

О ЗАДАЧЕ АКЕРСА-ФРИДМАНА

В.В.Сервах

Задача Акерса-Фридмана заключается в минимизации времени выполнения N работ на M машинах. Работа n ($n = 1, 2, \dots, N$) состоит из τ_n операций, выполняемых в заданной последовательности. Операция i работы n выполняется на машине с номером $m_{ni} \in \{1, 2, \dots, M\}$ непрерывно в течение τ_{ni} единиц времени ($n = 1, 2, \dots, N$; $i = 1, 2, \dots, \tau_n$). Невозможно одновременное выполнение на одной машине двух и более операций.

Подробно эта задача рассматривается в [3, гл. 6]. Из других работ выделим [1], где предложена удобная геометрическая интерпретация данной задачи, и [2], в которой имеется малотрудоемкий алгоритм отыскания ее точного решения для случая двух работ.

В настоящей статье предложен малотрудоемкий алгоритм нахождения точного решения задачи Акерса-Фридмана в случае трех работ и целочисленности (или соизмеримости) длительностей операций. Для этого алгоритма при условии, что число выполняемых на одной машине операций ограничено константой, получены верхние оценки трудоемкости $O(MS \log_2 M)$ и необходимого объема памяти $O(MS)$, где S - суммарная длительность всех операций.

§ 1. Геометрическая интерпретация задачи

Кратко изложим некоторые результаты [1] в удобной для нас форме. Отметим:

$$S_{n,i} = \sum_{j=1}^{i-1} \tau_{nj} \quad (n = 1, 2, \dots, N; \quad i = 1, 2, \dots, \tau_n),$$

$$S_n = S_{n, \tau_n + 1} = \sum_{j=1}^{\tau_n} \tau_{nj} \quad (n = 1, 2, \dots, N).$$

Рассмотрим N -мерное пространство с метрикой $\rho(x, y) = \max_{n=1, 2, \dots, N} |x_n - y_n|$.

Каждой работе n ($n = 1, 2, \dots, N$) поставим в соответствие координатную ось и отметим на ней точки $S_{n,1}, S_{n,2}, \dots, S_{n,\tau_n}, S_n$. Значение координаты x_n ($0 \leq x_n \leq S_n$) будем интерпретировать как состояние выполнения работы n . Тогда процесс выполнения работ может быть задан в виде пути из точки

$(0, 0, \dots, 0)$ в точку (S_1, S_2, \dots, S_N) . Такой путь является ломаной линией. Каждый ее отрезок соответствует одновременному выполнению некоторого множества работ $\{n_1, n_2, \dots, n_k\}$. Он образует равные острые углы с осями $x_{n_1}, x_{n_2}, \dots, x_{n_k}$, а к остальным осям перпендикулярен. Длина этого отрезка полагается равной расстоянию между его конечными точками. Длина пути есть сумма длин всех отрезков ломаной.

Блоком будем называть область рассматриваемого пространства, соответствующую одновременному выполнению двух операций на одной машине. Каждый блок является N -мерным параллелепипедом $\{\bar{x} | S_{\alpha,i} < x_{\alpha} < S_{\alpha,i+1}, S_{\beta,j} < x_{\beta} < S_{\beta,j+1}, 0 \leq x_{\gamma} \leq S_{\gamma} (\gamma = 1, 2, \dots, N, \gamma \neq \alpha, \gamma \neq \beta)\}$, где операции i и j соответственно работ α и β таковы, что $m_{\alpha i} = m_{\beta j}$. Это множество обозначим через $B(i, j, \alpha, \beta)$. Путь, не пересекающий ни одного блока, назовем допустимым.

Для решения задачи Акерса-Фридмана необходимо найти кратчайший допустимый путь из точки $(0, 0, \dots, 0)$ в точку (S_1, S_2, \dots, S_N) . Определим некоторое конечное множество путей \mathcal{P} , содержащее требуемый оптимум. Первоначально из точки $(0, 0, \dots, 0)$ проводим отрезок под равными углами ко всем осям. Далее, продолжая уже построенные части путей, каждый раз проводим отрезок под равными острыми углами к максимально возможному числу осей. По достижении некоторого блока $B(i, j, \alpha, \beta)$ осуществим его обход, причем двумя способами, которые соответствуют двум различным порядкам выполнения операций i и j на машине $m = m_{\alpha i} = m_{\beta j}$. Ветвление пути произойдет в точке его первого пересечения с одной из следующих гиперплоскостей: $\{\bar{x} | x_{\alpha} = S_{\alpha,i}\}$ или $\{\bar{x} | x_{\beta} = S_{\beta,j}\}$. Все ветви, которые образуются в процессе обхода блоков, строим по тем же правилам и продолжаем до точки (S_1, S_2, \dots, S_N) .

Т е о р е м а. Множество \mathcal{P} содержит кратчайший допустимый путь между точками $(0, 0, \dots, 0)$ и (S_1, S_2, \dots, S_N) .

В [1] это утверждение доказывается методом математической индукции по числу недопустимых областей. Каждая такая область является объединением $(N(N-1))/2$ блоков, соответствующих некоторой машине m , и задается совокупностью операций (i_1, i_2, \dots, i_N) , для которых $m_{i_1} = m_{i_2} = \dots = m_{i_N} = m$. Однако с помощью предложенного метода можно установить справедливость теоремы только при $N=2$. При $N \geq 3$ после нескольких ветвлений пути не гарантируется уменьшение количества недопустимых областей в непройденной части параллелепипеда и индуктивный переход не удается осуществить.

