2.12 流水线问题(FLOWSHOP)

流水线问题(见[22,301-306页]和[4,142-145页])是一个工件排定问题，这里对于每个工件i都有两道工序，A和B，且工序A必须在工序B之前完成。这些工序分别在不同的机器上处理。工件的序列应该满足各工序的先后次序。例如，假定工序A的执行时间$p_i = \{3, 4, 8, 10\}$,工序B的执行时间$q_i = \{6, 2, 9, 15\}$，这里$i = \{0, 1, 2, 3\}$。如果各工件按照给定顺序处理，那么各工序的开始时间和结束时间就如下表所示：

表格2.1. FLOWSHOP例子的时间表

机器1	A1:0-3	A2:3-7	A3:7-15	A4:15-25	
机器2		B1:3-9	B2:9-11	B3:15-24	B4:25-40

表2.1阐明了各工序的运作情况，及其起始时间和终止时间。这里我们强调，B_3之所以要延时到B_2完成之后(即11时刻以后)，是因为A_3在15时刻才完成。一个排定的总代价是第一个工序开始到最后一个工序完成所花费的时间，如上例的总代价就是40。流水线问题就是要我们找到一个总代价最小的排定。我们不难注意到总代价是以下三个时间的和：

1. 工序A的执行时间,记作c_1;
2. 最后一个工序B的执行时间，记作c_2;
3. 执行最后一个工序B前的延时，记作c_3。

在前面的例子中，$c_1 = 25$，$c_2 = 15$且$c_3 = 0$，其总代价为40。
为了使用DP解决此题，我们必须对每个决策分别定义一个代价。我们可以通过虚拟一个阶段来做到这一点。如果我们接下来要处理的工件是d，定义执行其工序A的代价为p_d; 其工序B的执行时间q_d因为加上虚拟阶段变量k而可能发生延时。明确地说，我们定义状态为(k, S)，这里虚拟阶段k是指当前已考虑的工序中工序A和工序B最后完成时间的间隔，且S是尚未选择的工件集合。如果$k < p_d$(即如果没有延时)，那么这个间隔时间为0。决策d是S的一个元素。在状态(k,S)中选择工件d的代价是p_d，决策d带来的延时是$\max(k - p_d, 0)$。因此，下一状态就是$(\max(k - p_d, 0) + q_d, S - \{d\})$。综上所述，该问题可用下述状态转移方程解决:

$$f(k, S) = \min_{d \in S} \{p_d + \max(k - p_d, 0) + q_d, S - \{d\}\}$$

我们的目标是计算$f(0, S^*)$，这里S^*是最初给出的N个工件的集合。其边界条件是当$S = \emptyset$ 时，$f(k, S) = k$。对于上面的例子，$f(0, S^*) = 38$，其对应的最优决策序列是$d_1 = 0, d_2 = 2, d_3 = 3, d_4 = 1$。