

С. М. АНЦЫЗ, Л. Т. ПЕТРОВА

ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В СЕТЕВОМ ПЛАНИРОВАНИИ

1. Сетевой график и расписания

Основной моделью, на которой рассматриваются задачи сетевого планирования, является сетевой график - конечный ориентированный граф без контуров. Конкретным содержанием задачи определяются те или иные функции на вершинах и дугах графа и условия, которым эти функции должны удовлетворять. При постановке и решении задач существенно используются частичная упорядоченность вершин графа и некоторое отношение предпорядка для вершин. Рассмотрим эти вопросы подробнее.

Ориентированный граф G задается парой множеств X и U :

$$G = (X, U).$$

Здесь X - некоторое конечное множество элементов x, y, \dots , а множество U состоит из упорядоченных пар (x, y) некоторых элементов из X . Элементы множества X называются вершинами графа, элементы множества U - дугами. Дуга (x, y) ориентирована - вершина x является ее началом,

вершина y - концом. Путем из вершины x в вершину y называется такая последовательность дуг (x, x_1) (x_1, x_2) , ..., (x_n, y) , что начало каждой следующей дуги является концом предыдущей. Длинной пути называется число дуг, составляющих этот путь. Контуром называется конечный путь, начинающийся и заканчивающийся в одной и той же вершине. В дальнейшем рассматриваются графы без контуров.

Для вершин графа вводится следующее бинарное отношение $(<)$: x предшествует y , или y следует за x ; $(x < y)$, если в графе существует путь из x в y . Очевидно, что в ориентированном графе без контуров это отношение является отношением порядка, т.е. удовлетворяет условиям:

- 1) $x < x$ (т.к. всегда имеется тривиальный путь нулевой длины из x в x);
- 2) если $x < y$ и $y < z$, то $x < z$;
- 3) если $x < y$ и $y < x$, то $x = y$.

Таким образом, множество X вершин графа является частично упорядоченным множеством. Аналогично можно ввести порядок в множество дуг. Дуга (x, y) предшествует дуге (v, z) , если в графе существует путь из вершины y в вершину v .

При наличии в графе дуги (x, y) или (y, x) вершины x и y являются смежными. Пусть имеется дуга (x, y) . В этом случае x является левой смежной для y , а y является правой смежной для x (обозначается $x < < y$ или $y > > x$).

Каждой вершине x графа сопоставляется число $o(x)$, равное максимальной длине пути, заканчивающегося в данной вершине. Этим порождается следующее отношение предпорядка $(< *)$ в множестве X : вершина x мажорируется вершиной y ($x < * y$), если $o(x) \leq o(y)$. Рефлексивность и транзитивность этого отношения легко проверяется, а третье условие порядка для отношения $(< *)$ не выполняется. Отношением предпорядка порождается отношение эквивалентности (\sim) в множестве X . А именно, вершина x эквивалентна вершине y ($x \sim y$), если $x < * y$ и $y < * x$ (т.е. если $o(x) = o(y)$).

Таким образом, множество вершин X разбивается на классы X_i эквивалентных элементов: вершины x и y относятся к одному классу, если $o(x) = o(y)$. Обычным образом предпорядок $(< *)$ индуцирует порядок (старшинство) в

множестве классов эквивалентности: класс X_i , старше класса X_j , если $o(x) < o(y)$ при $x \in X_i, y \in X_j$. Так классы эквивалентности X_i могут быть упорядочены по возрастанию $o(x)$.

З а м е ч а н и е. Отметим, что никакие две вершины одного класса не являются сравнимыми по порядку ($<$), определенному выше. Очевидно также, что из соотношения $x < y$ следует $x \neq y$ (а обратное неверно).

Соответственно разбиению вершин на классы эквивалентности, множество дуг U также может быть разбито на классы (притом двумя способами). А именно, две дуги относятся к одному классу, если они заканчиваются (начинаются) в эквивалентных вершинах. Классы дуг также могут быть упорядочены по возрастанию $o(x)$. Такое упорядочение полезно в ряде алгоритмов анализа сетевых графиков. В работе [4] приведены программы вычисления на ЭВМ величин $o(x), x \in X$ и упорядочения дуг графа по классам.

Проиллюстрируем введенные понятия. Пусть граф G имеет множество вершин $X = \{x, y, z, t, s, r\}$ и множество дуг $U = \{(x, y), (x, s), (x, r), (y, z), (y, s), (s, t), (r, t)\}$. Если вершины изображать точками, а дуги стрелками, граф может быть представлен рисунком 1.

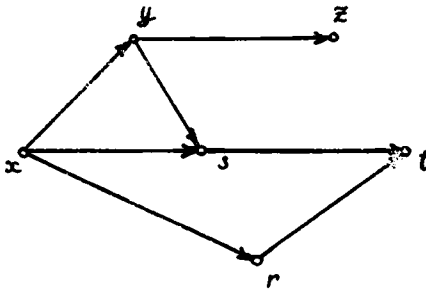


Рис. 1

Легко видеть, что, например, $x < t$, т.к. имеется путь $(x, s), (s, t)$ из x в t , а вершины z и r не сравнимы по порядку, т.к. в графе нет пути, соединяющего эти вершины. Просто определяются в этом примере величины $o(x)$ для вершин графа: $o(x) = 0, o(y) = 1, o(s) = 2, o(r) = 1, o(z) = 2, o(t) = 3$. Таким образом, вершина z мажорирует вершину r , $r < * z$, а вершины y и r (или z и s) являются эквивалентными и относятся к одному классу. Множество дуг раз-

бывается по конечным вершинам на следующие классы:

$$\begin{aligned} U_1 &= \{ (x, y), (x, r) \} , \\ U_2 &= \{ (y, z), (y, s), (x, s) \} , \\ U_3 &= \{ (s, t), (r, t) \} . \end{aligned}$$

В дальнейшем для удобства рассматривается граф с одной начальной вершиной a (в вершине a не заканчивается ни одна дуга) и одной конечной вершиной z (в вершине z не начинается ни одна дуга). Всегда можно формально перейти от графа с несколькими начальными и конечными вершинами к графу указанного вида. При этом не изменяется эквивалентность вершин и их частичная упорядоченность.

Итак, пусть на дугах (x, y) графа G задана функция $\tau(x, y) > 0$, т.е. имеется сетевой график $G(\tau)$. Как правило, сетевой график описывает технологию некоторого процесса во времени: дуги интерпретируются как некоторые работы, вершины являются событиями, символизирующими начало или завершение группы работ. Частичное упорядочение дуг отражает то технологическое требование, что никакая работа не может начаться раньше, чем закончатся все предшествующие ей работы. Функция $\tau(x, y)$ имеет смысл длительности работы.

Определим на вершинах графа функцию $\beta(x)$, $x \in X$, удовлетворяющую следующим соотношениям (*):

$$(*) \begin{cases} \beta(x) + \tau(x, y) \leq \beta(y) & \text{для всех } (x, y) \in U, \\ \beta(a) = 0. \end{cases}$$

Решение $\beta = \{ \beta(x), x \in X \}$ системы неравенств определяет некоторое расписание работ сетевого графика $G(\tau)$, допустимое его технологией. Время наступления начального события принимается за начало отсчета: $\beta(a) = 0$. Значение $\beta(x)$ определяет время наступления события x , значение $\beta(z) = T$ определяет срок завершения всего процесса. Таким образом, весь процесс, описываемый сетевым графиком, протекает в период времени от 0 до T , при этом каждая работа (x, y) длительности $\tau(x, y)$ должна выполняться в период времени от $\beta(x)$ до $\beta(y)$.

Как известно, простейшей задачей сетевого планирования является определение кратчайшего срока $\beta(z) = T^*$ завершения всего процесса и нахождение соответствующего оптимального

расписания:

минимизировать $\beta(z)$ при ограничениях

$$(*) \quad \begin{cases} \beta(x) + \tau(x, y) \leq \beta(y), & (x, y) \in U, \\ \beta(a) = 0. \end{cases}$$

Эта задача известна как задача о критическом пути, для ее решения имеются простые и эффективные алгоритмы (см., например, [1], [2], [3]). В частности, оптимальное расписание может быть построено по рекуррентным соотношениям:

$$\beta(y) = \max_{x \prec y} [\beta(x) + \tau(x, y)], \quad y \in X.$$

Алгоритм и его машинная реализация описаны в [3] и [4]. Из замечания, приведенного выше, следует, что если дуги графа упорядочены по классам, вычисление критического пути по этому алгоритму выполняется за один просмотр списка дуг.

При решении задачи о нахождении расписания $\{\beta(x), x \in X\}$, удовлетворяющего условиям:

$$(A^{\circ}) \quad \beta(x) + \tau(x, y) \leq \beta(y), \quad (x, y) \in U,$$

$$(B^{\circ}) \quad \beta(a) = 0,$$

$$(C^{\circ}) \quad \beta(z) \leq T \quad (T \geq T^*),$$

этот алгоритм критического пути позволяет определить два допустимых расписания $\{A(x), x \in X\}$ и $\{B(x), x \in X\}$ таких, что для каждой вершины x из X

$$A(x) = \min_{\beta} \beta(x),$$

$$B(x) = \max_{\beta} \beta(x).$$

Здесь минимум и максимум выбираются для данной вершины x по всем допустимым расписаниям. Таким образом, величина $A(x)$ определяет самый ранний из допустимых сроков наступления события x , а $B(x)$ - самый поздний срок.

