T_{k',\,i},\ T_{k'',\,j}\}\in U_{ij}$, то $\{T_{k',\,i},\ T_{k''+1,\,j}\}\in U_{ij}$ и $l\{T_{k',i},\ T_{k''+1,\,j}\}=l\{T_{k',\,i},\ T_{k'',\,j}\}+$ $+T$. Следовательно, U_{ij} состоит из бесконечного числа классов, веса дуг в «соседних» классах отличаются на T.

Замечание 2. Вид графа $\overline{G}$ не зависит от точки отсчета календарного времени, т. е. при переходе к новому календарному времени $T'^{\text{кон}}_{k,i}=$ $=T^{\text{кон}}_{k,i}+\Delta t$ граф $\overline{G}$ не меняется.

Замечание 3. Пусть $C=\{u_1, u_2, \ldots, u_N\}$ — ориентированный цикл в графе $\overline{G}$. Тогда величина $l(C)/T=\sum_{j=1}^{N} l\{u_j\}/T$ целая.

Докажем последнее. Пусть $u_j=\{T_{k'(j),\, i'(j)},\ T_{k''(j),\, i''(j)}\}$, $j=1,\ 2,\ \ldots,\ N$. Тогда

$$l(C)=\sum_{j=1}^{N}(T^{\text{кон}}_{k''(j),\, i''(j)}-T^{\text{кон}}_{k'(j),\, i'(j)})=\sum_{j=1}^{N}(T^{\text{кон}}_{k''(j),\, i''(j)}-T^{\text{кон}}_{k'(j+1),\, i'(j+1)}),$$

где $T_{k'(N+1),\, i'(N+1)}$ суть $T_{k'(1),\, i'(1)}$. Поскольку $i'(j+1)=i''(j)$, то в силу А2 $T^{\text{кон}}_{k''(j),\, i''(j)}-T^{\text{кон}}_{k'(j+1),\, i'(j+1)}\equiv 0 \pmod T$, $j=1,\ 2,\ \ldots,\ N$, откуда следует справедливость замечания 3.

Совокупность $P=\{\xi_1,\ \xi_2,\ldots,\xi_M\}$ периодических непересекающихся путей в графе G, содержащих все его вершины, будем называть планом, а еси при этом число путей минимальное — оптимальным планом задачи $\mathfrak{A}$. Пусть s_r — период пути ξ_r. Наименьшее общее кратное $s(P)$ чисел $s_1, s_2, \ldots, s_M$ назовем общим периодом совокупности P.

Зафиксируем момент времени t_0 и рассмотрим временной полуинтервал $(t_0,\ t_0+s(P)T)$. Предположим, путь ξ_r на этом полуинтервале последовательно проходит через вершины $T^{\text{кон}}_{k_{r(1)},\, i_{r(1)}},\ T^{\text{кон}}_{k_{r(2)},\, i_{r(2)}},\ldots,T^{\text{кон}}_{k_{r(N)},\, i_{r(N)}}$, а следующая вершина этого пути — $T^{\text{кон}}_{k_{r(N+1)},\, i_{r(N+1)}}$. Так как $s(P)$ — общий период совокупности P, то $i_{r(N+1)}=i_{r(1)}$ и $T^{\text{кон}}_{k_{r(N+1)},\, i_{r(N+1)}}-T^{\text{кон}}_{h_{r(1)},\, i_{r(1)}}=s(P)T$. Выделим на графе $\overline{G}$ дуги $\{T_{k_{r(m)},\, i_{r(m)}},\ T_{k_{r(m+1)},\, i_{r(m+1)}}\}$, $m=1, 2, \ldots, N$, соответствующие дугам пути ξ_r. Поступим так с каждым путем из P. Обозначм $l(P)=\Sigma l\{u\}$, где суммирование ведется по всем выделенным дугам u, и $\overline{G}(P)$ — часть графа $\overline{G}$, содержащую все его вершины и выделенные дуги. Справедлива следующая лемма.

Лемма 2. 1. Из каждой вершины в графе $\overline{G}(P)$ исходит и в каждую вершину входит ровно $s(P)$ дуг *. 2. Число путей M равно $l(P)/s(P)T$.