Рассмотрим путь $\pi_0 \in \mathcal{P}$. Блок $B(i, j, \alpha, \beta)$ назовем π_0 -смежным, если непусто пересечение пути π_0 с одним из следующих множеств:

$$\{\bar{x} | x_{\alpha} = S_{\alpha,i}, S_{\beta,j} < x_{\beta} < S_{\beta,j+1},$$

$$0 \leq x_{\gamma} \leq S_{\gamma} (\gamma = 1, 2, \dots, N; \gamma \neq \alpha, \gamma \neq \beta)\},$$

$$\{\bar{x} \mid S_{\alpha,i} < x_{\alpha} < S_{\alpha,i+1}, x_{\beta} = S_{\beta,j},$$

$$0 \leq x_{\gamma} \leq S_{\gamma} \ (\gamma = 1, 2, \dots, N; \gamma \neq \alpha, \gamma \neq \beta)\}.$$

Допустимый путь π назовем π_0 -согласованным, если для каждого π_0 -смежного блока $B(i, j, \alpha, \beta)$ пути π и π_0 определяют одинаковый порядок выполнения операций i и j . Обозначим через t_{ni}^{π} время начала выполнения операции i работы n , отвечающее пути π .

Л е м м а. Если $\pi_0 \in \mathcal{P}$, то для каждого π_0 -согласованного пути π и любых $n = 1, 2, \dots, N$, $i = 1, 2, \dots, r_n$ справедливо неравенство $t_{ni}^{\pi_0} \leq t_{ni}^{\pi}$.

Д о к а з а т е л ь с т в о. Предположим противное: пусть существуют такие n и i , что $t_{ni}^{\pi_0} > t_{ni}^{\pi}$. Определим n_0 и i_0 из соотношения

$$t_{n_0 i_0}^{\pi_0} = \min_{n=1, \dots, N, i=1, \dots, r_n} \{t_{ni}^{\pi_0} \mid t_{ni}^{\pi_0} > t_{ni}^{\pi}\} = M^*.$$

Если для некоторых n и i допустить равенство $\tau_{ni} = 0$, то указанный минимум может достигаться сразу на нескольких операциях, технологически связанных между собой. В этом случае выбираем ту из них, которая выполняется первой. Пусть $\bar{p}^1, \bar{p}^2, \bar{p}^3$ - точки рассматриваемого пространства, лежащие на пути π , расстояния до которых от начала координат вдоль этого пути равны соответственно

$$t_{n_0, i_0-1}^{\pi_0} + \tau_{n_0, i_0-1}^{\pi}, \quad t_{n_0, i_0}^{\pi} \quad \text{и} \quad t_{n_0, i_0}^{\pi_0}.$$

При $i_0 = 1$ будем полагать $t_{n_0, i_0-1}^{\pi_0} = t_{n_0, i_0-1}^{\pi} = 0$ и $\tau_{n_0, i_0-1} = 0$.

Легко показать, что $\rho_n^1 \leq \rho_n^2 \leq \rho_n^3$ ($n = 1, 2, \dots, N$) и $\rho_n^2 < \rho_n^3$, хотя бы при одном $n \in \{1, 2, \dots, N\}$. При этом $\rho_{n_0}^1 = \rho_{n_0}^2 = \rho_{n_0}^3 = S_{n_0, i_0}$. Первое вытекает из следующих неравенств:

$$t_{n_0, i_0-1}^{\pi_0} + \tau_{n_0, i_0-1}^{\pi} \leq t_{n_0, i_0-1}^{\pi} + \tau_{n_0, i_0-1}^{\pi} \leq t_{n_0, i_0}^{\pi} < t_{n_0, i_0}^{\pi_0}.$$

Здесь $t_{n_0, i_0-1}^{\pi_0} \leq t_{n_0, i_0-1}^{\pi}$, в силу $t_{n_0, i_0-1}^{\pi_0} < t_{n_0, i_0}^{\pi_0} = M^*$. Второе верно по-

тому, что точки \bar{p}^1 и \bar{p}^3 соответствуют одному и тому же состоянию выполнения работы n_0 . Таким образом, участок пути π_0 между точками \bar{p}^2 и \bar{p}^3 имеет ненулевую длину и целиком лежит в гиперплоскости $\{\bar{x} \mid x_{n_0} = S_{n_0, i_0}\}$.

Такое возможно только в случае обхода блоков вида $B(i_0, j, n_0, \beta)$, так как каждый отрезок пути π_0 проходит под равными острыми углами к максимально возможному числу осей. Причем обход завершается в точке \bar{p}^3 . Поэтому хотя бы для одного из указанных блоков выполнено равенство $t_{n_0, i_0}^{\pi_0} = t_{\beta j}^{\pi_0} + \tau_{\beta j}$.

Отсюда $t_{\beta j}^{\pi_0} < t_{n_0, i_0}^{\pi_0} = M^*$. Значит, $t_{\beta j}^{\pi_0} \leq t_{\beta j}^{\pi}$. Соотношения $\rho_{n_0}^2 = \rho_{n_0}^3 = S_{n_0, i_0}$, $\rho_{\beta}^2 < S_{\beta, j+1} = \rho_{\beta}^3$ возможны только при пересечении пути π_0 с множеством

$$\{\bar{x} | x_{n_0} = S_{n_0, i_0}, S_{\beta, j} < x_{\beta} < S_{\beta, j+1}, \\ 0 \leq x_{\gamma} \leq S_{\gamma} \ (\gamma = 1, 2, \dots, N, \gamma \neq n_0, \gamma \neq \beta)\}.$$

Следовательно, блок $B(i_0, j, n_0, \beta)$ является π_0 -смежным. А так как путь π , по условию леммы, π_0 -согласованный, то $t_{n_0, i_0}^{\pi} \geq t_{\beta, j}^{\pi} + \tau_{\beta, j}$. Окончательно получаем

$$t_{n_0, i_0}^{\pi} \geq t_{\beta, j}^{\pi} + \tau_{\beta, j} \geq t_{\beta, j}^{\pi_0} + \tau_{\beta, j} = t_{n_0, i_0}^{\pi_0},$$

что противоречит нашему предположению.

