

2.7 最终期限排定问题(DEADLINE)

最终期限排定问题(见[22,206-212页]和[10,399-401页])是指在一个“单位时间作业”的集合中选择一个最优的子集,让该子集中作业在一个处理器上工作,每个作业都有一个特定的终止时间和收益,这里的收益只有该作业在终止时间前完成才可获得。故最优子集就是使收益最大的作业集合。单位时间的意思是说该作业会在开始时刻的一个单位之间之后完成。这个排定问题可以使用贪心算法解决,但是这里我们告诉你它还可以用和其他排定问题类似的DP方法来解决。状态为 (k, S) , k 表示阶段数, S 表示尚未考虑的作业的集合。决策 d 是集合 S 中的一个元素。它的下一个状态是 $(k + 1, S - \{d\})$ 。选择作业 d 的花费究竟是它的收益还是0,这要看它是否在“可行”集合中(注:该集合中每个作业都能赶上终止时间)。如果作业的集合是按终止时间递增给出的,这个“可行性”的判断会更简单些。假定它们是按照终止时间排序的,那么选择子集中的作业会依次连续执行;由于我们假定是单位时间作业,排在第 j 位的作业会在时刻 j 终止。因此,对于大小为 k 的作业序列 $S = 1, \dots, k$,如果对 $1 \leq j \leq k$ 均有 $j \leq t_j$,则该集合是可行的,即 S 中的每个作业 j 都能在终止时间 t_j 之前完成。这个最优化问题可以使用下述状态转移方程来解决:

$$f(k, S) = \max_{d \in S} \{c(d|S) + f(k + 1, S - \{d\})\},$$

这里如果作业 d 在可行集合中则 $c(d|S) = w_d$, 否则 $c(d|S) = 0$ 。我们的目标是解决 $f(1, S^*)$, 边界情况是当 $k = N + 1$ 或 $S = \emptyset$ 时有 $f(k, S) = 0$ 。假定工作的集合 $S^* = \{0, 1, 2, 3, 4\}$ 对应的收益是 $p = \{10, 15, 20, 1, 5\}$, 终止时间是 $t = \{1, 2, 2, 3, 3\}$ 。那么 $f(1, \{0, 1, 2, 3, 4\}) = 15 + 20 + 5 + 0 + 0 = 40$, 其对应的最优决策序列为: $d_1 = 1, d_2 = 2, d_3 = 4, d_4 = 0, d_5 = 3$ 。