Доказательство. На любом временном полуинтервале $[t,\ t+T)$ на каждом процессоре происходит ровно одно окончание обработки детали (ср. со свойством А2), откуда при фиксированном i число вершин $T^{\text{кон}}_{k,i}$ в полуинтервале $[t_0,\ t_0+s(P)T)$ равно $s(P)$. Теперь утверждение 1 непосредственно следует из того, что из каждой такой вершины исходит и в каждую такую вершину входит ровно одна дуга в объединении путей $\xi_1, \xi_2, \ldots, \xi_M$. Поскольку для любого пути $\xi_r\in P$ $T^{\text{кон}}_{k_{r(N+1)},\, i_{r(N+1)}}-T^{\text{кон}}_{k_{r(1)},\, i_{r(1)}}=s(P)T$, то $l(P)=\sum_{r=1}^{M}(T^{\text{кон}}_{k_{r(N+1)},\, i_{r(N+1)}}-T^{\text{кон}}_{k_{r(1)},\, i_{r(1)}})=Ms(P)T$, откуда следует утверждение 2. Лемма доказана.

Рассмотрим произвольный d-фактор $G^{(d)}$ графа $\overline{G}$. Любой ориентированный d-фактор можно представить в виде объединения d 1-факторов (см., например, [4], где приводится соответствующая теорема для d-регулярного двудольного графа, которая легко переделывается в данное утверждение).

Обозначим $l(\Gamma)$ сумму весов дуг взвешенного графа Γ, и если Γ — m-фактор, то положим $\chi(\Gamma)=l(\Gamma)/mT$.

Лемма 3. При любом разложении графа $G^{(d)}$ на 1-факторы ($G^{(d)}=G_1\cup G_2\cup\ldots\cup G_d$) среди них найдется такой 1-фактор G_p, что $\chi(G_p)\leqslant\chi(G^{(d)})$.

Доказательство. Очевидно $\chi(G^{(d)})=\dfrac{1}{d}\sum_{p=1}^{d}\chi(G_p)$, т. е. $\chi(G^{(d)})$ есть среднее арифметическое чисел $\chi(G_p)$, что и доказывает лемму.

* Ориентированный граф, в котором для каждой вершины число исходящих, а также число входящих дуг равно d, называется (ориентированным) d-фактором.

Пусть Γ — 1-фактор графа $\bar{G}$. Покажем, как по Γ найти план задачи $\mathfrak{A}$ с числом операторов $\chi(\Gamma)$. Граф Γ представляет собой объединение непересекающихся циклов ($\Gamma = C_1 \cup C_2 \cup \ldots \cup C_\gamma$). Рассмотрим цикл C_j, $1 \leqslant j \leqslant \gamma$. Пусть

$$C_j = \{T_{j(1)}, u_{j(1)}, T_{j(2)}, u_{j(2)}, \ldots, T_{j(q)}, u_{j(q)}, T_{j(1)}\}.$$

По замечанию 3 величина $\chi(C_j) = \Sigma_{p=1}^{q} l\{u_{j(p)}\}/T$ целая.