Доказательство теоремы. Рассмотрим произвольный допустимый путь π . Легко построить такой путь $\pi_0 \in \mathcal{P}$, для которого π является π_0 -согласованным. Действительно, достаточно в процессе формирования множества \mathcal{P} по достижении некоторого блока $B(i, j, \alpha, \beta)$ рассмотреть только один вариант его обхода. А именно тот, при котором построенная ветвь и путь π определяют одинаковый порядок выполнения операций i и j . Пути π и π_0 удовлетворяют условиям леммы. Значит, $t_{ni}^{\pi_0} \leq t_{ni}^{\pi}$ ($n = 1, 2, \dots, N$; $i = 1, \dots, \tau_n$). Пусть ℓ и ℓ_0 - длины путей π и π_0 соответственно. Имеем:

$$\begin{aligned} \ell_0 &= \max_{n=1, \dots, N, i=1, \dots, \tau_n} (t_{ni}^{\pi_0} + \tau_{ni}) = t_{n^*i^*}^{\pi_0} + \tau_{n^*i^*} \leq \\ &\leq t_{n^*i^*}^{\pi} + \tau_{n^*i^*} \leq \max_{n=1, \dots, N, i=1, \dots, \tau_n} (t_{ni}^{\pi} + \tau_{ni}) = \ell. \end{aligned}$$

Так как путь π выбран произвольно, то теорема доказана.

§ 2. Общая схема алгоритма

Опишем идею малотрудоёмкого поиска минимума на множестве \mathcal{P} . Мощность \mathcal{P} растёт, вообще говоря, экспоненциально с ростом величин τ_n ($n = 1, 2, \dots, N$). Однако многие пути этого множества частично совпадают. Одинаковые участки путей выгодно строить предварительно и только один раз. Для реализации этого подхода в параллелепипеде $\{\bar{x} | 0 \leq x_n \leq S_n, n = 1, 2, \dots, N\}$ определённым образом зафиксируем множество точек, называемых узловыми. По описанным выше правилам можно определить множество путей между любой узловой точкой и вершиной параллелепипеда (S_1, S_2, \dots, S_N) . Чтобы избежать повторных построений, каждый из этих путей проводим только до ближайшего пересечения с узловой точкой. Информацию о построенных участках и их последовательном расположении удобно задавать с помощью взвешенного сетевого графа. Каждая вершина этого графа соответствует некоторой узловой точке, а каждая дуга отвечает некоторому участку пути между двумя узловыми точками. Вес дуги полагается рав-

ным длине этого участка. Такое представление позволяет свести нашу задачу к задаче отыскания кратчайшего пути в сетевом графе. Небольшой пример, иллюстрирующий описанную схему, изображен на рис. 1 и 2. Блоки заштрихованы. Узловые точки выделены и перенумерованы. Около каждой дуги выписан ее вес.

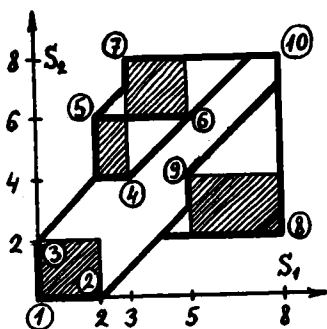


Рис. 1

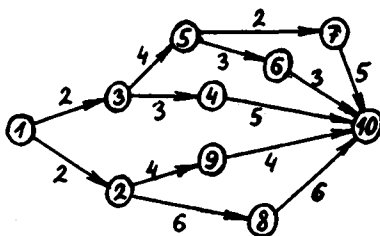


Рис. 2

§ 3. Постановка и решение вспомогательной двумерной задачи

При $N=2$ блок $B(i, j, 1, 2) = \{S_{1i} < x < S_{1, i+1}, S_{2j} < y < S_{2, j+1}\}$ будем задавать упорядоченной парой (i, j) . Рассмотрим следующую двумерную задачу.

Задача Д. В параллелепипеде $\{(x, y) | 0 \leq x \leq S_1, 0 \leq y \leq S_2\}$ имеется L блоков (i_ℓ, j_ℓ) ($\ell = 1, 2, \dots, L$) и вне этих блоков задано K точек (x_k, y_k) ($k = 1, 2, \dots, K$). Необходимо для каждого K определить номер ближайшего блока $\ell(x_k, y_k)$, с которым пересечется луч, проведенный из точки (x_k, y_k) под равными углами к обоим осям. При отсутствии пересечений луча с блоком будем полагать $\ell(x_k, y_k) = 0$.

Определим величины:

$$z_k = x_k - y_k, \quad k = 1, 2, \dots, K;$$

$$c_\ell = S_{1, i_\ell} - S_{2, j_\ell + 1}; \quad d_\ell = S_{1, i_\ell + 1} - S_{2, j_\ell}, \quad \ell = 1, 2, \dots, L.$$

Для фиксированного K выполнение условий

$$c_\ell < z_k < d_\ell, \tag{1}$$

$$x_k < S_{1, i_\ell + 1}, \quad y_k < S_{2, j_\ell + 1}$$

означает, что луч, проведенный из точки (x_k, y_k) под равными углами к обоим осям, пересекает блок с номером ℓ . Таких блоков может быть несколько. Необходимо выбрать тот из них, для которого пара (i_ℓ, j_ℓ) лексикографически минимальна.