В дальнейшем будем рассматривать расписания $\alpha(x, y)$, составленные для работ: $\alpha(x, y)$ определяет время начала работы (x, y) . Функция $\alpha(x, y)$ определяется на дугах графа и удовлетворяет соотношениям:

$$\alpha(x, y) + \tau(x, y) \leq \alpha(y, v), \quad (A)$$

$$(x, y) \in U, v \succ y,$$

$$\alpha(a, y) \geq 0, \quad y \succ a, \quad (B)$$

$$\alpha(x, z) + \tau(x, z) \leq T, \quad x \prec z (T \geq T^*). \quad (C)$$

Легко проверить, что системы неравенств $(A^0) - (C^0)$ и $(A) - (C)$ являются эквивалентными. Так, имея расписание $\alpha(x, y)$ для дуг $(x, y) \in U$, можно построить расписание $\beta(x)$ для вершин $x \in X$ по формулам:

$$\beta(x) = \min_{y \succ x} \alpha(x, y), \quad x \in X.$$

И наоборот, имея расписание $\beta(x)$ для вершин, можно получить расписание $\alpha(x, y)$ для дуг:

$$\beta(x) \leq \alpha(x, y) \leq \beta(y) - \tau(x, y),$$

$$(x, y) \in U.$$

Таким образом, при помощи алгоритма критического пути можно определить границы для каждой компоненты допустимого расписания $\alpha(x, y)$:

$$A(x) \leq \alpha(x, y) \leq B(y) - \tau(x, y), \quad (x, y) \in U.$$

Это обстоятельство будет использоваться в дальнейшем.

2. Постановка задач об оптимальных расписаниях

Рассмотрим некоторые задачи сетевого планирования, связанные с распределением ресурсов. Пусть имеется сетевой график

$G(\tau) = (X; U)$ с одной начальной вершиной $a \in X$ и одной конечной вершиной $z \in X$. Для каждой работы (x, y) заданы ее длительность $\tau(x, y) \geq 0$ и интенсивность потребления некоторого ресурса на этой работе $s(x, y) \geq 0$. Для простоты считается, что все работы данного графика используют один и тот же ресурс (рабочая сила, оборудование и пр.) и в некоторых случаях может быть задана функция $R(t)$ наличия этого ре-

сурса как функция времени. Задан также максимальный срок T завершения всего процесса ($T \geq T^*$).

Пусть имеется расписание $\alpha = \{\alpha(x, y), (x, y) \in U\}$, удовлетворяющее условиям (A) - (C). Для такого расписания α и каждого $t \in [0, T]$ определяются множества дуг $U(t; \alpha)$:

$$U(t; \alpha) = \{(x, y) \in U \mid t \in [\alpha(x, y), \alpha(x, y) + \tau(x, y)]\}$$

и величина $\sigma(t; \alpha)$:

$$\sigma(t; \alpha) = \sum_{(x, y) \in U(t; \alpha)} r(x, y).$$

$U(t; \alpha)$ представляет собой множество работ, выполняемых в момент времени $t \in [0, T]$ при данном расписании α . Величина $\sigma(t; \alpha)$ есть суммарная интенсивность потребления ресурса графиком в момент времени t при данном расписании α .

Сформулируем задачу о минимизации срока завершения процесса в следующем виде.

Задача 1. Найти расписание $\{\alpha(x, y), (x, y) \in U\}$ и величину T_0 , удовлетворяющие ограничениям

$$(A) \quad \alpha(x, y) + \tau(x, y) \leq \alpha(y, v), \quad (x, y) \in U, \quad v \rightarrow y,$$

$$(B) \quad \alpha(a, x) \geq 0, \quad x \rightarrow a,$$

$$(C) \quad \alpha(x, z) + \tau(x, z) \leq T_0, \quad x \leftarrow z,$$

$$(D) \quad R(t) - \sigma(t, \alpha) \geq 0, \quad t \in [0, T],$$

и доставляющие минимум величине T_0 .

В этой задаче минимизируется срок T_0 завершения всего процесса при заданных технологических ограничениях (A) - (C) на последовательность работ и заданных ограничениях (D) на используемые ресурсы. Функция $R(t)$ считается заданной.

Рассмотрим другую задачу, в которой оптимальность расписания понимается в ином смысле.

Задача 2. Найти расписание $\{\alpha(x, y), (x, y) \in U\}$ и величину ρ , удовлетворяющие ограничениям

$$\alpha(x, y) + \tau(x, y) \leq \alpha(y, v), \quad (x, y) \in U, \quad v \rightarrow y, \quad (A)$$

$$\alpha(a, x) \geq 0, \quad x \gg a, \quad (B)$$

$$\alpha(x, z) + \tau(x, z) \leq T, \quad x \ll z, \quad (C)$$

$$\rho - \sigma(t, \alpha) \geq 0, \quad t \in [0, T], \quad (D)$$

и доставляющие минимум величине ρ .

В этой задаче минимизируется величина постоянного ресурса при заданных технологических ограничениях (A) - (C) и заданном сроке T завершения всего процесса.

Рассмотрим еще одну задачу об оптимальном расписании. Пусть здесь также задан срок T завершения всего процесса. Для данного графика вычислим величину

$$\bar{\sigma}_0 = \frac{1}{T} \sum_{(x, y) \in U} s(x, y) \cdot \tau(x, y).$$

$\bar{\sigma}_0$ есть средняя интенсивность потребления ресурса в данном процессе. Сформулируем задачу в следующем виде:

Задача 3. Найти расписание $\{\alpha(x, y), (x, y) \in U\}$, удовлетворяющее ограничениям

$$\alpha(x, y) + \tau(x, y) \leq \alpha(y, v), \quad (x, y) \in U, v \gg y, \quad (A)$$

$$\alpha(a, x) \geq 0 \quad x \gg a, \quad (B)$$

$$\alpha(x, z) + \tau(x, z) \leq T, \quad x \ll z, \quad (C)$$

и доставляющее минимум величине

$$F(\alpha) = \frac{1}{T} \int_0^T [\sigma(t; \alpha) - \bar{\sigma}_0]^2 dt.$$

Задача 3 является задачей оптимизации расписания в смысле наименьшего среднеквадратического отклонения интенсивности потребления ресурса в графике от средней интенсивности.

Можно также рассматривать взвешенные среднеквадратические отклонения, вводя в функционал $F(\alpha)$ вес $g(t; \alpha)$:

$$F(\alpha) = \frac{1}{T} \int_0^T g(t; \alpha) [\sigma(t, \alpha) - \bar{\sigma}_0]^2 dt.$$

Весовая функция $g(t; \alpha)$ может быть, например, следующей:

$$g(t; \alpha) = \begin{cases} 1 & \text{при } \sigma(t; \alpha) > \sigma_0, \\ 0 & \text{при } \sigma(t; \alpha) < \sigma_0. \end{cases}$$

В этом случае имеем задачу о наименьшем среднеквадратическом превышении средней интенсивности потребления ресурса.

Можно получить различные варианты задачи 3, рассматривая вместо средней интенсивности σ_0 заданную функцию $R(t)$ наличия ресурса или выбирая другие целевые функции. Так, можно рассматривать функционалы вида

$$F_1(\alpha) = \int_0^T |\sigma(t; \alpha) - \sigma_0| dt$$

$$\text{или } F_2(\alpha) = \max_t (\sigma(t; \alpha) - \sigma_0) \quad \text{и т.д.}$$

Не останавливаясь подробнее на этих задачах, отметим только, что для их приближенного решения может использоваться метод, изложенный ниже (п.4) применительно к задаче 3. Этот же подход может использоваться также и при решении задачи с несколькими ресурсами.

3. Задачи 1 и 2 как задачи целочисленного линейного программирования

Рассмотрим дискретный вариант задач. Пусть $t(x, y)$ принимает целые значения, T - целое, и рассматриваются целочисленные расписания $\alpha(x, y)$. Промежуток $[0, T]$ разбивается целыми точками на интервалы $[k, k+1)$, $k = 0, 1, 2, \dots, T$. Тогда $\sigma(t; \alpha)$ при фиксированном α будет ступенчатой функцией времени с разрывами в целых точках,

$$\sigma(t; \alpha) = \sigma(k; \alpha) = \text{const} \quad \text{при } t \in [k, k+1).$$

Функция $R(t)$ предполагается также ступенчатой с разрывами в целых точках

$$R(t) = R_k = \text{const} \quad \text{при } t \in [k, k+1), \\ k = 0, 1, \dots, T-1.$$

Вместо расписания $\alpha(x, y)$ введем новые переменные $\xi_\kappa(x, y)$ и $\beta(x)$. Каждой дуге $(x, y) \in U$ отвечают переменные $\xi_\kappa(x, y)$, $\kappa = 0, 1, \dots, T-1$, принимающие значения 0 или 1.

$$\xi_\kappa(x, y) = \begin{cases} 1, & \text{если } \alpha(x, y) = \kappa, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Каждой вершине $x \in X$ отвечает переменная $\beta(x)$ - время наступления события x .