Зададим пути операторов, которые будут обслуживать все операции переноса деталей от процессов $A_{j(1)}, A_{j(2)}, \ldots, A_{j(q)}$. Зафиксируем номер k_1 и построим путь ξ_1 первого оператора. «Первая» вершина этого пути — $T^{\text{кон}}_{k_1, j(1)}$. Пусть $u_{j(1)} = \{T_{k'(1), j(1)}, T_{k''(1)\, j(2)}\}$, тогда следующей вершиной в ξ_1 будет $T^{\text{кон}}_{k_2, j(2)}$, где $k_2 = k_1 + (k''(1) - k'(1))$. Вообще, если проделано p шагов ($p \leqslant q$) и найдены вершины $T^{\text{кон}}_{k_1, j(1)}, T^{\text{кон}}_{k_2, j(2)}, \ldots, T^{\text{кон}}_{k_p j(p)}$ и дуга $u_{j(p)}$ суть $\{T_{k'(p), j(p)}, T_{k''(p), j(p+1)}\}$, то следующей вершиной в ξ_1 будет $T^{\text{кон}}_{k_{p+1}, j(p+1)}$, где $k_{p+1} = k_p + (k''(p) - k'(p))$ (в случае $p = q - j(p+1) = j(1)$). Построенные $q+1$ вершин и определяют периодическую часть пути ξ_1. Период пути ξ_1, т. е. $(T^{\text{кон}}_{k_{q+1}, j(1)} - T^{\text{кон}}_{k_1, j(1)})/T$, равен $\chi(C_j)$. Путь ξ_2 второго оператора получается из пути ξ_1 сдвигом на T, третьего — на $2T$. Путь $\xi_{\chi(C_j)}$ последнего оператора этой группы получается из ξ_1 сдвигом на $(\chi(C_j) - 1)T$. Легко убедиться, что пути $\xi_1, \xi_2, \ldots, \xi_{\chi(C_j)}$ не пересекаются и содержат в совокупности все вершины вида $T^{\text{кон}}_{k, j(p)}$, $-\infty k < +\infty$, $1 \leqslant p \leqslant q$ и только их.

Ту же самую процедуру проделаем и для любого другого цикла $C_{j'}$, $1 \leqslant j' \leqslant \gamma$. Таким образом, мы найдем план задачи $\mathfrak{A}$ с числом операторов $\chi(\Gamma) = \Sigma_{j=1}^{\gamma} \chi(C_j)$.

4. Обоснование алгоритма

Подведем итоги анализа. Пусть P — план задачи $\mathfrak{A}$. Тогда на основании леммы 2 граф $\bar{G}(P)$ является $s(P)$-фактором, а число операторов этого плана равно $\chi(\bar{G}(P))$. По лемме 3 в $\bar{G}(P)$ найдется 1-фактор G_p, для которого $\chi(G_p) \leqslant \chi(\bar{G}(P))$. Из рассмотренного выше следует, что 1-фактор G_p определяет план задачи $\mathfrak{A}$ с числом операторов $\chi(G_p)$, не большим, чем число операторов для плана P. Таким образом, справедлива следующая теорема.

Теорема. Оптимальный план задачи $\mathfrak{A}$ достаточно искать среди планов, соответствующих 1-факторам графа $\bar{G}$. 1-фактор Γ с минимальной суммой весов дуг определяет оптимальный план.

Пусть T_i и T_j — произвольные вершины графа $\bar{G}$ (быть может, $i = j$). Ясно, что если 1-фактор Γ, определяющий оптимальный план, содержит дугу $u \in U_{ij}$, то эта дуга u имеет минимальный вес среди всех дуг из U_{ij}. Таким образом, поиск оптимального 1-фактора Γ достаточно проводить на графе $\tilde{G} \subset \bar{G}$, в котором для любых i, j, $1 \leqslant j$, $j \leqslant n-1$, вместо множества дуг U_{ij} присутствует одна дуга минимального веса. Заметим, что в графе $\tilde{G}$ на каждой вершине имеется петля с весом, большим нуля.

Задача нахождения 1-фактора минимального веса в полном ориентированном графе $\tilde{G}$ эквивалентна задаче о назначении, а именно: зададим матрицу $D = D(\tilde{G}) = \|d_{i,j}\|_{n-1}^{n-1}$, полагая $d_{i,j} = l\{T_i, T_j\}$. Рассмотрим набор элементов матрицы, являющийся решением задачи о назначении. Каждому элементу $d_{i,j}$ в этом наборе соответствует дуга $\{T_i, T_j\}$ графа $\tilde{G}$. Из условия «а» для задачи о назначении (см. шаг 3 алгоритма) следует, что совокупность таких дуг образует 1-фактор графа $\tilde{G}$, условие же «б» эквивалентно тому, что этот 1-фактор имеет минимальный вес.