Пусть $(\ell_1, \ell_2, \dots, \ell_L)$ — номера блоков, расположенные в порядке лексикографического возрастания пар (i_ℓ, j_ℓ) , и $v_0 = \min\{v \in \{1, 2, \dots, L\} | S_{1, i_{\ell_v} + 1} >$

$\{x_k, s_{2, j_{e_v}+1} > y_k\}$. Введем характеристическую функцию $\theta(v)$, $v=1, 2, \dots, L$. Полагаем $\theta(v) = 1$, если блок с номером e_v удовлетворяет условию (1), и $\theta(v) = 0$ - в противном случае. Требуется найти минимальное значение v^* , для которого $v^* \geq v_0$ и $\theta(v^*) = 1$. Это можно сделать за $O(\log_2 L)$ шагов, если предварительно известны суммарные значения функции θ для пар индексов, четверок, восьмерок и т.д. Построим полное бинарное дерево высоты $H = \lceil \log_2 L \rceil$. Перенумеруем 2^H листьев этого дерева слева направо и листу $v = 1, 2, \dots, L$ поставим в соответствие значение $\theta(v)$. При $v = L+1, L+2, \dots, H$ доопределим $\theta(v) = 0$. Для остальных вершин дерева полагаем $\theta(v) = \theta(v') + \theta(v'')$, где v' и v'' - сыновья вершины v . Поиск оптимума v^* начинаем с листа v_0 . При $\theta(v_0) = 1$ имеем $v^* = v_0$. Пусть $\theta(v_0) = 0$. Последовательность вершин дерева $v_0, v_1, \dots, v_{\eta_0}, \dots, v_{\eta_0+1}$ определим индуктивно. Если $v_{\eta-1}$ - левый сын, то v_{η} - непосредственный предок вершины $v_{\eta-1}$. Если $v_{\eta-1}$ - правый сын, то v_{η} - правый сосед вершины $v_{\eta-1}$. Значение η_0 найдем из двух следующих условий:

- 1) $\theta(v_0) = \theta(v_1) = \dots = \theta(v_{\eta_0}) = 0$,
- 2) либо v_{η_0+1} не определена, либо $\theta(v_{\eta_0+1}) > 0$.

Первый случай имеет место, если $\theta(v) = 0$ для всех $v \in \{v_0, v_0+1, \dots, L\}$. Тогда значение v^* не определено и $\ell(x_k, y_k) = 0$. Перейдем ко второму случаю. Неравенство $\theta(v_{\eta_0+1}) > 0$ означает, что дерево с корнем v_{η_0+1} содержит листья с ненулевой характеристической функцией. Необходимо найти самый левый из них. На очередном шаге перейдем в корень левого поддерева вершины v_{η_0+1} , если в нем функция θ имеет ненулевое значение. В противном случае переходим в корень правого поддерева. Из вновь определенной вершины переход на следующий уровень осуществляем аналогично. Процедура завершает работу при достижении уровня H . Тем самым будет определено значение v^* , для которого $\theta(v_0) = \theta(v_0+1) = \dots = \theta(v^*-1) = 0$ и $\theta(v^*) = 1$. Тогда $\ell(x_k, y_k) = \ell_{v^*}$.

Теперь опишем алгоритм решения задачи Д. Первоначально для всех вершин дерева полагаем $\theta(v) = 0$. Найдем такую перестановку $(\ell_1, \ell_2, \dots, \ell_L)$, что лексикографически $(i_{\ell_1}, j_{\ell_1}) < (i_{\ell_2}, j_{\ell_2}) < \dots < (i_{\ell_L}, j_{\ell_L})$. Числа $\{z_1, z_2, \dots, z_K, c_1, c_2, \dots, c_L, d_1, d_2, \dots, d_L\}$ будем рассматривать в порядке неубывания. Возможны следующие случаи:

- 1) Очередным числом является c_e . Тогда для листа v такого, что $e_v = e$, полагаем $\theta(v) = 1$, а для всех его предков значение функции θ увеличиваем на единицу.
- 2) Очередным числом является d_e . В этом случае для листа v такого, что $e_v = e$, полагаем $\theta(v) = 0$, а для всех его предков значение функции θ уменьшаем на единицу.
- 3) Очередным числом является z_k . Методом дихотомии находим $v_0 = \min\{v \in \{1, 2, \dots, L\} \mid s_{1, i_{e_v}+1} > x_k, s_{2, j_{e_v}+1} > y_k\}$ и применяем описанную выше процедуру поиска $\ell(x_k, y_k)$. Если v_0 не определено, то $\ell(x_k, y_k) = 0$.

Трудоёмкость однократного обращения к каждому из пп. 1) - 3) не превосходит $O(\log_2 L)$ операций, а общее число обращений равно $\mathcal{K} + 2L$. Для предварительного упорядочивания необходимо $\mathcal{K} \log_2 \mathcal{K} + 3L \log_2 L$ операций. Таким образом, верхняя оценка трудоёмкости решения задачи Д составляет

$$O(\mathcal{K} \log_2 \mathcal{K} + L \log_2 L + \mathcal{K} \log_2 L).$$

Заметим, что предложенный алгоритм позволяет найти точное решение задачи Акерса-Фридмана для случая двух работ в самой общей постановке: при произвольных длительностях операций τ_{ni} ($n = 1, 2$; $i = 1, \dots, \tau_n$) и произвольной совокупности блоков (i_ℓ, j_ℓ) ($\ell = 1, 2, \dots, L$). В качестве узловых рассмотрим точки $(S_1, i_{\ell+1}, S_2, j_\ell)$, $(S_1, i_\ell, S_2, j_{\ell+1})$, ($\ell = 1, 2, \dots, L$), $(0, 0)$ и (S_1, S_2) . Установим соответствие между этими точками и $2L+2$ вершинами графа: $(0, 0) \leftrightarrow 1$, $(S_1, S_2) \leftrightarrow 2L+2$, $(S_1, i_{\ell+1}, S_2, j_\ell) \leftrightarrow 2\ell$, $(S_1, i_\ell, S_2, j_{\ell+1}) \leftrightarrow 2\ell+1$ ($\ell = 1, 2, \dots, L$). Пусть (x_K, y_K) - координаты узловой точки с номером K . Для каждого $K = 1, 2, \dots, 2L+1$ находим $\ell(x_K, y_K)$. Список дуг сетевого графа формируем следующим образом: если $\ell(x_K, y_K) = 0$, то вводим дугу $(K, 2L+2)$ с весом $\max\{S_1 - x_K, S_2 - y_K\}$. При $\ell(x_K, y_K) = \ell > 0$ вводим дуги $(K, 2\ell)$ и $(K, 2\ell+1)$ с весами $S_1, i_{\ell+1} - x_K$ и $S_2, j_{\ell+1} - y_K$ соответственно. В построенном графе определяем кратчайший путь между вершинами 1 и $2L+2$. Общая трудоёмкость алгоритма при $N=2$ не превосходит $O(L \log_2 L)$ операций.

