

加工顺序问题

“加工顺序问题”又被称为“批处理作业调度问题”。

设有 n 个工件需要在机器 M_1 和 M_2 上加工，每个工件的加工顺序都是先在 M_1 上加工，然后在 M_2 上加工。 t_{1j} , t_{2j} 分别表示工件 j 在 M_1 , M_2 上所需的加工时间 ($j=1, 2, \dots, n$)。问应如何在两机器上安排生产，使得第一个工件从在 M_1 上加工开始到最后一个工件在 M_2 上加工完所需的总加工时间最短？关于此（类）问题的回溯法求解被作为经典案例在很多教材或参考文献中出现，现要求设计求解此问题的动态规划算法。

提示（分析最优解的性质，刻画最优解的结构—最优子结构性质）

将 n 个工件的集合看做 $N=\{1, 2, \dots, n\}$ ，设 P 是给定 n 个工件的一个最优加工顺序方案，则 $P(i)$ 是该调度方案的第 i 个要调度的工件 ($i=1, 2, \dots, n$)。先考虑初始状态，第一台机器 M_1 开始加工集合 N 中的 $P(1)$ 工件时，第二台机器 M_2 空闲。随着时间的推移，经过 $t_{1P(1)}$ 的时间，进入一个新的状态：第一台机器 M_1 开始加工集合 $N-\{P(1)\}$ 中的 $P(2)$ 工件时，第二台机器 M_2 开始加工 $P(1)$ 工件，需要 $t_{2P(1)}$ 的时间才能空闲。以此类推，可以将每一个状态表示成更一般的形式，即：当第一台机器 M_1 开始加工集合 S ($S \subseteq N$ 是 N 的作业子集) 中的工件 i 时，第二台机器 M_2 需要 t 时间才能空闲下来。这种状态下，从集合 S 中的第一个工件开始在机器 M_1 上加工到最后一个工件在机器 M_2 上加工结束时所耗的时间为 $T(S, t)$ 。设集合 S 的最优加工顺序中第一个要加工的工件为 i ，那么，经过 t_{1i} 的时间，进入的状态为第一台机器 M_1 开始加工集合 $S-\{i\}$ 中的工件时，第二台机器 M_2 需要 t' 时间才能空闲下来，这种情况下机器 M_2 加工完 $S-\{i\}$ 中的工件所需的时间为 $T(S-\{i\}, t')$ ，其中

$$t' = \begin{cases} t_{2i} + t - t_{1i} & t > t_{1i}, \text{ 即 } t' = t_{2i} + \max\{t - t_{1i}, 0\} \\ t_{2i} & t \leq t_{1i} \end{cases}$$

$$\text{则 } T(S, t) = t_{1i} + T(S-\{i\}, t_{2i} + \max\{t - t_{1i}, 0\}) \quad (2-1)$$

从式 (1-1) 可以看出，如果 $T(S, t)$ 是最小的，那么其包含的子问题 $T(S-\{i\}, t_{2i} + \max\{t - t_{1i}, 0\})$ 肯定也是最小的。即该问题的整体最优一定包含其子问题的最优。