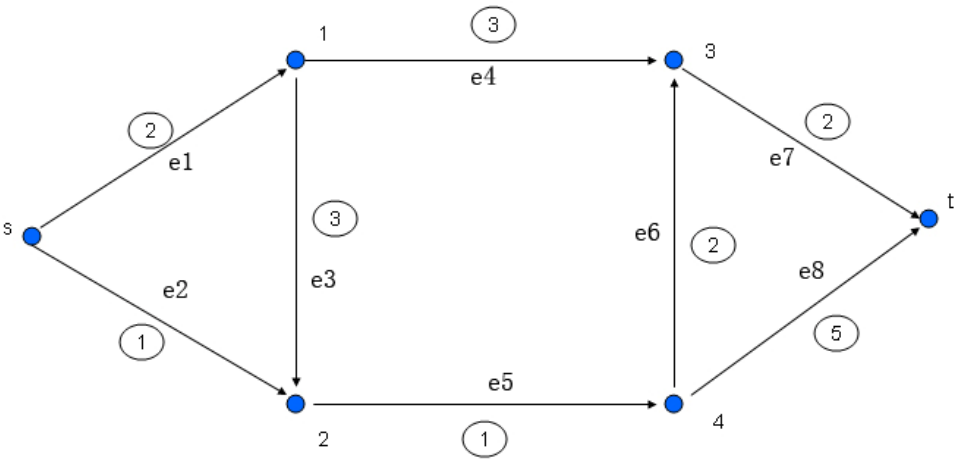


第五章 组合优化问题的原始-对偶算法-2

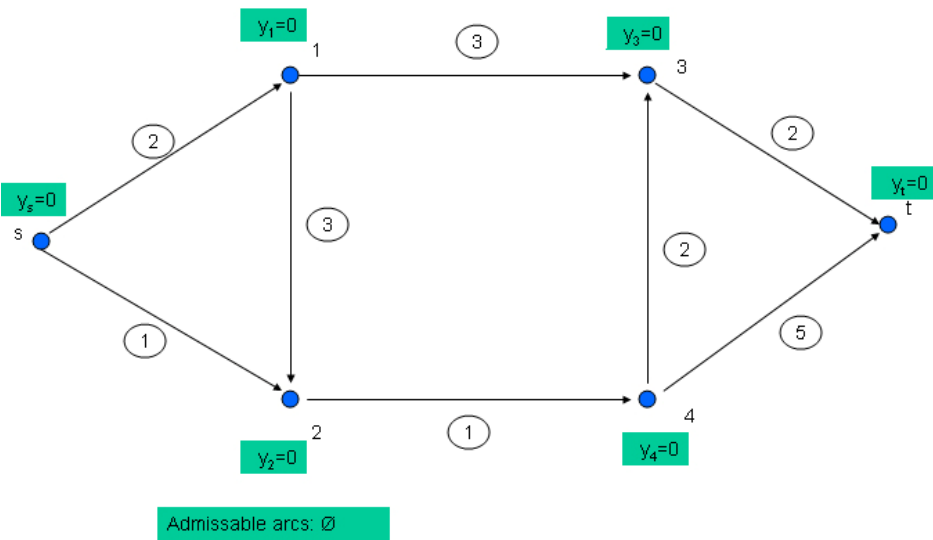
例如：



由于费用非负，所以 $y = 0$ 为(D)：

$$\begin{aligned} \max \quad & y_s \\ \text{s.t.} \quad & y_i - y_j \leq c_{ij} \quad (i, j) \in E \\ & y_i \text{ 无限制, } i \in V \\ & y_t = 0 \end{aligned} \tag{D}$$