§ 4. Некоторые построения для случая трех работ

Пусть операция i работы α и операция j работы β таковы, что $m_{\alpha i} = m_{\beta j}$. Без ограничения общности можно положить $\alpha < \beta$. Через γ обозначим номер третьей работы. Упорядоченная тройка чисел (i, j, γ) однозначно определяет блок $B(i, j, \alpha, \beta)$. Величину γ назовем направлением блока. Число блоков направления γ обозначим через L_γ . Их множество будем задавать списком $\{(i_\ell, j_\ell, \gamma)\}_{\ell=1, 2, \dots, L_\gamma}$.

Подробнее опишем правила ветвления путей, используемые при построении множества \mathcal{P} . Каждую ветвь пути, выходящего из некоторой точки $\bar{u} = (\bar{u}_1, \bar{u}_2, \bar{u}_3)$ будем продолжать до завершения обхода некоторого блока. Первоначально рассмотрим блоки только одного направления γ и путь проведем до ближайшего пересечения с одним из них. Пусть ℓ_γ - номер этого блока. Длина построенного участка равна $\rho_\gamma = \max\{S_{\alpha, i} - u_\alpha, S_{\beta, j} - u_\beta\}$, где $i = i_{\ell_\gamma}$, $j = j_{\ell_\gamma}$. Если путь минует все блоки данного направления, то полагаем $\ell_\gamma = 0$ и $\rho_\gamma = \max\{S_{\alpha} - u_\alpha, S_{\beta} - u_\beta\}$. Заметим, что для определения значений ℓ_γ и ρ_γ достаточно рассмотреть луч, проведенный в плоскости $\{\bar{x} | x_\gamma = 0\}$ из точки (u_α, u_β) под равными углами к осям X_α и X_β . Ближайший блок, с которым этот луч пересекается, и является требуемым. Аналогично находим величины $\ell_\alpha, \ell_\beta, \rho_\alpha$ и ρ_β . Для упрощения вида формул введем новые обозначения: $i_{\ell_1} = \mu_1$, $j_{\ell_1} = \nu_1$, $i_{\ell_2} = \lambda_2$.

$j_{e_2} = v_2$, $i_{e_3} = \lambda_3$, $j_{e_3} = \mu_3$. (Если $e_\gamma = 0$, то доопределим $i_{e_\gamma} = j_{e_\gamma} = 0$.) Таким образом, имеем три блока: $(\mu_1, v_1, 1)$, $(\lambda_2, v_2, 2)$ и $(\lambda_3, \mu_3, 3)$. Расстояния до них равны соответственно ρ_1 , ρ_2 и ρ_3 . Пусть $\min\{\rho_1, \rho_2, \rho_3\} = \rho_1$. Опишем правила ветвления пути.

Случай 1. Пусть $e_1 = 0$. Существует единственный путь из точки \bar{u} в точку (S_1, S_2, S_3) , длина которого равна $\rho(S) = \max\{S_1 - u_1, S_2 - u_2, S_3 - u_3\}$.

Случай 2. Пусть $e_1 \neq 0$, $\mu_1 \neq \mu_3$, $v_1 \neq v_2$. Возможны два варианта обхода блока $(\mu_1, v_1, 1)$. Проводим ветви до точек \bar{a}^1 и \bar{a}^2 , где

$$\bar{a}^1 = (u_1 + \rho(\bar{a}^1), S_{2, \mu_1+1}, S_{3, v_1}), \rho(\bar{a}^1) = S_{2, \mu_1+1} - u_2,$$

$$\bar{a}^2 = (u_1 + \rho(\bar{a}^2), S_{2, \mu_1}, S_{3, v_1+1}), \rho(\bar{a}^2) = S_{3, v_1+1} - u_3.$$

Случай 3. Пусть $e_1 \neq 0$, $\mu_1 = \mu_3$, $v_1 \neq v_2$. Тогда одна из ветвей, построенных в случае 2, до завершения обхода блока $(\mu_1, v_1, 1)$ ветвится еще раз. При этом осуществляется обход блока $(\lambda_3, \mu_3, 3) = (\lambda_3, \mu_1, 3)$. Проводим три ветви до следующих точек:

$$\bar{b}^1 = (S_1, \lambda_3, S_{2, \mu_1+1}, S_{3, v_1}), \rho(\bar{b}^1) = S_{2, \mu_1+1} - u_2,$$

$$\bar{a}^2 = (u_1 + \rho(\bar{a}^2), S_{2, \mu_1}, S_{3, v_1+1}), \rho(\bar{a}^2) = S_{3, v_1+1} - u_3,$$

$$\bar{a}^3 = (S_1, \lambda_3+1, S_{2, \mu_1}, S_{3, v_1}), \rho(\bar{a}^3) = S_1, \lambda_3+1 - u_1.$$

Если $\mu_1 \neq \mu_3$ и $v_1 = v_2$, то осуществляем обход блоков $(\mu_1, v_1, 1)$ и $(\lambda_2, v_2, 2) = (\lambda_2, v_1, 2)$. Определяем ветви до точек \bar{a}^1 , $\bar{b}^2 = (S_1, \lambda_3, S_{2, \mu_1}, S_{3, v_1+1})$ и \bar{a}^3 .