Для каждой работы (x, y) может быть вычислена величина $\delta_\kappa(x, y)$:

$$\delta_\kappa(x, y) = \sum_{r=\max(0, \kappa-\tau(x, y))}^{\kappa} \xi_r(x, y), \quad \kappa = 0, 1, \dots, T-1.$$

$\delta_\kappa(x, y) = 1$, если работа (x, y) выполняется на интервале $[\kappa, \kappa+1]$. В остальных случаях $\delta_\kappa(x, y) = 0$. Тогда суммарная интенсивность потребления ресурса на интервале $[\kappa, \kappa+1]$ может быть представлена в виде:

$$b(\kappa; \alpha) = \sum_{(x, y) \in U} s(x, y) \cdot \delta_\kappa(x, y).$$

Запишем дискретный вариант задач 1 и 2 в переменных $\xi_\kappa(x, y)$ и $\beta(x)$.

Задача 1⁰. Минимизировать $\beta(z)$ при ограничениях

$$\left. \begin{aligned} (a) \quad & \sum_{\kappa=0}^{T-1} \xi_\kappa(x, y) = 1, \\ (b) \quad & \sum_{\kappa=0}^{T-1} \kappa \xi_\kappa(x, y) > \beta(x), \\ (c) \quad & \sum_{\kappa=0}^{T-1} \kappa \xi_\kappa(x, y) + \tau(x, y) \leq \beta(y) \end{aligned} \right\} \text{ для всех } (x, y) \in U,$$

$$(d) \quad \sum_{(x, y) \in U} s(x, y) \left[\sum_{r=\max(0, \kappa-\tau(x, y))}^{\kappa} \xi_r(x, y) \right] \leq R_\kappa,$$

$$\kappa = 0, 1, \dots, T-1,$$

$$(e) \quad \beta(a) = 0, \quad \beta(z) \leq T,$$

относительно переменных $\xi_k(x, y) \geq 0$ (целых) и $\beta(x)$.
Задача 2⁰. Минимизировать ρ при ограничениях

$$\left. \begin{array}{l} (a) \\ (b) \\ (c) \end{array} \right\}$$

те же, что и в задаче I⁰.

$$(d) \sum_{(x,y) \in U} s(x,y) \left[\sum_{r=\max(0, k-\tau(x,y))}^{\xi_r(x,y)} \xi_r(x,y) \leq \rho, \right. \\ \left. k=0, 1, \dots, T-1, \right.$$

$$(e) \beta(a)=0, \beta(z)=T,$$

относительно переменных $\xi_k(x, y) \geq 0$ (целых), $\beta(x)$

и ρ .

Таким образом, рассматриваемые задачи о минимизации срока завершения процесса и о минимизации постоянного ресурса формулированы как задачи целочисленного линейного программирования. В этих задачах условия (а) обеспечивают обязательное и однократное выполнение каждой работы (x, y) в промежуток времени от 0 до T . Условия (б) и (с) являются технологическими ограничениями на последовательность работ. Условия (d) и (d') - ограничения на ресурсы. Условия (е) и (е') задают временной период выполнения всех работ графика. Отметим, что число переменных $\xi_k(x, y)$ в задачах I⁰ и 2⁰ можно значительно сократить. Из определения $\xi_k(x, y)$ следует, что $\xi_k(x, y) = 0$, если $k < A(x)$ или $k > B(y) - \tau(x, y)$, т.е. каждой дуге $(x, y) \in U$ отвечают лишь переменные $\xi_k(x, y)$ для $k \in [A(x), B(y) - \tau(x, y)]$. Однако и после такого сокращения эти задачи остаются задачами с большим числом переменных и в настоящее время не имеется удобных алгоритмов для точного решения таких задач целочисленного линейного программирования. Некоторые другие формализации этих задач и некоторые эвристические алгоритмы для их приближенного решения имеются в [1] и [5].

4. Дискретный вариант задачи 3
и приближенный метод ее решения

Рассмотрим дискретный вариант задачи 3. Пусть $\tau(x, y)$ принимает целые значения, T - целое и рассматриваются целочисленные расписания $\alpha = \{\alpha(x, y), (x, y) \in U\}$. Промежуток $[0, T]$ разбивается целыми точками на интервалы $[k, k+1), k = 0, 1, \dots, T-1$. Тогда $\sigma(t; \alpha)$ при фиксированном α будет ступенчатой функцией времени с разрывами в целых точках,

$$\sigma(t; \alpha) = \sigma(k; \alpha) = \text{const} \quad \text{при } t \in [k, k+1]$$

Дискретный вариант задачи 3 запишется так:

Задача 3⁰. Найти целочисленное расписание $\{\alpha(x, y), (x, y) \in U\}$, удовлетворяющее ограничениям:

$$\alpha(x, y) + \tau(x, y) \leq \alpha(y, v), \quad (x, y) \in U, \quad v > y, \quad (\text{A})$$

$$\alpha(a, x) \geq 0, \quad x \gg a, \quad (\text{B})$$

$$\alpha(x, z) + \tau(x, z) \leq T, \quad x \ll z \quad (T > T^*), \quad (\text{C})$$

и доставляющее минимум величине

$$F(\alpha) = \frac{1}{T} \sum_{k=0}^{T-1} [\sigma(k, \alpha) - \sigma_0]^2.$$

В случае задачи о наименьшем среднеквадратичном превышении средней интенсивности имеем функционал

$$F(\alpha) = \frac{1}{T} \sum_{k=0}^{T-1} g(k, \alpha) [\sigma(k; \alpha) - \sigma_0]^2,$$

где

$$g(k, \alpha) = \begin{cases} 1 & \text{при } \sigma(k, \alpha) > \sigma_0 \\ 0 & \text{при } \sigma(k, \alpha) \leq \sigma_0. \end{cases}$$

В силу целочисленности $\alpha(x, y)$ количество всех допустимых расписаний, удовлетворяющих условиям (А) - (С), конечно. Поэтому перебором всех допустимых решений принципиально можно получить оптимальное решение задачи Z^0 , оптимальное расписание. Однако хорошо известно, что полный перебор в настоящее время не является сколько-нибудь реальным путем решения таких задач.

Рассмотрим один специальный метод сокращенного перебора для приближенного решения задачи. Метод является эвристическим. В общих чертах это метод покоординатной оптимизации. При выборе очередной координаты расписания перебираются и оцениваются при помощи некоторой оценочной функции все допустимые значения данной координаты. Фиксируется то значение $\alpha(x, y)$, которое составляет локальный экстремум усредненной оценочной функции. Выбранное значение, которое условно может быть названо "оптимальным в среднем", обозначается $\alpha^*(x, y)$. На каждом шаге вычислений фиксируется значение $\alpha^*(x, y)$ для очередной дуги. С исчерпанием всех дуг процесс заканчивается. Полученное таким образом расписание $\alpha^* = \{ \alpha^*(x, y), (x, y) \in U \}$ принимается за приближенное решение задачи.

Основным моментом в рассматриваемом подходе является построение и использование усредненных оценочных функций, которые позволяют характеризовать каждое допустимое значение данной координаты с точки зрения ее выгодности для всего графика в целом.

Рассмотрим теперь один из возможных вариантов построения оценочной функции для решения задачи. Эта функция учитывает дефицитность ресурса во времени для данного графика. Достигается это за счет введения усредненного расписания $\bar{\alpha} = \{ \bar{\alpha}(x, y), (x, y) \in U \}$. Усредненное расписание вводится следующим образом. Считается, что для каждой дуги $(x, y) \in U$ величина $\bar{\alpha}(x, y)$ может принимать с равной вероятностью целые значения из промежутка $[A(x), B(y) - \tau(x, y)]$ независимо от других компонент расписания. В этих условиях для каждого целого $k \in [0, T]$ может быть подсчитана вероятность $p_k(x, y)$ выполнения работы (x, y) в момент времени k (точнее, на интервале $[k; k+1)$):

$$p_k(x, y) = \begin{cases} \frac{1}{q} & \text{при } k \in [A(x), B(y) - 1] \\ 0 & \text{в остальных случаях.} \end{cases}$$

Здесь r_k - число наложений дуги (x, y) на интервал $[k, k+1]$ при q различных допустимых положениях дуги.

$$q = B(y) - A(x) - \tau(x, y) + 1.$$

Пусть $h = \min(q, \tau(x, y))$. Тогда $r_{A(x)} = r_{B(y)-1} = 1$,
 $r_{k-1} = r_k + 1$ при $k \in [A(x), A(x) + h - 2]$, причем
 $r_{A(x)+i} = r_{B(y)-i-1}$ при $i = 0, 1, \dots, h-2$. Для остальных
 $k \in [A(x) + h - 1, B(x) - h - 2]$ $r_k = h = \text{const}$.

Поясним сказанное на примере. Пусть для дуги (x, y) известны величины $\tau(x, y) = 5$, $A(x) = 0$, $B(y) = 8$. На рисунке 2 изображены промежутки $[A(x), B(y)] = [0, 8]$ и допустимые положения дуги на нем.

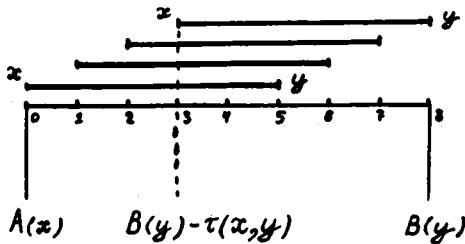


Рис. 2

Допустимыми расписаниями для дуги (x, y) являются целые значения из промежутка $[A(x), B(y) - \tau(x, y)] = [0, 3]$:

$$\alpha_1(x, y) = 0, \quad \alpha_2(x, y) = 1, \quad \alpha_3(x, y) = 2 \text{ и } \alpha_4(x, y) = 3.$$

Подсчитаем величины $\rho_k(x, y)$, $k = 0, 1, \dots, 7$.

