Оглавление

Глава 1

Графы

1.1 Сетевой график

1.1.1 Нахождение резервов работ

Задан сетевой график $G = \langle M, N \rangle$, работы – дуги, $u \in N, t(u) \geq 0$
Знаем $t_{\text{кр}}$ – критическое время
$v[i]$ – времена наступления событий (закончены все работы, которые туда входят, и можно приступать к любому из тех, которое выходит)

Замечание. В форме работы – вершины, $v[i]$ – самое раннее возможное время входа в эту вершину (начала работы)

$w[i]$ – самое позднее время наступления события i

Алгоритм (определения $w[i]$).

1. Топологическая сортировка, $w[i] := t_{\text{кр}} \quad \forall i$
2. **for** $i := m$; i-- *to 1* **do**
   - **for** $u \in N_i^+$ **do**
     - **if** $w[\text{beg}(u)] > w[i] - t[u]$ **then**
       - $w[\text{beg}(u)] := w[i] - t[u]$
   - **end**

   end

Задача. Есть задачи, которые выполняются на m параллельных процессорах
Задан сетевой график в форме работы – вершины
Для каждой вершины задано время работы $t[i] > 0$
Для каждой дуги задано $c(i, j)$ – время передачи данных от i к j, если i и j выполняются на разных процессорах

1.2 Паросочетания

Определение 1. Паросочетание в графе – набор дуг, не имеющих общих начал и концов

Задача. Найти паросочетание наибольшего размера

Определение 2. Вершинная база графа – подмножество вершин, которым инцидентны все другие

Утверждение 1. Размер максимального паросочетания равен размеру минимальной вершинной базы

Наша задача эквивалентна этой:

Задача. Найти минимальную вершинную базу

Мы будем решать задачу о паросочетании, тем самым решим и задачу о вершинной базе
Задачу о паросочетании будем рассматривать на двудольном графе

Определение 3. Двудольный граф $G = \langle M_1 \cup M_2, N \rangle : u = (i, j) \in N \quad i \in M_1,\ j \in M_2$, т. е. вершины рабиты на два множества, и все рёбра идут из одного множества в другое

Алгоритм (построения максимального паросочетания). Дан двудольный граф $G = \langle M_1 \cup M_2, N \rangle$
$\overline{N}$ – паросочетание, $\overline{N} \neq \emptyset$
$X(\overline{N}) \subset M_1, Y(\overline{N}) \subset M_2$ – вершины из M_1 и M_2 соответственно, которые **не** покрываются паросочетанием

Начальное состояние. $X = M_1, \quad Y = M_2$

Считаем, что все дуги идут из M_1 в M_2

1. Дуги из $\overline{N}$ перенаправляем, чтобы они шли из M_2 в M_1
2. Ищем путь из $X(\overline{N})$ в $Y(\overline{N})$
3. Если пути нет, алгоритм заканчивается
4. $P = u_1 v_1 u_2 v_2 ... u_k$, где $\begin{cases} u_i \notin \overline{N} \\ v_j \in \overline{N} \end{cases}$ (последняя обязательно u)
5. $\overline{N} \coloneqq \left(\overline{N} \setminus \{\, v_j \mid v_j \in P \,\}\right) \cup \{\, u_i \mid u_i \in P \,\}$
6. Все дуги идут из M_1 в M_2

Алгоритм (нахождения минимального вершинного покрытия).

1.