Оглавление

Глава 1

Графы

1.1 Критический путь в сетевом графике

Дан сетевой график $G = \langle M, N \rangle$. Для каждой дуги задана $t(u) \geq 0$. N – работы
Максимальный путь показывает **минимальное** время работ (если максимально распараллелить)
Можно свести задачу к алгоритму Дейкстры, если все веса домножить на -1. Но мы так делать не будем

Алгоритм.

1. Топологическая сортировка
2. $\forall i \in M \quad v[i] := 0$
3. **for** $i := 1$ *to* w **do**
       for $u \in N_i^-$ **do**
           if $v[end(u)] < v[i] + t[u]$ **then**
               $v[\mathrm{end}(u)] := v[i] + t[u]$
       end
   end

   На выходе получаем вектор v, где $v[i]$ – максимальная длина пути из i_0 (начальная фиктивная вершина) в i. Тогда $v[i_+]$ (конечная фиктивная вершина) – длина критического пути

Пусть теперь задано W – отношение частичного порядка
N – множество работ

Алгоритм.

1. Строим транзитивное замыкание W
2. $u \in N \to \pi_i, \qquad i \in 1 : p$
3. У нас есть разбиения $D_i = \{\, \pi_i, N \setminus \pi_i \,\}$
4. Строим произведение разбиений D_i. Обозначим его $D := \bigcup_j \rho_j$
5. Строим граф $G = \langle M_1 \cup M_2, N \cup F \rangle$, где:
   - Каждое π_i даст нам вершину из M_1
   - Каждое ρ_j даст нам вершину из M_2
   - N – наши изначальные работы
   - F – фиктивные работы

   Процесс построения графа:

   (a) $u = (\mathrm{beg}\, u, \mathrm{end}\, u)$, где:
      - $\mathrm{beg}\, u = \pi_i(u)$
      - $\mathrm{end}\, u = \rho_j, \quad u \in \rho_j$

(b) Добавляем фиктивные дуги F, которые будут вести из M_2 в M_1
Каждая дуга будет $\rho_j \to \pi_i$ так, что $\pi_i = \bigcap \rho_j$

(c) Удаляем лишние фиктивные дуги по двум правилам:

- Если фиктивная дуга соединяет две вершины, между которыми есть другой путь
- Если из какой-то вершины выходит ровно одна фиктивная дуга, то можно склеить эти вершины
  Если входит ровно одна – аналогично