的可行解，即



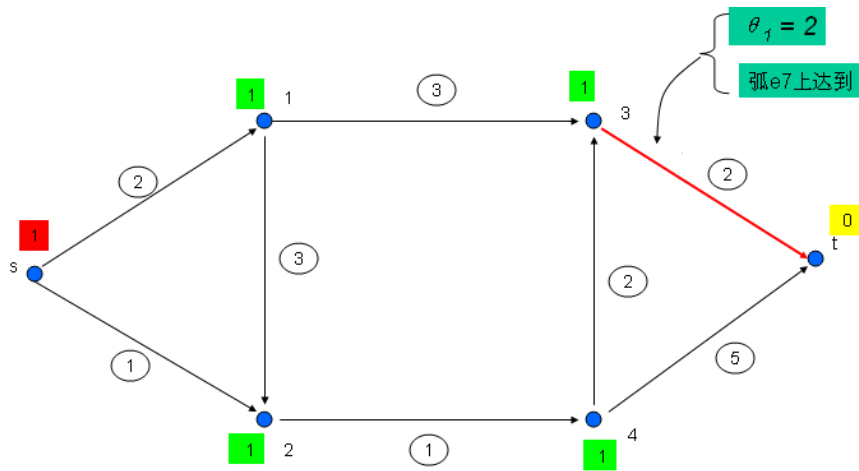
$J = \Phi$

第一次迭代：对应的(DRP)为

$$\begin{aligned}
\max \omega &= y_s \\
y_i - y_j &\leq 0, \quad (i, j) \in J = \Phi \\
y_i &\leq 1, \quad \text{对一切 } i \\
y_i &\text{无限制} \\
y_t &= 0
\end{aligned}$$

显然不存在仅利用 J 中的弧自 s 到 t 的路, 令 $\overline{y_s} = 1$, 在不破坏约束 $y_i - y_j \leq 0$ 的条件下,

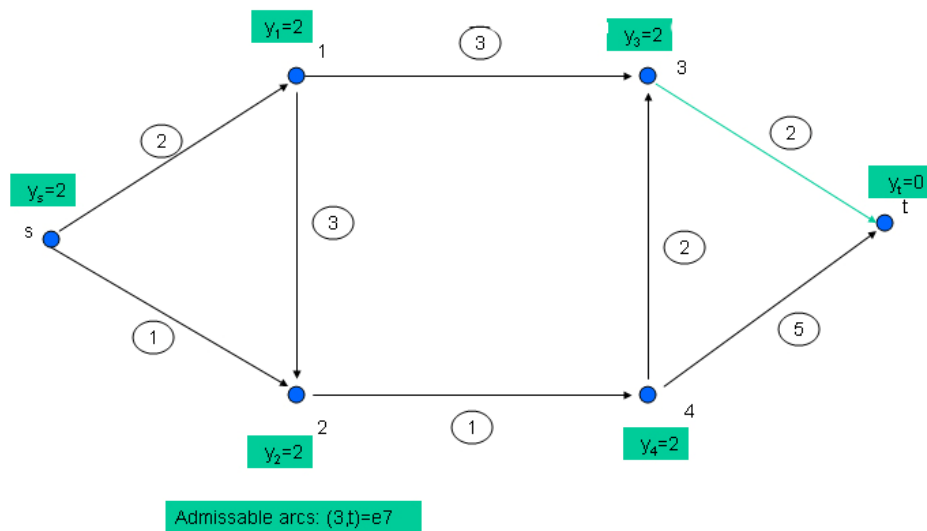
- (1) 将仅利用 J 中的弧自 s 有路可到达的所有节点对应的对偶变量分配值 1;
- (2) 将仅利用 J 中的弧自 i 到达节点 t 的所有节点对应的对偶变量分配值 0 ($\overline{y_t} = 0$);
- (3) 其余情形节点对应的对偶变量取值为 1: $\overline{y_i} = 1 (i = 1, 2, 3, 4)$



计算 $\theta_1 = \min_{\substack{(i,j) \notin J \\ y_i - y_j > 0}} \{c_{ij} - (y_i - y_j)\} = 2$, 弧 e_7 上达到。