$q = B(y) - A(x) - \tau(x, y) + 1 = 4$ - число различных допустимых положений дуги. r_k - число наложений дуги на интервал $[k, k+1]$

$$\rho_0 = \rho_1 = \frac{1}{4}, \quad \rho_2 = \rho_6 = \frac{3}{4} = \frac{1}{2}, \quad \rho_3 = \rho_5 = \frac{3}{4}, \quad \rho_4 = \rho_7 = \frac{4}{4} = 1.$$

На рисунке 3 приведен график $\rho_k(x, y)$.

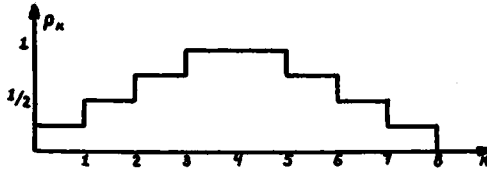


Рис. 3

Отметим, что для тех интервалов $[K, K+1]$, где работа (x, y) выполняется при любом допустимом расписании, $\rho_K = 1$. В нашем примере такими являются интервалы $[3, 4)$ и $[4, 5)$.

Итак, величина $\rho_K(x, y)$ есть вероятность выполнения работы (x, y) на интервале $[K, K+1]$ при усредненном расписании

$$\{\bar{\alpha}(x, y), (x, y) \in U\}.$$

Тогда величина

$$\bar{b}(K, \bar{\alpha}) = \sum_{(x, y) \in U} \rho_K(x, y) \cdot s(x, y)$$

есть математическое ожидание интенсивности суммарного потребления ресурса графиком на интервале $[K, K+1]$ при рассматриваемом усредненном расписании $\bar{\alpha}$. В частности, может быть вычислено значение минимизируемого функционала F при усредненном расписании $\bar{\alpha}$:

$$F(\bar{\alpha}) = \frac{1}{T} \sum_{K=0}^{T-1} [b(K; \bar{\alpha}) - b_0]^2.$$

$F(\bar{\alpha})$ и является той оценочной функцией, которую мы будем использовать при решении задачи 3⁰.

Проследим, как используется эта оценочная функция в одном конкретном алгоритме. Пусть (x_0, y_0) есть очередная дуга, для которой на данном шаге определяется и фиксируется значение $\alpha^*(x_0, y_0)$. На данном шаге дуга (x_0, y_0) является ведущей. Обозначим U^* множество

тех дуг из U , для которых к началу данного шага уже зафиксировано расписание α^* (На первом шаге U^* пусто). Все остальные дуги из U образуют множество V

$$V = (U \setminus U^*) \setminus (x_0, y).$$

Вычисления на каждом шаге начинаются с того, что для каждой дуги $(x, y) \in U \setminus U^*$ определяется промежуток изменения соответствующей компоненты $\alpha(x, y)$:

$$A(x) \leq \alpha(x, y) \leq B(y) - \tau(x, y), \quad (x, y) \in U \setminus U^*.$$

Величины $A(x)$ и $B(x)$ для $x \in X$ вычисляются при помощи алгоритма критического пути с учетом фиксированного расписания $\{\alpha(x, y), (x, y) \in U\}$. Алгоритм критического пути используется в этом методе многократно. Поэтому целесообразно предварительно упорядочить список дуг графика по классам так, чтобы вычисления выполнялись экономно, за один просмотр дуг графика. Тем более, что и для рассматриваемого эвристического алгоритма представляется естественным фиксировать расписание для дуг в порядке их следования. Итак, для определенности считаем, что дуги графика упорядочены по классам так, что к одному классу отнесены дуги с эквивалентными конечными вершинами. Отметим, что порядок перебора, вообще говоря, влияет на результаты вычислений. Поэтому возможны дальнейшие уточнения алгоритма, учитывающие те или иные эвристические соображения относительно порядка перебора.

Далее, значение $\alpha^*(x_0, y_0)$ для ведущей дуги выбирается в два этапа.

1 этап. На первом этапе исследуется влияние ведущей работы (x_0, y_0) на все последующие работы и график в целом. При этом выбирается "оптимальный в среднем" срок завершения этой работы.

Итак, рассмотрим последовательность расписаний

$$\tilde{\alpha}_i = \{\tilde{\alpha}_i(x, y), (x, y) \in U\}, \quad i = 1, 2, \dots, n,$$

и оценим каждое из этих расписаний. Расписание $\tilde{\alpha}_i$ строится следующим образом:

$$\tilde{\alpha}_i(x, y) = \begin{cases} \alpha^*(x, y) & \text{для } (x, y) \in U^* \\ \tilde{\alpha}_i(x, y) & \text{для } (x, y) \in U \setminus U^*. \end{cases}$$

Компоненты $\alpha^*(x, y)$ остаются постоянными во всех расписаниях $\bar{\alpha}_i$, $i = 1, 2, \dots, n$. Усредненная компонента $\bar{\alpha}_i(x, y)$ - ведущая компонента в расписании $\bar{\alpha}_i(x, y)$ - с равной вероятностью и независимо от других компонент принимает целые значения из промежутка $[A(x_0), C_i(y_0) - \tau(x_0, y_0)]$. $C_i(y_0)$ принимает целые значения из промежутка $[A(y_0), B(y_0)]$:

$$C_1(y_0) = A(y_0), C_2(y_0) = A(y_0) + 1, \dots,$$

$$C_n(y_0) = B(y_0).$$

Каждая усредненная компонента $\bar{\alpha}_i(x, y)$, $(x, y) \in U$ с равной вероятностью и независимо от

других компонент принимает целые значения из промежутка $[A_i(x_0), B(y) - \tau(x, y)]$. Для вершины y_0 считаем $A(y_0) = C_i(y_0)$. Границы $A_i(x)$ для $x > y_0$

изменяются при каждом изменении ведущей компоненты и перевычисляются при помощи алгоритма критического пути. Границы

$A_i(x)$ для $x > y_0$ остаются постоянными во всех расписаниях $\bar{\alpha}_i$, $i = 1, 2, \dots, n$.

$$A_i(x) = A(x) \quad \text{для } x \neq y_0.$$

Для каждого расписания $\bar{\alpha}_i$, $i = 1, 2, \dots, n$, вычисляется значение минимизируемого функционала $F(\bar{\alpha}_i)$:

$$F(\bar{\alpha}_i) = \frac{1}{T} \sum_{k=0}^{T-1} [\sigma(k; \bar{\alpha}_i) - \sigma_0]^2.$$

Здесь $\sigma(k; \bar{\alpha}_i) = \sigma_{u^*}(k; \alpha^*) + \sigma_{u \setminus u^*}(k; \bar{\alpha}_i)$.

Величины $\sigma(k; \alpha^*)$ и $\sigma(k; \bar{\alpha}_i)$ определены ранее, нижние индексы U и $U \setminus U^*$ означают, что эти величины относятся здесь, соответственно, к множествам U и $U \setminus U^*$.

Пусть m_i и $F(\bar{\alpha}_i) = F(\bar{\alpha}_i)$. В этом случае величина $C_{i_0}(y_0)$ является "оптимальным в среднем" сроком завершения ведущей работы (x_0, y_0) . Выбором $A_{i_0}(y_0) = C_{i_0}(y_0)$ и соответствующих значений $A_{i_0}(x)$ для $x > y_0$, фиксируется оптимальное усредненное расписание $\{\bar{\alpha}_{i_0}(x, y)$ для $(x, y) \in U\}$. На этом заканчивается первый этап вычислений, и переходим ко второму этапу.

П е т а п. На втором этапе вычислений для ведущей дуги

(x_0, y_0) выбирается и фиксируется значение $\alpha^*(x_0, y_0)$ из промежутка $[A(x_0), C_{i_0}(y_0) - \tau(x_0, y_0)]$, определенного на I этапе. Для этой цели рассматривается и оценивается последовательность расписаний $\alpha_j^0 = \{\alpha_j^0(x, y), (x, y) \in U\}$, $j = 1, 2, \dots, m$. Расписание α_j^0 имеет вид:

$$\alpha_j^0(x, y) = \begin{cases} \alpha^*(x, y) & \text{для } (x, y) \in U^*, \\ \alpha_j(x_0, y_0) & \text{для } (x_0, y_0), \\ \bar{\alpha}_{i_0}(x, y) & \text{для } (x, y) \in V. \end{cases}$$

Компоненты $\alpha^*(x, y)$ и $\bar{\alpha}_{i_0}(x, y)$ остаются постоянными для всех расписаний α_j^0 , $j = 1, 2, \dots, m$. Ведущая компонента $\alpha_j(x_0, y_0)$ принимает целые значения из промежутка

$$[A(x_0), C_{i_0}(y_0) - \tau(x_0, y_0)] : \\ \alpha_1(x_0, y_0) = A(x_0), \alpha_2(x_0, y_0) = A(x_0) + 1, \dots, \\ \alpha_m(x_0, y_0) = C_{i_0}(y_0) - \tau(x_0, y_0).$$

Для каждого расписания α_j^0 , $j = 1, 2, \dots, m$, вычисляется значение минимизируемого функционала $F(\alpha_j^0)$:

$$F(\alpha_j^0) = \frac{1}{T} \sum_{k=0}^{T-1} [\sigma(k; \alpha_j^0) - \sigma_0]^2.$$

Величина $\sigma(k; \alpha_j^0)$ состоит из трех слагаемых:

$$\sigma(k; \alpha_j^0) = \sigma_{u^*}(k; \alpha_j^0) + \bar{\sigma}_v(k; \bar{\alpha}_{i_0}) + \delta(k; j), \\ \delta(k; j) = \begin{cases} s(x_0, y_0) & \text{при } \alpha_j(x_0, y_0) \leq k \leq \alpha_j(x_0, y_0) + \tau(x_0, y_0) \\ 0 & \text{в остальных случаях.} \end{cases}$$

Составляющие $\sigma_{u^*}(k; \alpha)$ и $\bar{\sigma}_v(k; \bar{\alpha}_{i_0})$ постоянны для всех расписаний α_j^0 , $j = 1, \dots, m$, изменяется только величина $\delta(k, j)$.