Теперь для обоснования алгоритма нам надо показать, что числа $d_{i,j}$, определенные в алгоритме,— это в точности веса дуг графа $\tilde{G}$. Очевидно, что числа T_i, определенные в алгоритме, с точностью до фиксированного слагаемого совпадают с числами T_i в определении графа $\bar{G}$. Рассмотрим дугу $\{T_i, T_j\}$ графа $\bar{G}$. Ей соответствует некоторая дуга $(T^{\text{кон}}_{k',i}, T^{\text{кон}}_{k'',j})$ графа G и $l\{T_i, T_j\}=T^{\text{кон}}_{k'',j}-T^{\text{кон}}_{k',i}$. Но $T^{\text{кон}}_{k'',j}-T^{\text{кон}}_{k',i} \geqslant \rho_i+\nu_{i+1,j}$ или $T_j-T_i+kT \geqslant \rho_i+\nu_{i+1,j}$, откуда $d_{i,j} \geqslant l\{T_i, T_j\}$. Обратно, пусть k_{ij} — минимальное целое k, при котором $T_j-T_i+kT \geqslant \rho_i+\nu_{i+1,j}$. Выберем произвольно k' и подберем такое k'', чтобы $T^{\text{кон}}_{k'',j}-T^{\text{кон}}_{k',i}=T_j-T_i+k_{ij}T$, что в силу А2 всегда можно сделать. Тогда $(T^{\text{кон}}_{k',i}, T^{\text{кон}}_{k'',j})$ — дуга графа G и, следовательно,

$$d_{i,j}=T^{\text{кон}}_{k'',j}-T^{\text{кон}}_{k',i} \geqslant l\{T_i, T_j\}.$$

Этим заканчивается обоснование алгоритма.

Замечание 4. Можно показать, что полученное решение оптимально и в классе произвольных (не обязательно периодических) движений операторов.

Это замечание носит чисто теоретический характер. Наметим план доказательства. Пусть $P=\{\xi_1, \xi_2, \ldots, \xi_M\}$ — некоторый (не обязательно периодический) план и s — достаточно большое целое число. Рассмотрим полуинтервал $[t_0, t_0+sT)$, и пусть путь ξ_r на этом полуинтервале последовательно проходит через вершины

$$T^{\text{кон}}_{k_{r(1)}, i_{r(1)}}, \; T^{\text{кон}}_{k_{r(2)}, i_{r(2)}}, \; \ldots, \; T^{\text{кон}}_{k_{r(N)}, i_{r(N)}}.$$

На графе $\bar{G}$ выделим дуги $\{T_{k_{r(m)}, i_{r(m)}}, T_{k_{r(m+1)}, i_{r(m+1)}}\}$, $m=1, 2, \ldots, N-1$, а также выделим дугу u_r из $U_{i_{r(N)} i_{r(1)}}$, имеющую минимальный вес, для которой

$$\sum_{m=1}^{N-1} l\{T_{k_{r(m)}, i_{r(m)}}, T_{k_{r(m+1)}, i_{r(m+1)}}\} + l\{u_r\} \geqslant sT.$$

Поступим так с каждым путем из P. Тогда по аналогии с леммой 2 доказывается, что выделенные дуги образуют s-фактор в G и что число путей M равно $\lim\limits_{s \to \infty} l_s/sT$, где l_s — суммарный вес выделенных дуг. Но по лемме 3 для каждого s найдется 1-фактор $G(s)$ такой, что $\chi(G(s)) \leqslant l_s/sT$.

5. Пример

Проиллюстрируем работу алгоритма на следующем примере. Число процессоров равно 7, $T=10$; значения t_i, $1 \leqslant i \leqslant 7$, ρ_i, $1 \leqslant i \leqslant 6$, и $\nu_{i,j}$, $2 \leqslant i \leqslant 7$, $1 \leqslant j \leqslant 6$ приведены в таблице (*а*) (обратим внимание на то, что равенства нулю чисел $\nu_{i,i}$ в условиях

t_1	5
t_2	6
t_3	9
t_4	10
t_5	9
t_6	8
t_7	$\leqslant 10$

ρ_1	5
ρ_2	3
ρ_3	3
ρ_4	2
ρ_5	4
ρ_6	2

	1	2	3	4	5	6
2	2	7	5	6	1	9
3	5	6	0	7	2	4
4	2	9	5	11	1	10
5	1	10	3	15	0	10
6	8	15	2	12	5	0
7	8	15	1	10	5	15