Случай 4. Пусть $e_1 \neq 0$, $\mu_1 = \mu_3$, $v_1 = v_2$. Требуется обойти все три блока: $(\mu_1, v_1, 1)$, $(\lambda_2, v_1, 2)$ и $(\lambda_3, \mu_1, 3)$. Проводим ветви до точек \bar{b}^1 , \bar{b}^2 и \bar{a}^3 .

Другие случаи, когда $\min\{\rho_1, \rho_2, \rho_3\} = \rho_\gamma < \rho_1, \gamma = 2, 3$, аналогичны.

Возможно, что при условии $S_{\alpha, i} < u_\alpha < S_{\alpha, i+1}$ блок вида $B(i, j, \alpha, \beta)$ необходимо обойти двумя различными способами. По правилам ветвления обход, соответствующий первоначальному выполнению операции j работы β , завершается в точке \bar{a} с координатами $a_\alpha = S_{\alpha, i}$, $a_\beta = S_{\beta, j+1}$, a_γ . При этом $\rho(\bar{u}, \bar{a}) = a_\beta - u_\beta$. Такая ветвь не является допустимой, поскольку $u_\alpha > a_\alpha = S_{\alpha, i}$. На самом деле ветвление произойдет до того, как будет достигнута точка \bar{u} . Каждый путь, ведущий в эту точку, будет ветвиться при его пересечении с плоскостью $\{\bar{x} | x_\alpha = S_{\alpha, i}\}$. Новый участок, проведенный в этой плоскости, пройдет через точки \bar{a} и \bar{u}' с координатами $u'_\alpha = S_{\alpha, i}$, $u'_\beta = u_\beta$, $u'_\gamma = u_\gamma$. Получаем, что каждому пути из \bar{O} в \bar{u} отвечает единственный путь из \bar{O} в \bar{u}' , причем той же длины. Также справедливо $\rho(\bar{u}', \bar{a}) = \rho(\bar{u}, \bar{a}) = a_\beta - u_\beta$. Это позволяет не рассматривать каждое ветвление отдельно, а использовать одно фиктивное, что и сделано в описанных правилах. Если оптимальный путь, найденный в результате работы алгоритма, будет содержать недопустимую ветвь между точками \bar{u} и \bar{a} , то \bar{u} необходимо заменить на \bar{u}' .

В качестве узловых рассмотрим конечные точки возможных ветвей. Из правил ветвления следует, что такие точки принадлежат множеству

$$R = \bigcup_{\gamma=1}^3 \bigcup_{\ell=1}^{L_\gamma} \bigcup_{q=1}^2 R_{\ell,q}^\gamma,$$

где $R_{\ell,q}^\gamma$ - ребро блока (i_ℓ, j_ℓ, γ) . При $\alpha < \beta$ имеем

$$R_{\ell,1}^\gamma = \{\bar{x} \mid x_\alpha = S_{\alpha, i_\ell+1}, x_\beta = S_{\beta, j_\ell}, 0 \leq x_\gamma \leq S_\gamma\},$$

$$R_{\ell,2}^\gamma = \{\bar{x} \mid x_\alpha = S_{\alpha, i_\ell}, x_\beta = S_{\beta, j_\ell+1}, 0 \leq x_\gamma \leq S_\gamma\}.$$

Кроме того, необходимо, чтобы множество узловых точек было возможно меньшим, так как от его мощности зависит трудоемкость алгоритма. Лучший результат удалось получить в случае, когда длительности операций τ_{ni} ($n=1,2,3$; $i=1,2,\dots,\tau_n$) - целые числа. Тогда узловыми являются все целочисленные точки ребер $R_{\ell,q}^\gamma$, $\gamma=1,2,3$; $\ell=1,2,\dots,L_\gamma$, $q=1,2$, а также точки $(0,0,0)$ и (S_1, S_2, S_3) .

§ 5. Описание алгоритма для случая трех работ

Этап 1. Предварительные вычисления.

Рассмотрим работы α и β и блоки (i_ℓ, j_ℓ, γ) , $\ell=1,2,\dots,L_\gamma$. Спроецируем каждую узловую точку на плоскость $\{\bar{x} \mid x_\gamma = 0\}$. Полученная совокупность точек является объединением двух следующих множеств:

$$\{(x_\alpha, x_\beta) \mid x_\alpha \in \{S_{\alpha,1}, S_{\alpha,2}, \dots, S_{\alpha,\tau_\alpha}, S_\alpha\}, x_\beta \in \{0,1,\dots,S_\beta\}\},$$

$$\{(x_\alpha, x_\beta) \mid x_\alpha \in \{0,1,\dots,S_\alpha\}, x_\beta \in \{S_{\beta,1}, S_{\beta,2}, \dots, S_{\beta,\tau_\beta}, S_\beta\}\}.$$

Для каждого из этих множеств и двумерных блоков (i_ℓ, j_ℓ) , $\ell=1,2,\dots,L_\gamma$, решим задачу Д. Значения величин $\ell(x_\alpha, x_\beta)$ будем хранить в двумерных массивах A_α^β и A_β^α размерностей $(\tau_\alpha+1) \times (S_\beta+1)$ и $(\tau_\beta+1) \times (S_\alpha+1)$ соответственно. При этом $A_\alpha^\beta(i, x) = \ell(S_{\alpha,i}, x)$, $A_\beta^\alpha(i, x) = \ell(x, S_{\beta,i})$. Решая еще четыре аналогичные задачи, определим значения элементов массивов $A_\alpha^\gamma, A_\gamma^\alpha, A_\beta^\gamma$ и A_γ^β . Оценим трудоемкость решения первой из шести задач. Количество точек в данном случае равно $K = (\tau_\alpha+1) \cdot (S_\beta+1)$, а количество блоков - $L = L_\gamma$. Достаточно просто показать, что элементы множества $\{\bar{x} = x_\alpha - x_\beta \mid x_\alpha \in \{S_{\alpha,1}, S_{\alpha,2}, \dots, S_{\alpha,\tau_\alpha}, S_\alpha\}, x_\beta \in \{0,1,\dots,S_\beta\}\}$ могут быть упорядочены по неубыванию за $(\tau_\alpha+1) \cdot (S_\beta+1)$ операций. Следовательно, для вычисления элементов массива A_γ^β требуется $O((\tau_\alpha+1) \cdot (S_\beta+1) + L_\gamma \log_2 L_\gamma + (\tau_\alpha+1) \cdot (S_\beta+1) \log_2 L_\gamma)$ операций. Так как $L_\gamma < \tau_\alpha \cdot \tau_\beta < (\tau_\alpha+1) \cdot (S_\beta+1)$, то выписанное выражение можно упростить и будем иметь $O(\tau_\alpha S_\beta \times \log_2 L_\gamma)$. Таким образом, трудоемкость реализации первого этапа алгоритма не превосходит $O(\sum_{\gamma=1}^3 (\tau_\alpha S_\beta + \tau_\beta S_\alpha) \log_2 L_\gamma)$ операций.