Пусть $\min F(\alpha_j) = F(\alpha_{j_0}^0)$. В этом случае значение $\alpha_{j_0}^0(x_0, y_0)$ фиксируется для дуги (x_0, y_0) :

$$\alpha^*(x_0, y_0) = \alpha_{j_0}^0(x_0, y_0).$$

Таким образом, множество U^* пополняется дугой (x_0, y_0) и на этом заканчивается очередной шаг процесса.

После того, как расписание $\{\alpha^*(x, y), (x, y) \in U\}$ полностью построено, вычисления заканчиваются. Полученное расписание принимается в качестве приближенного решения задачи.

5. Запись алгоритма на АЛГОЛе.

Изложенный выше метод приближенного решения задачи о ресурсах был применен для решения ряда экспериментальных задач. Вычисления проводились на машине М-20 СО АН СССР. При записи алгоритма использован альфа-вариант языка АЛГОЛ, а программы получены при помощи альфа-транслятора [6].

Итак, рассматривается сетевой график, содержащий m событий и n работ. Массивы x, y, t, s содержат исходную информацию о графике и заполняются, соответственно, значениями величин:

- x - предшествующее событие,
- y - последующее событие,
- $t(x, y)$ - длительность работы (x, y) ,
- $s(x, y)$ - интенсивность потребления ресурса работой (x, y) .

Компоненты с одинаковым номером в этих массивах относятся к одной и той же работе. Срок завершения процесса - T . Задача решается для целых $T \in [T^*, T^* + E]$, E задано.

Ниже приводится рабочая схема алгоритма (альфа-программа); на рисунке 4 условно изображена структура схемы.

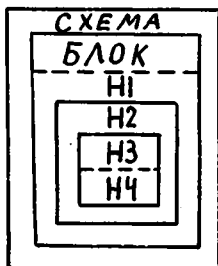


Рис.4

Схема состоит из 4 основных частей:

- H1 - обработка исходной информации и некоторые предварительные вычисления,
- H2 - описания процедур, используемых в схеме,
- H3 - вычисление "оптимального в среднем" расписания $\{\alpha^*(x, y), (x, y) \in U\}$, значения которого заполняют массив α ,
- H4 - уплотнение расписания *).

начало

СХЕМА:

целый m, n ; ввод (m, n) ;

БЛОК:

начало

целый $a, b, d, e, g, i, j, k, l, q, t, x, C, CO, E, T, \tau, \varphi$;

целый массив $x, y, t, \alpha[1:n]$,
 $A, B [1:m]$;

вещественный f, F, FO, p, sl, so, Y ;

вещественный массив $s[1:n]$;

H1 : ПРЕДВАРИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ:

начало

ввод (E, x, y, t, s) ;

Инициализация работ:

для $i := 1, \dots, m$ цикл $\{A[i] := 0; B[i] := 1\}$;

для $j := 1, \dots, n$ цикл $\alpha[j] := 0$;

Y1: просмотр списка:

$i := 0$; для $j := 1, \dots, n$ цикл
 $\{k := B[x[j]]$; если $B[y[j]] \leq k$, то

*) Отметим, что в некоторых случаях расписание, полученное основным алгоритмом, может быть улучшено за счет сдвига сроков отдельных работ в рамках этого расписания. Такое уплотнение расписания и предусматривается в приведенном здесь алгоритме.

$\{ B[y_{ij}] := k+1 ; i := 1 \}$;

если $i = I$ то на $Y1$;

вычисление размеров классов:

для $j := 1, \dots, n$ цикл

$\{ i := B[y_{ij}] ; A[i] := A[i] + 1 \}$;

для $i := 2, \dots, m$ цикл $A[i] := A[i-1] + A[i]$;

перестановка работ:

для $j := 1, \dots, n$ цикл начало

$Y2$: если $\alpha_{ij} = 1$ то на $Y3$;

$i := B[y_{ij}] - 1 ; k := A[i] + 1 ; a := x_{ij} ;$

$b := y_{ij} ; d := t_{ij} ; f := s_{ij} ;$

$x_{[j]} := x[k] ; y_{[j]} := y[k] ; t_{[j]} := t[k] ;$

$s_{[j]} := s[k] ; x[k] := a ; y[k] := b ;$

$t[k] := d ; s[k] := f ; \alpha[k] := 1 ;$

$A[i] := k ;$ на $Y2$;

$Y3$: конец ;

Определение общего объема ресурса, потребляемого графиком:

$Y := 0 ;$ для $j := 1, \dots, n$ цикл

$Y := Y + t_{[j]} \times s_{[j]} ;$

вывод $(x, y, t, s, Y) ;$

нахождение критического пути:

$T := 0 ;$ для $i := 1, \dots, m$ цикл $A[i] := B[i] := 0 ;$

для $j := n$ шаг - I до I цикл

$\{ k := B[y_{ij}] + t_{[j]} ; i := x_{ij} ;$

если $B[i] < k$ то $\{ B[i] := k ;$

если $T < k$ то $T := k \}$;

Определение срока завершения процесса:

$T := T + E ;$

Определение средней интенсивности:

$60 := Y / T ;$

Определение поздних сроков наступления событий:

для $i := 1, \dots, m$ цикл $B[i] := T - B[i]$
конец H1 ;

H2: ПРОЦЕДУРЫ:

начало

вещественный массив $B[1:T]$;

Примечание

В массиве B вычисляется суммарное потребление ресурса всеми работами графика на K -ом интервале при усредненном расписании. Последнее изменяется от шага к шагу. В конце ℓ -того шага первым ℓ работам отвечает фиксированное расписание, а остальным работам усредненное;

процедура выч A ;

Примечание

$A(x)$ - самый ранний допустимый срок наступления события x , $x \in X$. В процедуре перевычисляются значения $A(x)$ с учетом фиксированного расписания для первых $\ell - 1$ работ и фиксированного значения $A(y) = a$ для ведущей ℓ -той работы (x, y) , $i = y$;

начало

если $A[i] \geq a$ то { для $j := \ell, \dots, n$ цикл

$A[y[j]] := 0$; для $j := 1, \dots, \ell - 1$ цикл

{ $k := \alpha[j] + t[j]$; $q := y[j]$;

если $A[q] < k$ то $A[q] := k$ } ;

если $A[i] < a$ то $A[i] := a$;

для $j := \ell + 1, \dots, n$ цикл { $k := A[x[j]] + t[j]$;

$q := y[j]$; если $A[q] < k$ то $A[q] := k$ }

конец A ;

процедура выч $FCD(q, \varphi)$;

Примечание

Вычисляется значение функционала F на промежутке

$[q, \varphi] \in [1, T]$ для данного расписания. Одновременно определяется минимальное значение функционала FO для данного набора расписаний и соответствующее значение CO . При обращении к процедуре на этапе 1 CO - срок завершения ведущей работы, на этапе 2 и при уплотнении расписания CO - начало ведущей работы;

начало
 $F := 0$; для $k := q, \dots, \varphi$ цикл
 $\{ f := b[k] - CO; F := F + f \times f \}$;
 $F := F / T$; если $F < FO$ то
 $\{ FO := F; CO := C \}$ конец FO

процедура $выч \ b$;

Примечание

В массив $b[1:T]$ добавляется ($\varphi = +1$) или вычитается ($\varphi = -1$) усредненная компонента $p_k(x, y) \times s(x, y)$, отвечающая данной работе (x, y) .

j - порядковый номер данной работы (x, y) , $a = A(x)$,

$b = B(y) + 1$;

начало $a := a[j]$; $sl := s[j] \times \varphi$; $tl := t[j]$;

$q := b - a - tl$; если $q = 1$ то на L ;

$f := sl / q$; если $tl < q$ то

$\{ q := tl; sl := f \times q \}$; $p := 0$;

для $k := a + 1, \dots, a + q - 1$ цикл

$\{ p := p + f; b[k] := b[k] + p \}$;

для $k := b - q + 1, \dots, b - 1$ цикл

$\{ b[k] := b[k] + p; p := p - f \}$;

L : для $k := a + q, \dots, b - q$ цикл

$b[k] := b[k] + sl$

конец b ;

процедура $ок \ b(\tau)$;

Примечание

Вычисляется массив b при данном усредненном расписании, которое определяется сроком τ завершения ведущей работы