注: 因为 $\overline{y_t} = 0$ 且 $\overline{y_i} = 1 (i = 1, 2, 3, 4)$, 此时 $\overline{y_i} - \overline{y_j} > 0$ 只有 $j = t$ 时才成立。

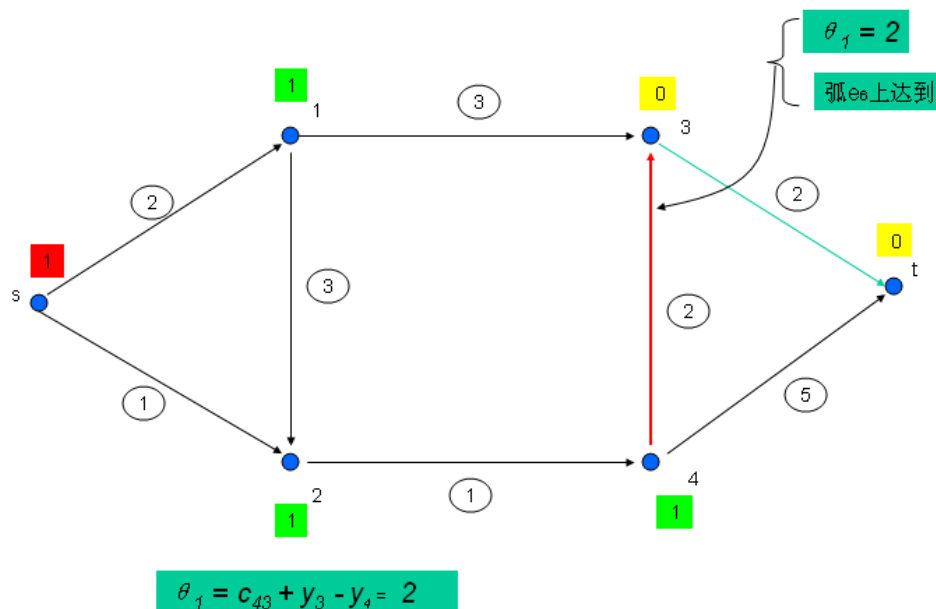
得到(D)的新的可行解: $y = (y_s, y_1, y_2, y_3, y_4)^T = (2, 2, 2, 2, 2)^T$



$$J = \{e_7\}$$

第二次迭代：令 $\overline{y_s} = 1$ ，显然不存在仅利用 J 中的弧自 s 到 t 的路，令 $\overline{y_s} = 1$ ，在不破坏约束 $y_i - y_j \leq 0$ 的条件下，

- (1) 将仅利用 J 中的弧自 s 有路可到达的所有节点对应的对偶变量分配值 1；
- (2) 将仅利用 J 中的弧自 i 到达节点 t 的所有节点对应的对偶变量分配值 0 ($\overline{y_t} = 0, \overline{y_3} = 0$)；
- (3) 其余情形节点对应的对偶变量取值为 1: $\overline{y_1} = \overline{y_2} = \overline{y_4} = 1$