Этап 2. Нумерация узловых точек и вершин сетевого графа.

Узловую точку \bar{u} , принадлежащую ребру $R_{\ell, q}^{\gamma}$, будем определять четырьмя параметрами: (γ, ℓ, q, x) , где $\gamma \in \{1, 2, 3\}$, $\ell \in \{1, 2, \dots, L_{\gamma}\}$, $q \in \{1, 2\}$, $x \in \{0, 1, \dots, S_{\gamma}\}$. Если $q = 1$, то координаты точки равны $u_{\alpha} = S_{\alpha, \ell+1}$, $u_{\beta} = S_{\alpha, j_{\ell}}$, $u_{\gamma} = x$. В противном случае $u_{\alpha} = S_{\alpha, \ell}$, $u_{\beta} = S_{\beta, j_{\ell}+1}$, $u_{\gamma} = x$. Здесь и всюду $\alpha < \beta$. Для хранения номеров узловых точек сформируем три массива \mathcal{D}_{γ} ($\gamma = 1, 2, 3$) размерностей $L_{\gamma} \times 2 \times (S_{\gamma} + 1)$, полагая $\mathcal{D}_{\gamma}(\ell, q, x) =$

$$= \sum_{\gamma'=1}^{\gamma-1} 2 \cdot L_{\gamma'}(S_{\gamma'}+1) + 2(S_{\gamma}+1)(\ell-1) + (S_{\gamma}+1)(q-1) + x + 1.$$

В строящемся сетевом графе точке с параметрами (γ, ℓ, q, x) поставим в соответствие вершину с номером $w = \mathcal{D}_{\gamma}(\ell, q, x)$, а точкам с координатами $(0, 0, 0)$ и (S_1, S_2, S_3) - вершины с номерами 0 и $w = \sum_{\gamma=1}^3 2 L_{\gamma}(S_{\gamma}+1) + 1$.

Этап 3. Формирование списка дуг сетевого графа.

Рассмотрим узловую точку $\bar{u} = (u_1, u_2, u_3)$ с параметрами (γ, ℓ, q, x) . Применяя правила ветвления, построим ветви, выходящие из этой точки. Используемые при построении величины ℓ_1, ℓ_2 и ℓ_3 принимают следующие значения: $\ell_{\alpha} = A_{\beta}^{\gamma}(j_{\ell} + \delta_{2q}, x)$, $\ell_{\beta} = A_{\alpha}^{\gamma}(\ell + \delta_{1q}, x)$, $\ell_{\gamma} = A_{\alpha}^{\beta}(\ell + \delta_{1q}, S_{\beta, j_{\ell}} + \delta_{2q})$, где $\delta_{kq} = 1$, если $k = q$, и $\delta_{kq} = 0$ - в противном случае. Пусть $\mathcal{D}_{\gamma}(\ell, q, x) = w$. Если $\bar{u} = (0, 0, 0)$, то $\ell_{\gamma} = A_{\alpha}^{\beta}(\ell, 0)$, $\gamma = 1, 2, 3$ и $w = 0$. Определим номера конечных ветвей. Для рассмотренных в правилах ветвления случаев имеем:

$$\bar{a}^1 \rightarrow w_1 = \mathcal{D}_1(\ell_1, 1, u_1 + \rho(\bar{a}^1)), \quad \bar{b}^1 \rightarrow w_1' = \mathcal{D}_1(\ell_1, 1, S_1, \lambda_3),$$

$$\bar{a}^2 \rightarrow w_2 = \mathcal{D}_1(\ell_1, 2, u_2 + \rho(\bar{a}^2)), \quad \bar{b}^2 \rightarrow w_2' = \mathcal{D}_1(\ell_1, 2, S_1, \lambda_3).$$

Точка \bar{a}^3 имеет номер w_3 , равный либо $\mathcal{D}_3(\ell_3, 1, S_3, \lambda_1)$, либо $\mathcal{D}_2(\ell_2, 1, S_2, \lambda_1)$. Каждой ветви поставим в соответствие дугу графа, выходящую из вершины w . Например, в четвертом из рассмотренных случаев вводятся три дуги (w, w_1') , (w, w_2') и (w, w_3) , а в первом - одна (w, W) . Дуге (w, w') припишем вес $\rho(w, w')$, равный длине соответствующей ветви.

При составлении списка дуг сетевого графа описанные построения выполняем последовательно для каждой узловой точки, за исключением (S_1, S_2, S_3) . Запомним адрес \mathcal{X}_w ($w = 0, 1, \dots, W-1$), начиная с которого записаны дуги, выходящие из вершины w .

