

DSA_14

邵宁录 2018202195

2020 年 12 月 18 日

目录

1	25.2-4	2
2	25.3-1	2
3	26-3	3
3.1	a	3
3.2	b	3
3.3	c	3

1 25.2-4

不妨以 d_{ij} 为例。

1. 若 $\min(d_{ij}, d_{ik} + d_{kj}) = d_{ij}$, 则显然, $d_{ij} = d_{ij}$ 。这与之前的算法一样, 没有任何变化。
2. 若 $\min(d_{ij}, d_{ik} + d_{kj}) = d_{ik} + d_{kj}$, 那么 d_{ik} 与 d_{kj} 可能经历了 $k-1$ 次或 k 次松弛操作。但不管经历了几次, d_{ij} 是中间结点取自 $1, 2, 3, \dots, k$ 的一条最短路径, 这是一个循环不变量。则当 $k = n$ 时, 全部的 d_{ij} 均得到了包含所有结点的最短路径。

2 25.3-1

v	h
1	-5
2	-3
3	0
4	-1
5	-6
6	-8

表 1: h 的值

u	v	\hat{w}	u	v	\hat{w}
1	2	<i>NIL</i>	4	1	0
1	3	<i>NIL</i>	4	2	<i>NIL</i>
1	4	<i>NIL</i>	4	3	<i>NIL</i>
1	5	0	4	5	8
1	6	<i>NIL</i>	4	6	<i>NIL</i>
2	1	3	5	1	<i>NIL</i>
2	3	<i>NIL</i>	5	2	4
2	4	0	5	3	<i>NIL</i>
2	5	<i>NIL</i>	5	4	<i>NIL</i>
2	6	<i>NIL</i>	5	6	<i>NIL</i>
3	1	<i>NIL</i>	6	1	<i>NIL</i>
3	2	5	6	2	0
3	4	<i>NIL</i>	6	3	2
3	5	<i>NIL</i>	6	4	<i>NIL</i>
3	6	0	6	5	<i>NIL</i>

表 2: \hat{w} 的值

最短路径结果如下：

$$\begin{bmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{bmatrix}$$

3 26-3

3.1 a

由题意得，对于每个结点 $A_k \in R_i$ ，则图 G 包含边 (A_k, J_i) ，其容量为边 $c(A_k, J_i) = \infty$ 。
由于分割 (S, T) 的容量是有限的，根据分割的容量定义：

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$$

若对于每个结点 $A_K \in R_i$ ，有 $A_k \notin T$ ，则有 $A_k \in S$ 。
则显然会有 S 到 T 的流 $c(A_k, J_i)$ 存在，而 $c(A_k, J_i) = \infty$ ，这与切割 (S, T) 容量有限矛盾。
因此假设不成立，即对于每个结点 $A_K \in R_i$ ，有 $A_k \in T$ 。

3.2 b

最大净收入相当于所有接受的工作的营收减去所有聘请的专家的费用。设所有接受的工作的集合为 J_{yes} ，所有拒绝的工作的集合为 J_{no} 。所有聘请的专家的子领域集合为 A_{yes} ，所有没聘请的为 A_{no} 。
有净收入如下式：

$$Profit = \sum_{J_{yes}} p_i - \sum_{A_{yes}} c_i$$

而 $\sum_{J_{yes}} p_i = \sum_J p_i - \sum_{J_{no}} p_i$ ，则有 $Profit = \sum_J p_i - (\sum_{J_{no}} p_i + \sum_{A_{yes}} c_i)$ 。
下面说明 $\sum_{J_{no}} p_i + \sum_{A_{yes}} c_i$ 为有限切割 (S, T) 的容量。
不妨构造这样的一个切割。其中 A_{yes} 都在 T 中， A_{no} 都在 S 中， J_{yes} 都在 T 中， J_{no} 都在 S 中。
由于 (S, T) 是有限的，所以没有穿过 S 和 T 的边 (A_k, J_i) 。那么有：

$$c(S, T) = \sum_{A_i \in A_{yes}} c(s, A_i) + \sum_{J_i \in J_{no}} c(J_i, t)$$

而根据题意，上式可以写成：

$$c(S, T) = \sum_{J_{no}} p_i + \sum_{A_{yes}} c_i$$

所以， $\sum_{J_{no}} p_i + \sum_{A_{yes}} c_i$ 为有限切割 (S, T) 的容量。
而 $Profit$ 最大，即 $\sum_{J_{no}} p_i + \sum_{A_{yes}} c_i$ 最小。因此根据最大流最小切割定理，用 S 到 T 的最大流算法就能求解这个问题。用全部工作的营收减去最小切割容量，就得到了最大利润。

3.3 c

算法步骤：

1. 根据题意构建一个流网络 G 。
2. 运用最大流最小切割算法获得一个最小切割 (S, T) 。

3. 根据最小切割的结果来判断哪些工作该接受（接受在 T 中的工作）。

总共的节点数为 $|V| = m + n + 2$ ，边数为 $|E| = r + m + n$ 。

构造流网络所花的时间为线性 $O(|E|)$ 。获得最小切割，如果用 *Edmonds-Karp* 算法的话，时间复杂度为 $O(|V||E|^2)$ 。

所以总的时间复杂度为算法的话，时间复杂度为 $O(|V||E|^2)$ 。