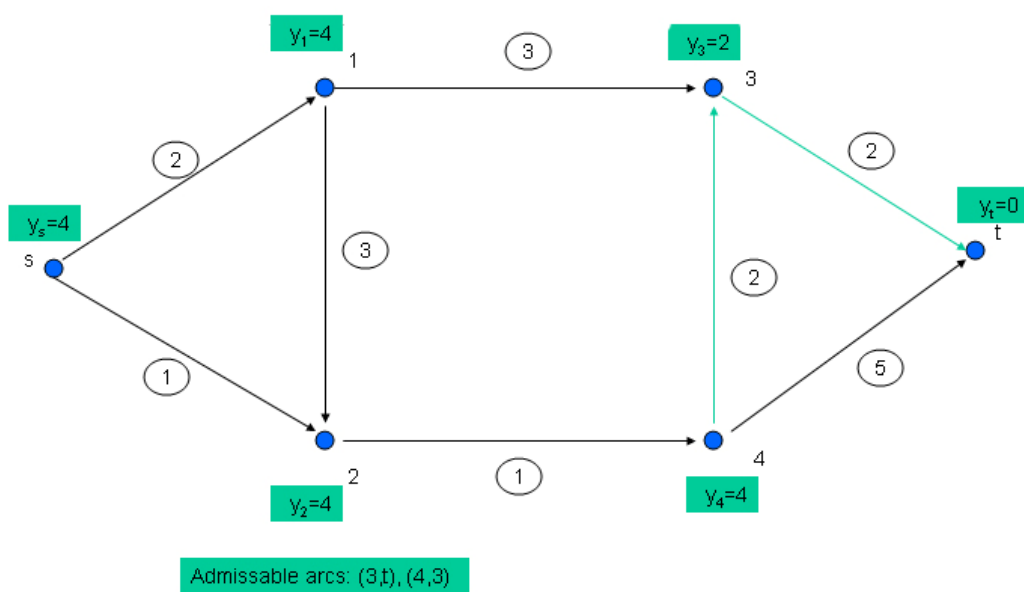
计算：

$$\theta_1 = \min_{\substack{(i,j) \notin J \\ y_i - y_j > 0}} \{c_{ij} - (y_i - y_j)\}$$

$$= \min\{c_{43} - (y_4 - y_3), c_{4t} - (y_4 - y_t)\} = 2$$

所以, (D)的新的可行解为:

$$y = (y_s, y_1, y_2, y_3, y_4)^T = (2, 2, 2, 2, 2)^T + \theta_1(1, 1, 0, 1)^T = (4, 4, 4, 2, 4)^T$$

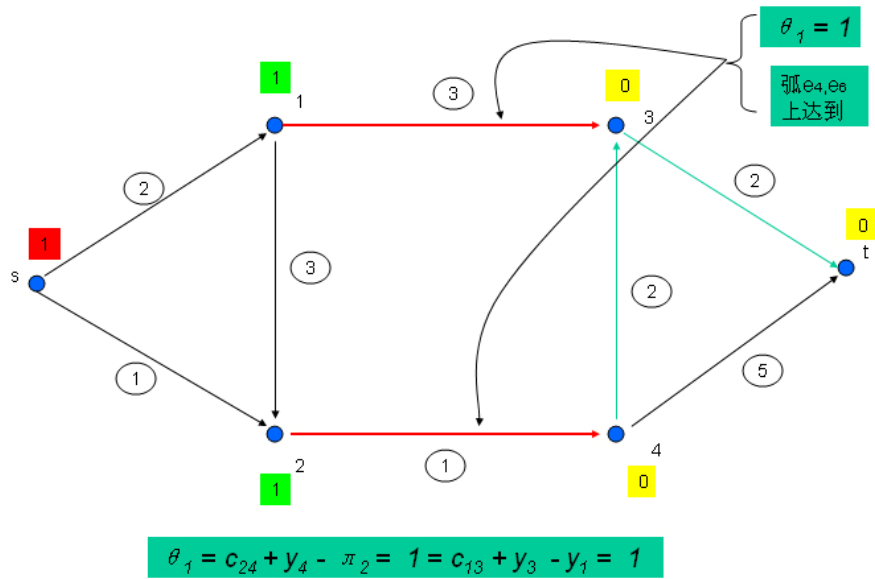


$$J = \{e_7, e_6\}$$

第三次迭代: 令 $\overline{y_s} = 1$, 显然不存在仅利用 J 中的弧自 s 到 t 的路, 令 $\overline{y_s} = 1$, 在不破坏约束 $y_i - y_j \leq 0$ 的条件下,