(x_0, y_0) , $i = y_0$. Вычисленные к моменту обращения к процедуре величины исправляются здесь за счет компонент, отвечающих работам вида (x, i) или работам (x, y) таким, что $x \succ i$. Работы (x, i) определяются условием $y[j] = i$. Работы (x, y) , $x \succ i$, определяются условием $\alpha[j] \neq A[x[j]]$;

начало
 $a := \tau$; выч A ; для $j := 1, \dots, n$ цикл
начало если $y[j] \neq 0$ то
 $\{ a := 0$; если $y[j] = i$ то
 $\{ e := \tau - B[i]$; на поправка $\}$;
если $\alpha[j] \neq A[x[j]]$ то
 поправка : $\{ v := B[y[j]] + 1$; $\varphi := -1$; выч b ;
 $v := v + e$; $\varphi := 1$; $\alpha[j] := A[x[j]]$; выч b $\}$
конец $B[i] := \tau$
конец $b(\tau)$;

НЗ: ВЫЧИСЛЕНИЕ РАСПИСАНИЯ:

начало
 Чистка полей:
для $k := 1, \dots, T$ цикл $b[k] := 0$;
 $v := 0$; $a := 0$; $i := 1$; $A[1] := -1$;
 Вычисление исходного усредненного расписания:
выч A ;
для $j := 1, \dots, n$ цикл
 $\{ \varphi := 1$; $v := B[y[j]] + 1$; $\alpha[j] := A[x[j]]$;
выч b $\}$;
Примечание

ℓ - порядковый номер ведущей работы (x, y) . На этапе 1 определяется "оптимальный в среднем" срок CO наступления события y и вычисляется b для соответствующего усредненного расписания. На этапе 2 вычисляется компонента $\alpha[\ell]$ "оптимального в среднем" расписания, отвечающая ведущей работе;

Этап 1:

$l := l + 1$; если $l < n$ то на $\Omega 1$; если $t[l] = 0$
то на Этап 1; $l := y[l]$; $d := A[l]$; $g := B[l]$;
 $FO := 10118$;

для $C := d, \dots, g$ цикл

$\{cz \sigma(C); \text{выч } FCO(1, T)\}$; $cz \sigma(CO)$;

Исключение ведущей работы из усредненного расписания:

$\varphi := -1$; $v := CO + 1$; $j := l$; выч σ ;

Этап 2:

$tl := t[l]$; $sl := s[l]$; $a := A[x[l]] + 1$;

$b := CO - tl$; $e := CO$;

Начальное размещение ведущей работы:

$C := a - 1$; для $k := a, \dots, a + tl - 1$

цикл $\sigma[k] := \sigma[k] + sl$; $FO := 10118$;

выч $FCO(a, e)$;

Сдвиг ведущей работы:

для $C := a, \dots, b$ цикл $\{\sigma[C] := \sigma[C] - sl$;

$\sigma[C + tl] := \sigma[C + tl] + sl$; выч $FCO(a, e)\}$;

Закрепление начала ведущей работы:

$\alpha[l] := CO$; $a := CO + tl$; выч A ;

для $k := 1, \dots, tl$ цикл

$\{\sigma[CO + k] := \sigma[CO + k] + sl$;

$\sigma[v + k] := \sigma[v + k] - sl\}$;

Подготовка следующего шага:

$B[i] := g$; для $j := l + 1, \dots, n$ цикл

начало если $t[j] \neq 0$ то

$\{\text{если } y[j] = i$ то

$\{v := e + 1$; $\varphi := -1$; выч σ ; $v := g + 1$;

$\varphi := 1;$ выч $\sigma \}$ конец ;

на Этап I ;

$\Omega 1$: вывод результата:

выч $FCO(1, T);$ вывод $(T, \sigma_0, \sigma, \alpha, F);$

$H4$: УПЛОТНЕНИЕ РАСПИСАНИЯ:

вычисление границ:

для $j := 1, \dots, n$ цикл $\{ i := x[j];$
если $B[i] > \alpha[j]$ то $B[i] := \alpha[j]; \}$

изменение расписания:

для $l := 1, \dots, n$ цикл если $t[l] \neq 0$ то

начало

$tl := t[l];$ $sl := s[l];$ $a := \alpha[l];$

$i := A[x[l]]; j := B[y[l]]; b := j - tl;$

Начальное размещение работ:

если $a \neq i$ то $\{$ для $k := 1, \dots, tl$ цикл
 $\{ \sigma[a+k] := \sigma[a+k] - sl;$
 $\sigma[i+k] := \sigma[i+k] + sl \}; \}$

выбор α :

$FO := 10118;$ $C := i;$ выч $FCO(i, j);$

для $C := i+1, \dots, b$ цикл
 $\{ \sigma[C] := \sigma[C] - sl;$
 $\sigma[C+tl] := \sigma[C+tl] + sl; \}$ выч $FCO(i, j);$

$\alpha[l] := CO;$

для $k := 1, \dots, tl$ цикл
 $\{ \sigma[CO+k] := \sigma[CO+k] + sl; \sigma[b+k] := \sigma[b+k] - sl \};$

если $CO < a$ то $\{ A[y[l]] := 0;$

для $j := 1, \dots, n$ цикл $\{$ если $\alpha[j] + t[j]$
 $< A[x[j]]$ то $A[y[j]] := \alpha[j] + t[j] \}$

иначе если $CO + tE > A[y[E]]$ то
 $A[y[E]] := CO + tE$
конец изменения расписания;

выч $FCO(1, T)$; вывод (σ, α, F) ;

если $E > 0$ то $\{E := E - 1;$ на
 нахождение критического пути }

конец НЗ и НЧ ;

конец Н2 ;

конец Блока ;

конец Схемы ; *

6. Пример .

Пусть решается задача о ресурсах (задача 3^0) для сетевого графика, заданного таблицей I и изображенного на рисунке 5^x). Оптимальное расписание (приближенное) строится описанным выше алгоритмом. Проследим за вычислениями, при этом не будем останавливаться на вычислениях, встречающихся в других алгоритмах (упорядочение дуг графа, нахождение критического пути и т.п.), и подробнее проиллюстрируем работу собственно алгоритма "оптимизации в среднем".

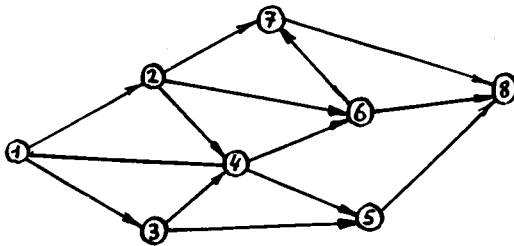


Рис. 5

x) Этот график имеется в [1].

В результате работы оператора $H1$ (предварительные вычисления) определяются длина критического пути $T^* = 15$ и величины $A(x)$ и $B(x)$ для всех $x \in X$, приведенные в таблице Ш. $A(x)$ - самый ранний допустимый срок наступления события

x , $B(x)$ - самый поздний срок. Таблица I не изменяется, т.к. исходные данные уже упорядочены нужным образом. Пусть общий срок завершения всего процесса $T = 22$ ($E = 7$). В этом случае средняя интенсивность потребления ресурса графиком

$$\sigma_0 = \frac{1}{T} \sum_{j \in J} j \tau = \frac{257}{22} = 11,68. \text{ Величины } A(x) \text{ и } B(x), x \in X,$$

полностью определяют усредненное расписание $\{\bar{\alpha}(x, y),$

$(x, y) \in U\}$ и функцию $\bar{\sigma}(k; \bar{\alpha})$ - ожидаемое потребление ресурса графиком на интервале $[k, k+1)$, $k = 0, 1, \dots, T-1$.

Рассмотрим, например, работу (2,6): $J(2,6) = 6$, $\tau(2,6) = 3$,

$A(2) = 2$, $B(6) = 8 + 7 = 15$ (таблицы I и Ш). Допустимые

сроки выполнения работы (2,6) изображены на рисунке 6. Имеется 8 вариантов размещения работы (2,6) на промежутке $[2, 15]$.

Величины $P_k(2,6)$ суть вероятности выполнения работы (2,6) на интервале $[k, k+1)$ и имеют следующие значения: $P_k(2,6) =$

$= 0$ при $k < 2$ или $k > 14$, $P_2(2,6) = P_{14}(2,6) = 1/8$,

$P_3 = P_{13} = 2/8 = 1/4$, $P_4 = P_{12} = 3/8$, $P_5 = P_{11} = 4/8 = 1/2$,

$P_6 = P_{10} = 5/8$, $P_7 = P_9 = P_8 = 6/8 = 3/4$.

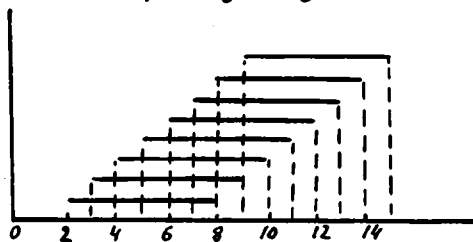


Рис. 6

ожидаемое потребление ресурса работой (x, y) на интервале $[k, k+1)$ выражается величиной $P_k(x, y) \tau(x, y)$ и подсчитывается процедурой "выч σ ". График величины

$P_k(2,6) \tau(2,6)$ приведен на рисунке 7.