Необходимо показать, что имеет место взаимно-однозначное соответствие между \mathcal{P} и множеством путей в графе из вершины 0 в вершину W . Рассмотрим произвольно $\pi \in \mathcal{P}$. Пусть $\bar{u}^0 = 0, \bar{u}^1, \bar{u}^2, \dots, \bar{u}^{n-1}, \bar{u}^n = \bar{S}$ - последовательность узловых точек, лежащих на этом пути. Точке \bar{u}^k ($k = 1, 2, \dots, n$) отвечают номера $w_1^k, w_2^k, \dots, w_{d_k}^k$ ($1 \leq d_k \leq 6$). Причем $d_0 = d_n = 1$, $w_1^0 = 0$ и $w_1^n = W$. Из

правил формирования списка дуг графа следует, что участку пути π , заключенному между точками \bar{u}^K и \bar{u}^{K+1} , $K = 0, 1, \dots, n-1$, ставятся в соответствие ровно d_K дуг: $(w_1^K, \tilde{w}_1), (w_2^K, \tilde{w}_2), \dots, (w_{d_K}^K, \tilde{w}_{d_K})$, где $\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_{d_K}$ - элементы множества $w_1^{K+1}, w_2^{K+1}, \dots, w_{d_K+1}^{K+1}$. Совокупность вершин $\{w_d^K\}_{d=1,2,\dots,d_K}^{K=0,1,\dots,n-1}$ и указанных дуг образуют подграф. Легко показать, что в этом подграфе существует единственный путь из 0 в W , который и соответствует пути π . В обратную сторону доказательство очевидно, так как каждой вершине графа отвечает только одна узловая точка.

Этап 4. Выделение максимального подграфа с единственной начальной вершиной $W = 0$.

Каждому $w = 0, 1, \dots, W$ поставим в соответствие величину $f_w \in \{0, 1\}$. Будем полагать $f_w = 1$, если установлено существование пути между вершинами 0 и w . Пусть F - некоторое множество вершин. Первоначально $F = \{0\}$, $f_0 = 1$, $f_w = 1$ и $f_w = 0$, $w = 1, 2, \dots, W-1$. Опишем один шаг процедуры выделения вершин требуемого подграфа. Выбираем произвольно $w \in F$ и рассматриваем все дуги вида (w, w') . Их расположение в списке задается адресом \tilde{x}_w . Если $f_{w'} = 0$, то включаем вершину w' во множество F и полагаем $f_{w'} = 1$. Вершину w из множества F исключаем и переходим к следующему шагу. Работа процедуры завершается, как только $F = \emptyset$.

Теперь исключим из списка все дуги (w', w'') , для которых $f_{w'} = 0$. Оставшийся список уплотним и вычислим новые значения адресов \tilde{x}_w .

Этап 5. Поиск кратчайшего пути между вершинами 0 и W .

Для каждой вершины w найдем число входящих в нее дуг p_w . Через g_w будем обозначать длину кратчайшего из построенных путей между вершинами 0 и w , а через t_w - предпоследнюю вершину этого пути. Пусть G - некоторое множество вершин. Первоначально $G = \{0\}$, $g_0 = 0$ и $g_w = \infty$ для всех $w \neq 0$. Опишем один шаг процедуры поиска необходимого пути. Выбираем произвольно $w \in G$. Начиная с адреса \tilde{x}_w , просматриваем все дуги вида (w, w') в выполнении неравенства $g_w + p(w, w') < g_{w'}$ полагаем $g_{w'} = g_w + p(w, w')$, $t_{w'} = w$. Уменьшаем на единицу значение $p_{w'}$. Если $p_{w'}$ стало равным нулю, то включаем вершину w' во множество G . Вершину w из множества G исключаем и переходим к следующему шагу. Работа процедуры завершается, когда $p_W = 0$. Полученное значение g_W задает длину кратчайшего пути. Вершины этого пути определяем с помощью рекуррентного соотношения $w_K = t_{w_{K-1}}$, $K = 1, 2, \dots$, где $w_0 = W$.

Для предложенного алгоритма определим верхние оценки трудоемкости T и необходимого объема памяти Π . Так как из каждой узловой точки выходит не более трех ветвей, то общее число дуг графа не превосходит $6 \sum_{\gamma=1}^n L_\gamma (S_\gamma + 1) + 3$.

Трудоемкость реализации последних трех этапов алгоритма линейно зависит от

числа дуг графа и, следовательно, равна $O(\sum_{\gamma=1}^3 L_{\gamma} S_{\gamma})$. Тогда

$$T \sim \sum_{\gamma=1}^3 (\tau_{\alpha} S_{\beta} + \tau_{\beta} S_{\alpha}) \log_2 L_{\gamma} + \sum_{\gamma=1}^3 S_{\gamma} L_{\gamma}$$

и

$$\Pi \sim \sum_{\gamma=1}^3 S_{\gamma} L_{\gamma}.$$

Величина L_{γ} равна $\sum_{m=1}^M K_{\alpha m} \cdot K_{\beta m}$, где K_{nm} - кратность использования машины m при выполнении работы n . Упростим вид выписанных выражений. Пусть

$$\tau = \max\{\tau_1, \tau_2, \tau_3\}, L = \max\{L_1, L_2, L_3\}, S = \sum_{\gamma=1}^3 S_{\gamma}. \text{ Тогда } T \sim S(\tau \log_2 L + L)$$

и $\Pi \sim S L$. Если для некоторой константы C выполнено $K_{nm} \leq C, n = 1, 2, 3, m = 1, 2, \dots, M$, то $T \sim S C M (\log_2 C^2 M + C) \sim S M \log_2 \Pi$ и $\Pi \sim S M$.

Поступила в ред.-изд.отдел

22 марта 1982 г.

Л и т е р а т у р а

1. Hardgrave W.W., Nemhauser G. A Geometric Model and Graphical Algorithm for a Sequencing Problem. - Operation Research, 1963, 11, № 6, p. 889-900.

2. Глебов Н.И. Алгоритм составления оптимального расписания для двух работ. - В кн.: Управляемые системы, Новосибирск, 1968, вып. I, с. 14-20.

3. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. - М.: Наука, 1975. - 360 с.

4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979. - 412 с.