- (1) 将仅利用 J 中的弧自 s 有路可到达的所有节点对应的对偶变量分配值 1;
- (2) 将仅利用 J 中的弧自 i 到达节点 t 的所有节点对应的对偶变量分配值 0
($\overline{y_t} = 0, \overline{y_3} = \overline{y_4} = 0$);

- (3) 其余情形节点对应的对偶变量取值为 1: $\overline{y_1} = \overline{y_2} = 1$



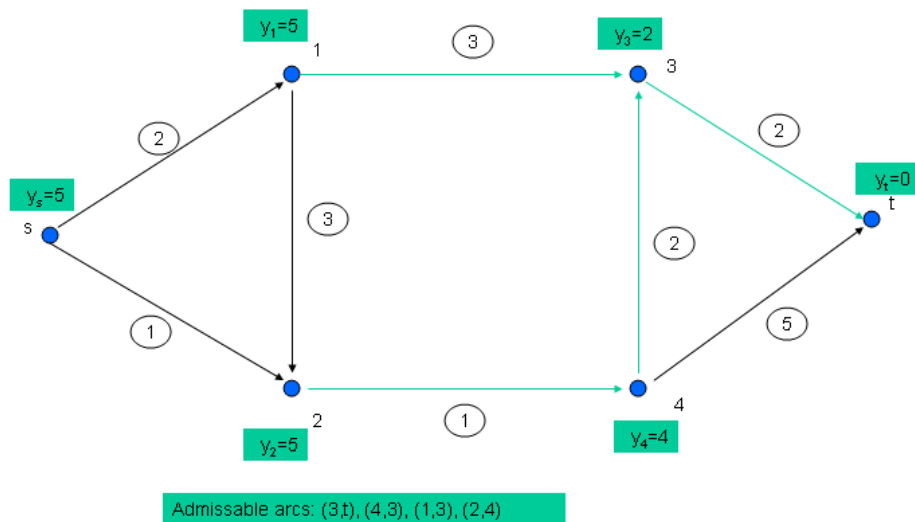
计算:

$$\theta_1 = \min_{\substack{(i,j) \notin J \\ y_i - y_j > 0}} \{c_{ij} - (y_i - y_j)\}$$

$$= \min\{c_{13} - (y_1 - y_3), c_{24} - (y_2 - y_4)\} = 1$$

所以, (D)的新的可行解为:

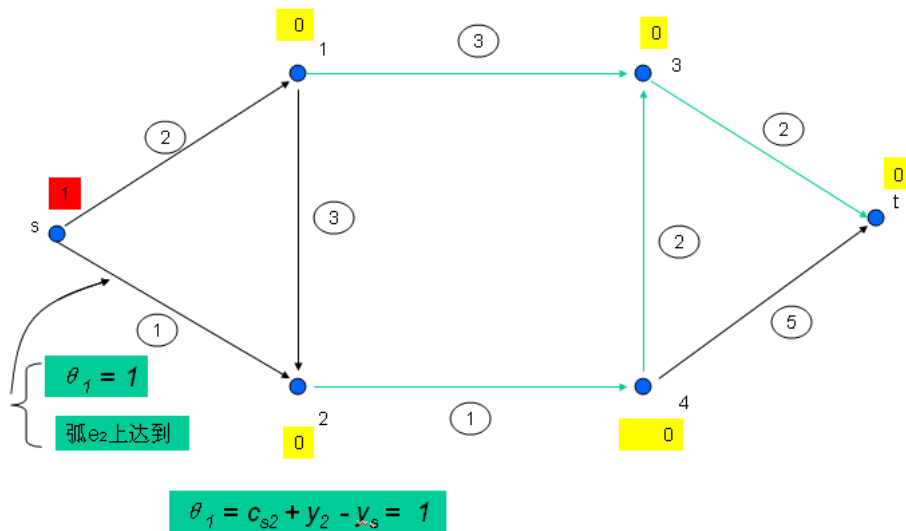
$$y = (y_s, y_1, y_2, y_3, y_4)^T = (4, 4, 4, 2, 4)^T + \theta_1 (1, 1, 1, 0, 0)^T = (5, 5, 5, 2, 4)^T$$



$$J = \{e_7, e_6, e_5, e_4\}$$

第四次迭代: 令 $\overline{y_s} = 1$, 显然不存在仅利用 J 中的弧自 s 到 t 的路, 令 $\overline{y_s} = 1$, 在不破坏约束 $y_i - y_j \leq 0$ 的条件下,