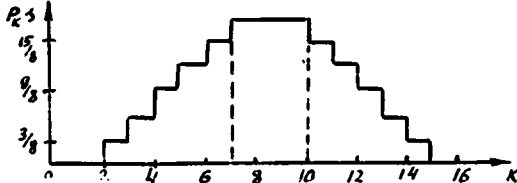


Рис. 7

Аналогично вычисляются величины $P_k(x,y) \delta(x,y)$ для других дуг. Значения функций

$$\sigma(k; \bar{\alpha}) = \sum_{(x,y) \in U} P_k(x,y) \delta(x,y), \quad k=0, 1, \dots, T-1$$

накапливаются в массиве σ при обращении к процедуре "выч σ ". График функции $\bar{\sigma}(k, \bar{\alpha})$ для исходного усредненного расписания приведен на рисунке 8.

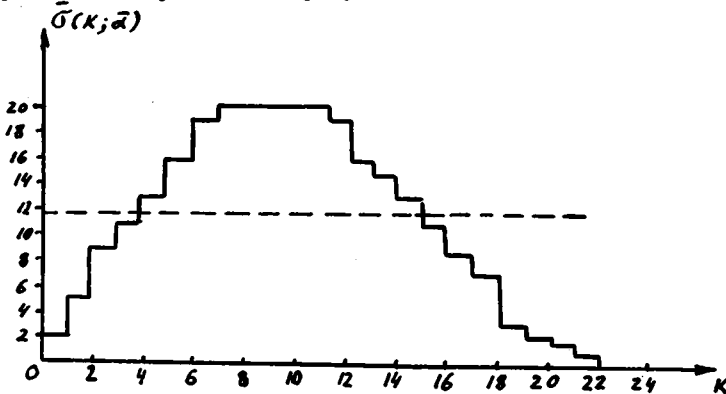


Рис. 8

Пусть первые три координаты расписания α^* вычислены, их значения приведены в таблице П. Рассмотрим подробнее вычисление следующей координаты расписания - значения $\alpha^*(3,4)$. На этом шаге работа (3,4) является ведущей, $(x_0, y_0) = (3, 4)$, множества U^* и V указаны в таблице П.

Первый этап данного шага вычислений - определение "оптимального в среднем" срока C наступления события $y_0 = 4$.

$C \in [A(y_0), B(y_0)] = [5, 12]$ (из таблицы Ш). Таким образом, C может принимать 8 различных значений: $C_1 = 5, C_2 = 6, C_3 = 7, \dots, C_8 = 12$. Каждому значению C_i отвечают перевычисленные значения $A_i(x), x \in X$ и соответствующее усредненное расписание $\bar{\alpha}_i, i = 1, 2, \dots, 8$. Значения $A_i(x)$ перевычисляются процедурой "выч А" с учетом уже зафиксированного расписания $\{\alpha^*(x,y), (x,y) \in U^*\}$ и выбранного значения C_i . При этом значения $A_i(x)$ изменяются только для $x > y_0$. Эти вычисления отражены в таблице

IV, $A_i(4) = B_i(4) = C_i$. На этапе I для каждого усредненного расписания

$$\tilde{\alpha}_i = \{ \alpha^*(x, y), (x, y) \in U^* \} \cup \{ \bar{\alpha}_i(x, y), (x, y) \in U \setminus U^* \}$$

вычисляются значения функции $\sigma(K; \tilde{\alpha}_i)$, $K = 0, 1, \dots, 2I$, и значение функционала $F(\tilde{\alpha}_i)$.

$$\sigma(K; \tilde{\alpha}_i) = \sigma_{U^*}(K; \alpha^*) + \sigma_{U \setminus U^*}(K; \bar{\alpha}_i).$$

Графики функций $\sigma(K; \tilde{\alpha}_i)$, $i = 1, 2, \dots, 8$, представлены на рисунке 9, соответствующие значения функционала $F(\tilde{\alpha}_i)$ даны в таблице V. Как показывает эта таблица, лучшим сроком наступления события $Y_0 = 4$ является $C = 12$

$$\min_i F(\tilde{\alpha}_i) = F(\tilde{\alpha}_2) = 8,72$$

Фиксируем соответствующее этому сроку расписание $\{ \bar{\alpha}(x, y), (x, y) \in U \}$ и переходим ко второму этапу.

Второй этап вычислений - определение "оптимального в среднем" начала $\alpha^*(3, 4)$ для ведущей работы. Здесь считается, что ведущая работа $(x_0, y_0) = (3, 4)$ может выполняться в период от $A(x)$ до C , т.е. от 3 до 12 (табл. IV). Считая зафиксированными расписание

$$\{ \alpha^*(x, y), (x, y) \in U^* \} \text{ и } \{ \bar{\alpha}(x, y), (x, y) \in U \},$$

перебираем и оцениваем все допустимые значения $\alpha(3, 4)$:

$\alpha_1 = 3$, $\alpha_2 = 4, \dots, \alpha_3 = 10$. График функции $\sigma_{U^*}(K; \alpha^*) + \sigma_U(K; \bar{\alpha})$ представлен на рисунке 10 (отметим, что здесь отсутствует слагаемое, отвечающее ведущей работе). Значения функционала даны в таблице VI. Из этой таблицы следует, что "оптимальным в среднем" началом ведущей работы является $\alpha_3 = 5$. Фиксируем это значение для работы $(3, 4)$: $\alpha^*(3, 4) = 5$. Работа $(3, 4)$ включается в множество U^* . На этом данный шаг вычислений закончен. На следующем шаге ведущей является работа $(2, 4)$ - очередная работа основного списка (таблица I).

Окончательный результат представлен на рисунке II и в таблице VII, $F(\alpha^*) = 3,48$. в этом случае полученное расписание не улучшается оператором H_4 .

Таблица I

| i | (x, y) | | $\mathcal{L}(x, y)$ | | $\mathcal{S}(x, y)$ | |
|-----|----------|-----|---------------------|-----------|---------------------|-----------|
| | x | y | \bar{x} | \bar{y} | \bar{z} | \bar{s} |
| I | I | 2 | 2 | 7 | | |
| 2 | I | 3 | I | 4 | | |
| 3 | I | 4 | 5 | 6 | | |
| 4 | 3 | 4 | 2 | 8 | | |
| 5 | 2 | 4 | 2 | 5 | | |
| 6 | 3 | 5 | 7 | 6 | | |
| 7 | 4 | 5 | 6 | 5 | | |
| 8 | 2 | 6 | 6 | 3 | | |
| 9 | 4 | 6 | 2 | 7 | | |
| 10 | 2 | 7 | 5 | 5 | | |
| 11 | 6 | 7 | 2 | 6 | | |
| 12 | 7 | 8 | 4 | 4 | | |
| 13 | 6 | 8 | 7 | 2 | | |
| 14 | 5 | 8 | 4 | 3 | | |

Таблица II

| | | | $\alpha^*(x, y)$ | |
|-----------------|----------|-----|------------------|--|
| | x | y | α | |
| \mathcal{U}^* | I | 2 | 0 | |
| | I | 3 | 2 | |
| | I | 4 | 0 | |
| \mathcal{V} | (x, y) | 3 | 4 | |
| | | 2 | 4 | |
| | | 3 | 5 | |
| | | 4 | 5 | |
| | | 2 | 6 | |
| | | 4 | 6 | |
| | | 2 | 7 | |
| | | 6 | 7 | |
| | | 7 | 8 | |
| | | 6 | 8 | |
| | | 5 | 8 | |

Таблица III

| x | $A(x)$ | $B(x)$ |
|-----|--------|--------|
| I | 0 | 0 |
| 2 | 2 | 2 |
| 3 | I | 3 |
| 4 | 5 | 5 |
| 5 | II | II |
| 6 | 8 | 8 |
| 7 | 10 | II |
| 8 | 15 | 15 |

Таблица IV

| x | $A_1(x)$ | $A_2(x)$ | $A_3(x)$ | $A_4(x)$ | $A_5(x)$ | $A_6(x)$ | $A_7(x)$ | $A_8(x)$ | $x > y$ |
|-----|----------|----------|----------|----------|----------|----------|----------|----------|---------|
| I | 0 | | | | | | | | |
| 2 | 2 | | | | | | | | |
| 3 | 3 | | | | | | | | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
| 5 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 5 |
| 6 | 8 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 6 |
| 7 | 10 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 7 |
| 8 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 8 |

Таблица У

| i | C_i | $F(\alpha_i)$ |
|-----|-------|---------------|
| I | 5 | 31,41 |
| 2 | 6 | 27,69 |
| 3 | 7 | 22,66 |
| 4 | 8 | 19,38 |
| 5 | 9 | 16,55 |
| 6 | 10 | 13,82 |
| 7 | 11 | 11,20 |
| 8 | 12 | 8,72 |

Таблица УІ

| j | $\alpha_j(\beta, \gamma)$ | $F(\alpha_j)$ |
|-----|---------------------------|---------------|
| I | 3 | 7,09 |
| 2 | 4 | 4,84 |
| 3 | 5 | 2,60 |
| 4 | 6 | 4,42 |
| 5 | 7 | 5,66 |
| 6 | 8 | 6,63 |
| 7 | 9 | 6,84 |
| 8 | 10 | 5,90 |

Таблица УІІ

| x | y | $\alpha(x, y)$ |
|-----|-----|----------------|
| I | 2 | 0 |
| I | 3 | 2 |
| I | 4 | 0 |
| 3 | 4 | 5 |
| 2 | 4 | 7 |
| 3 | 5 | 3 |
| 4 | 5 | 12 |
| 2 | 6 | 9 |
| 4 | 6 | 10 |
| 2 | 7 | 11 |
| 6 | 7 | 16 |
| 7 | 8 | 18 |
| 6 | 8 | 15 |
| 5 | 8 | 18 |