а

$\nu_{i,j}$

T_1	5
T_2	6
T_3	8
T_4	1
T_5	2
T_6	4

б

	1	2	3	4	5	6
1	10	21	13	**16**	7	19
2	9	**10**	12	15	6	8
3	7	18	10	23	**4**	16
4	**4**	15	7	20	11	13
5	13	24	6	19	10	**12**
6	11	22	**4**	17	8	20

$D = \| d_{i,j} \|$

задачи не требуется). Вычисленные значения T_i, $1 \leqslant i \leqslant 6$ и $d_{i,j}$, $1 \leqslant i, j \leqslant 6$, даны в таблице (б). Решая задачу о назначении с матрицей $D = \| d_{i,j} \|$, получаем набор элементов $\{(1,4), (2,2), (3,5), (4,1), (5,6), (6,3)\}$ (в таблице (б) они даны жирным шрифтом). Этот набор имеет следующие циклы: $C_1 = \{(1,4), (4,1)\}$, $C_2 = \{(2,2)\}$, $C_3 = \{(3,5), (5,6), (6,3)\}$. Веса этих циклов равны: $d(C_1) = 16+4 = 20$, $d(C_2) = 10$, $d(C_3) = 4+12+4 = 20$. Первая группа операторов в количестве $\chi(C_1) = 20:10 = 2$ осуществляет перенос деталей от процессоров A_1 к A_2 и A_4 к A_5. Вторая группа состоит из одного оператора (поскольку $\chi(C_2) = 10:10 = 1$). Этот оператор переносит детали от A_2 к A_3. Третья группа из двух операторов ($\chi(C_3) = 20:10 = 2$) обслуживает процессоры A_3, A_5, A_6, последовательность движений каждого оператора — перенос от A_3 к A_4, затем от A_5 к A_6, затем от A_6 к A_7 и возвращение к A_3.

6. Система без столкновений

В рассмотренной модели на траектории движения каждого оператора никаких ограничений не накладывалось. В реальной системе может потребоваться, чтобы операторы двигались по общим «рельсам», проложенным мимо каждого из процессоров. В этом случае необходимо, чтобы движения операторов исключали столкновения. Как для такой системы определить минимальное число операторов и составить программу их работы? Будем предполагать следующее:

1) замедления и ускорения операторов не учитываются,
2) скорости операторов на всех участках движения постоянны,
3) операторы могут передавать друг другу детали.

Теперь нетрудно показать, как обойтись тем же числом операторов, что и в системе без ограничений. Пусть $\mathfrak{A}$ — план для задачи без ограничений и $\xi_1, \xi_2, \ldots, \xi_M$ — маршруты операторов. В начальный момент все операторы начинают движение, предписанное планом $\mathfrak{A}$. Пусть t — момент первого столкновения в системе (пусть это случилось с операторами B_p и B_q). Тогда в момент t эти операторы должны передать друг другу свои функции, т. е. обменяться деталями (деталью) и «движением» (B_p должен далее двигаться как B_q, и наоборот). Тем же способом ликвидируются и дальнейшие столкновения в системе.

План для задачи без столкновений можно получить из плана $\mathfrak{A}$ как алгоритмически, так и графически. Заметим, что в этом случае движение каждого оператора останется периодическим.

Поступила в редакцию
17 марта 1977 г.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. *Танаев В. С., Шкурба В. В.* Введение в теорию расписаний. «Наука», 1975.
2. *Kuhn H. W.* The Hungarian method for the assignment problem. Naval Res. Logist. Quart., v. 2, pp. 83—97, 1955.
3. *Диниц Е. А., Кронрод М. А.* Один алгоритм решения задачи о назначении. Докл. АН СССР, т. 189, № 1, стр. 23—25, 1969.
4. *Харари Ф.* Теория графов. «Мир», 1973.

ON THE MINIMAL NUMBER OF OPERATORS IN SERVING A UNIFORM LINEAR PROCESS

A. V. KARZANOV, E. M. LIFSHITS

The paper suggests and effective algorithm for finding a minimal number of autooperators which transfer parts of the same kind between processors in sequential continuous processing.