- (1) 将仅利用 J 中的弧自 s 有路可到达的所有节点对应的对偶变量分配值 1;
- (2) 将仅利用 J 中的弧自 t 到达的所有节点对应的对偶变量分配值 0
 $(\bar{y}_t = 0, \bar{y}_1 = \bar{y}_2 = \bar{y}_3 = \bar{y}_4 = 0)$;
- (3) 其余情形节点对应的对偶变量取值为 1



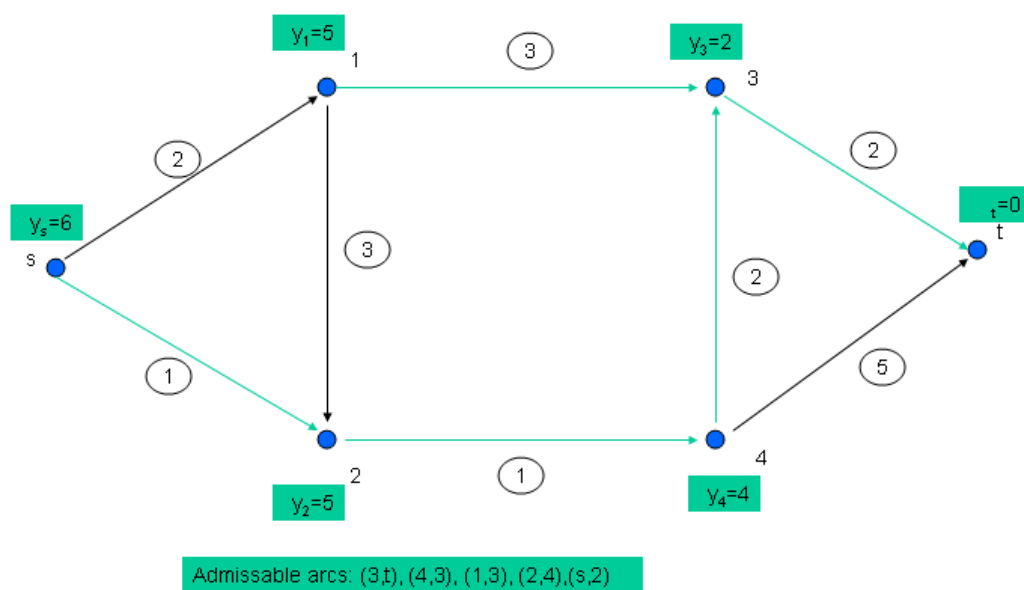
计算:

$$\theta_1 = \min_{\substack{(i,j) \notin J \\ y_i - y_j > 0}} \{c_{ij} - (y_i - y_j)\}$$

$$= \min\{c_{s1} - (y_s - y_1), c_{s2} - (y_s - y_2)\} = 1$$

所以, (D)的新的可行解为:

$$y = (y_s, y_1, y_2, y_3, y_4)^T = (5, 5, 5, 2, 4)^T + \theta_1 (1, 0, 0, 0, 0)^T = (6, 5, 5, 2, 4)^T$$



$$J = \{e_7, e_6, e_5, e_4, e_2\}$$

显然存在仅利用 J 中的弧自 s 到 t 的路, 得到最优路: e_2, e_5, e_6, e_7 , 其最优费用为 $y_s = 6$ 。

四、最短路问题原始-对偶算法的几点说明

(1) 在算法的任何一个阶段, 如果我们定义集合

$$W = \{i: \text{在允许弧集中有自 } i \text{ 到 } t \text{ 的路}\} = \{i: \bar{y}_i = 0\}$$

那么, 在接下来的迭代过程中, 任给 $i \in W$, 将始终保持 $\bar{y}_i = 0$, 从而, 始终保持第 i 个节点对应的对偶变量 y_i 的取值保持不变;