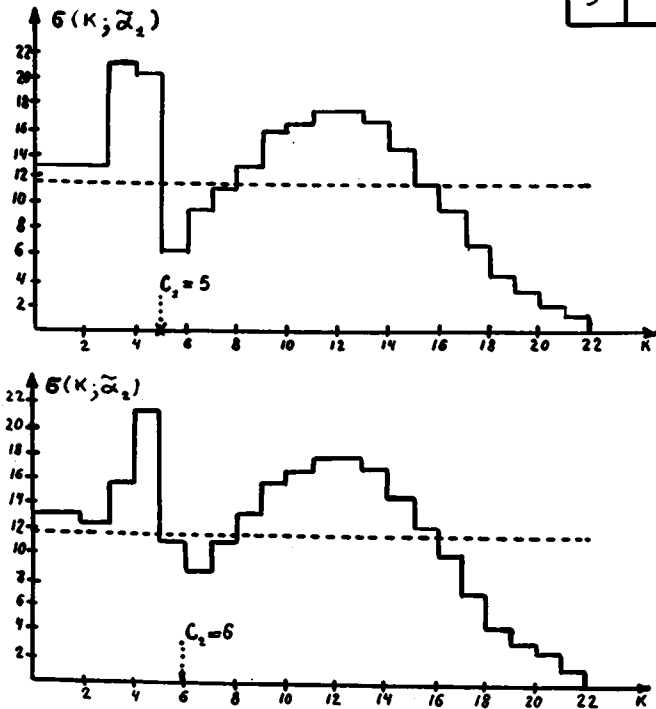


Рис. 9

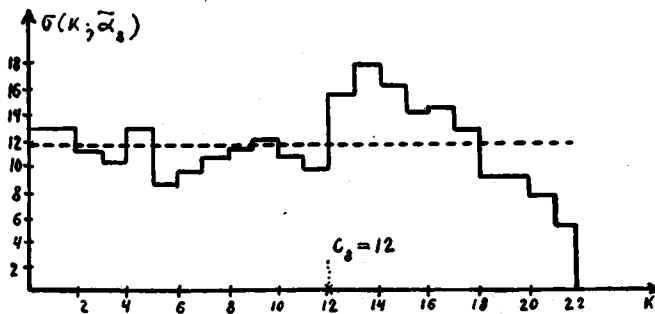
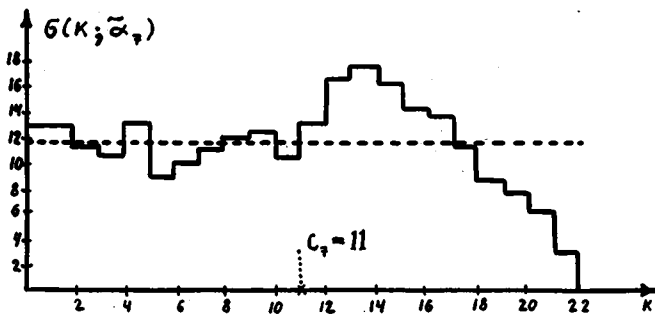
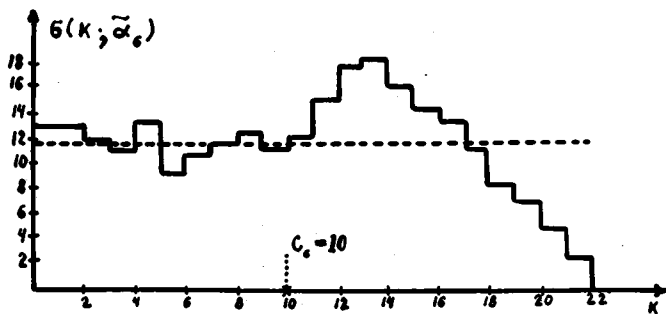


Рис. 9а

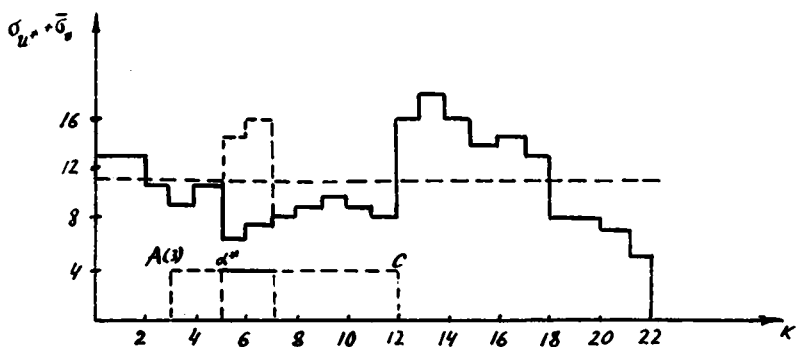


Рис. 10.

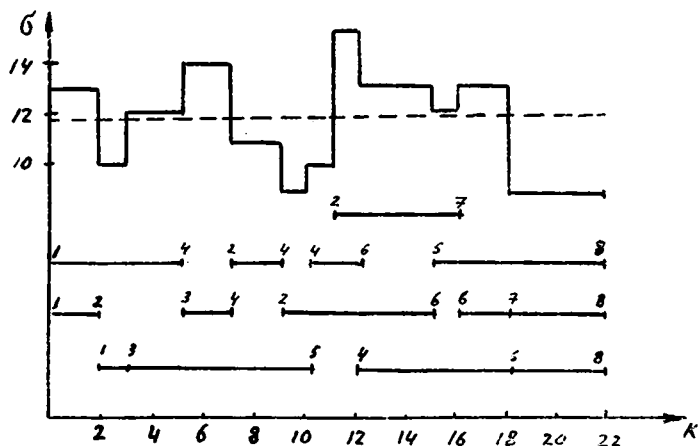


Рис. 11

Для иллюстрации работы алгоритма в различных случаях и выяснения степени его эффективности проводились некоторые экспериментальные вычисления. Так, для описанного примера были получены расписания на II промежутках $[0, T]$, $T = 15, 16, \dots, 25$, при трех различных упорядочениях исходного списка работ. На рисунке 12 приведен график зависимости значений функционала $F(\alpha^*)$ от длины промежутка T для описанного в при-

мере упорядочения исходного списка работ. Для двух других упорядочений значения $F(\alpha^*)$ отмечены на том же рисунке (точками для одного и звездочками - для другого).

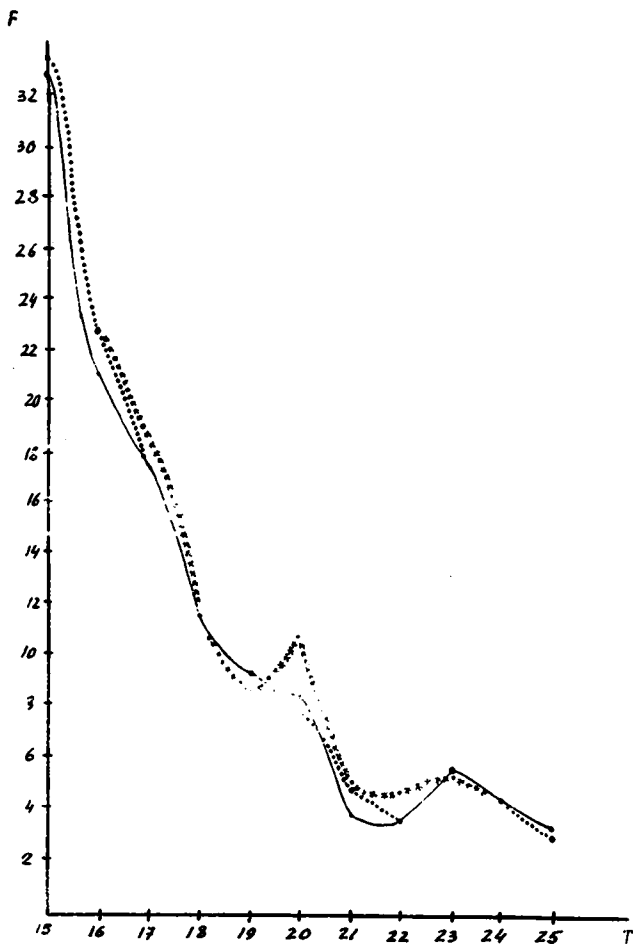


Рис. 12

ЛИТЕРАТУРА

1. С. И. Зуховицкий, И. А. Радчик. Математические методы сетевого планирования. Наука, 1965.
2. Д. Гейл. Теория линейных экономических моделей. ИЛ, 1963.
3. Л. Т. Петрова. Нахождение критического пути. Сб. Математическое программирование. Москва, 1966.
4. В. Ф. Фейлов. Некоторые программы для анализа сетевых графиков. Оптимальное планирование, вып.4, Изд-во "Наука", Сибирское отделение, Новосибирск, 1966.
5. Л. Я. Лейфман. В сб. Научные труды НГУ. Серия экономическая. Вып. 8. Математические методы решения экономических задач. Новосибирск, 1966.
6. Альфа-язык. Руководство для составления программ. Отчет ВЦ СО АН СССР, рот., Новосибирск, 1966.