(2) 若某一弧 $(i, j) \in E$, 一旦有 $y_i - y_j = c_{ij}$, 即弧 $(i, j) \in E$ 变成允许弧后, 则在接下来的迭代过程中:

(a) 若存在仅利用 J 中的弧自 j 到达节点 t 的路, 则必存在仅利用 J 中的弧自 i 到达节点 t 的路, 从而 $\bar{y}_i = \bar{y}_j = 0$;

(b) 若存在仅利用 J 中的弧自 s 有路可到达的节点 i 的路, 则必存在仅利用 J 中的弧自 s 有路可到达的节点 j 的路, 从而 $\bar{y}_i = \bar{y}_j = 1$;

(c) 其它情形: $\bar{y}_i = \bar{y}_j = 1$

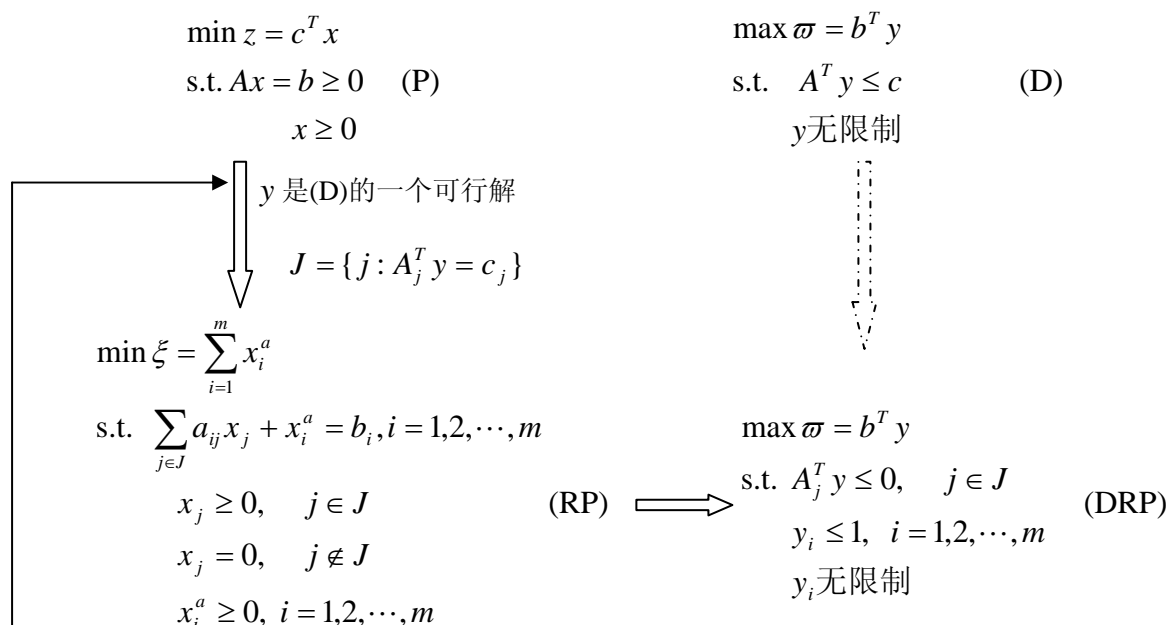
因此, 在接下来的迭代过程中, y_i 和 y_j 始终保持相同的改变量, 也即始终保持 $y_i - y_j = c_{ij}$, 所以, 任何一弧变为允许弧后 (即进入 J), 则在算法的接下来的过程中它始终保持是允许的。

(3) 算法表明, 对于 $i \in W$, y_i 表示自 i 到达节点 t 的最短路长度, 并且算法进行中, 每一步加到 W 里的点都是 $\bar{W} = E/W$ 中最靠近 t 的点。所以: 最短路问题的原始-对偶算法, 实际上就是 Dijkstra 算法。

(4) 由于算法的每一个阶段, W 中至少增加一个节点, 所以, 最短路问题的原始-对偶算法的迭代步数是有界的, 并且不超过 $|V|$ 。

五、关于原始-对偶算法的进一步说明

原始-对偶算法的过程：



用单纯形类方法求解(RP)， 若：

- (RP)的最优值 $\xi_{\text{opt}} = 0$ ，则得到(LP)和(D)的最优值，算法终止；
- (RP)的最优值 $\xi_{\text{opt}} > 0$ ，设 \bar{y} 是(DRP)的最优解，若 $A_j^T \bar{y} \leq 0, j \notin J$ ，则(D)无上界，从而(LP)不可行，算法终止；

否则，取 $\theta \leq \theta_1 = \min_{\substack{A_j^T \bar{y} > 0, \\ j \notin J}} \left\{ \frac{c_j - A_j^T \bar{y}}{A_j^T \bar{y}} \right\}, y := y + \theta \bar{y}$

1. 从原始问题的观点看

$(P, c) \rightarrow$ 迭代地求解 RP，其中 RP 不明显地依赖于费用向量 c ，而是通过允许变量集

$J = \{j : A_j^T y = c_j\}$ 间接地应用到费用向量 c 。

由价格向量 y 的迭代循环可见，我们已消除了一般费用向量的困难，所以称之为“**组合化费用**”。

2. 从对偶问题的观点看

在从 D 到 DRP 时，DRP 不明显地依赖于 D 的右端项，所以称之为“**组合化右端项**”。

因此：

$$\begin{array}{ll} P \rightarrow RP & \text{组合化费用} \\ D \rightarrow DRP & \text{组合化右端项} \end{array}$$

在最短路问题里，P 的右端项本质上平凡的，而所有数据是通过费用向量 c 表示出来的。所以，RP 和它的对偶 DRP 都不明显地依赖于数值问题的数据，而只依赖于允许变量集 J ，从而导致一个纯组合的子问题。在最短路问题里就是一个简单的可达性问题。

从一个问题开始，用原始一对偶算法迭代地求解组合性的子问题，这一技术是组合优化的核心。组合优化中的一类重要问题——流和匹配问题的几乎所有有效算法，都是以此为基础。

六、最短路问题的 Dijkstra 标号算法

Dijkstra 标号算法：最短路问题的原始一对偶算法的一种有效实施。迭代步数至多为 n ，算法的计算复杂度为 $O(n^2)$ 。

思考题：将 Dijkstra 标号算法与最短路问题的原始一对偶算法进行比较，说明 Dijkstra 标号算法实际上就是最短路问题的原始一对偶算法的一种具体实施。

七、最短路问题的 Floyd-Washall 算法

1. 特点：对一个图，同时求出所有节点对之间的最短路：

- (1) 算法程序简单有效；
- (2) 与 Dijkstra 标号算法比较，允许网络中的弧有负权，而且可以探测网络中的负费用圈；
- (3) 不像是原始一对偶算法。

2. 方法：直接在一个 $n \times n$ 的距离矩阵 $(d_{ik})_{n \times n}$ 上进行操作。

3. 算法的核心：三角运算

4. 算法的理论基础：如果对相继值 $j = 1, 2, \dots, n$ 对距离矩阵 $(d_{ik})_{n \times n}$ 相继进行三角运算，那么每一个 d_{ik} 都变为从 i 到 k 的最短路的长度。

5. 算法的计算复杂度： $O(n^3)$ 。

要求：理解掌握最短路问题的 Floyd-Washall 算法。