第 5 章 匹配

5.1 二分图的最大匹配

图的匹配问题有其丰富的实际背景，它涉及了二分图与一般图的最大匹配，二分图与一般图的最佳匹配等，除了一般图的最佳匹配之外，本章都将一一进行讨论。

例 5.1.1 m 项工作准备分配给 n 个人去做，如图 5.1 所示，其中边 $(x_i,\ y_j)$ 表示 x_i 可以从事 y_j，如果每个人最多从事其中一项，且每项工作只能由一人承担。问怎样才能给尽可能多的人安排上任务。

图 5.1 是二分图，按照要求，如果 x_i 从事了 y_j，就不允许再从事 y_k，同时 y_j 也不再允许其他人承担。因此，它相当于用一种颜色，例如红色对 G 的边进行着色，保证每个顶点最多只与一条红色边关联。这种红色边的集合记为 M，它就称为匹配。原问题就是计算 G 中包含边数最多的一个匹配 M。

例 5.1.2 第二次世界大战期间，盟军许多飞行人员到英国参加对法西斯德国的空袭行动，当时每架飞机需要领航员和飞行员各 1 人。其中有些人只能领航，一些人只会驾驶，也有人两者均会。加之二人语言要求相通，因此如果以顶点表示人，边表示两者语言相通并且一人可领航另一人可驾驶，就会得到如图 5.2 所示的一个简单图 G。那么最多的编队方案就是计算 G 中的一个最大匹配。

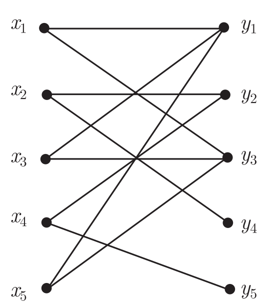

图 5.1

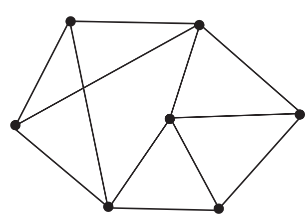

图 5.2

定义 5.1.1 令 M 是图 G 的边子集，若 M 中任意两条边都没有共同的顶点，则称 M 是 G 的一个**匹配**，其中与 M 的边关联的顶点称为**饱和点**，否则称为**非饱和点**。

定义 5.1.2 设 M 是 $G=(V,\ E)$ 中的一个匹配，如果对 G 的任意匹配 M'，都有 $|M| \geqslant |M'|$，就说 M 是 G 的一个**最大匹配**。

启发与思考

如果给出二分图以及一个匹配，如何确定它是不是最大匹配？如果不是，怎么在这个匹配的基础上找到一个更大的匹配？为了解决这些问题，需要相关定义。

定义 5.1.3 给定了 G 的一个匹配 M，G 中属于 M 与不属于 M 的边交替出现的道路称为**交互道路**。构成回路的交互道路称为交互回路。

有时这种交互道路可能构成交互回路。

定义 5.1.4 设 P 是 G 中关于匹配 M 的一条交互道路，如果 P 的两个端点是关于 M 的非饱和点，那么它就称为**可增广道路**。

可增广道路 P 一定包含奇数条边，且其中不属于匹配 M 的边比 M 中的边多一条。同时 $P \oplus M$ 仍然是 G 的一个匹配 M'，它使 P 的两个端点变成饱和点，这时 $|M'| = |M| + 1$，即 M' 是比 M 更大的匹配。

定理 5.1.1 M 是 G 的最大匹配当且仅当 G 中不存在关于 M 的可增广道路。

证明：必要性。若存在 M 的可增广道路 P，则 $M \oplus P = M'$ 是 G 的一个新匹配，且 $|M'| > |M|$，与 M 是最大匹配矛盾。充分性。如果匹配 M 不是 G 的最大匹配，则存在一个最大匹配 M'，做 $G' = M' \oplus M$，我们逐一分析 G' 中 3 种可能的连通支。

(1) 孤立顶点，当 $(v_i,\ v_j) \in M' \cap M$ 时会出现孤立点 v_i 和 v_j。

(2) 初级回路，该回路中属于 M' 和属于 M 的边数相同。

(3) 初级道路，如果不存在增广道路，那么 $|M'| = |M|$，与假设矛盾。如果存在 M 关于 M' 的增广路，又与 M' 是最大匹配矛盾。由于 $|M'| > |M|$，故必定存在 M' 关于 M 的可增广交互道路，即 G 中存在关于 M 的可增广道路。

定理 5.1.1 是二分图和一般图最大匹配算法的依据。不过由于二分图的所有回路都是偶回路的特点，因此它的最大匹配算法较为简单。

计算二分图最大匹配的一个好算法是匈牙利算法。描述如下。

匈牙利算法

（输入为二分图 $G = (X,\ Y,\ E)$；顶点标记 0 表示尚未搜索，顶点标记 1 表示是饱和点，顶点标记 2 表示是无法扩大匹配的顶点）。

1. 任给一初始匹配 M，给饱和点标记“1”。
2. 判 X 中的各顶点是否都已有非零标记。
   2.1 是。M 是最大匹配，结束。
   2.2 否。找一 0 标记点 $x_0 \in X$，
   令 $U \leftarrow \{x_0\}$，$V \leftarrow \phi$。
3. 判集合 U 的邻接点集 $\Gamma(U) = V$?
   3.1 是，x_0 无法扩大匹配，给 x_0 标记“2”，转 2。
   3.2 否，在 $\Gamma(U) - V$ 中找一点 y_i，判 y_i 是否标“1”。

3.2.1　是，则有边 $(y_i,\ z) \in M$。令

$U \leftarrow U \cup \{z\}$，$V \leftarrow V \cup \{y_i\}$，转 3。

3.2.2　否，存在从 x_0 至 y_i 的可增广路 P，

令 $M \leftarrow M \oplus P$，给 x_0 和 y_i 标记 “1”，转 2。

> ☞ **启发与思考**
>
> 我们已经学习过如何在有向图中寻找道路（使用深探法或广探法）。能否采用化归的思想，为 G 中的边规定方向，得到一个有向图，使得可增广道路都是其中的道路，从而解决最大匹配问题。实际上，按照边是否在匹配 M 中来规定方向，使得交互道路都是有向图中的道路即可。以此思路，可以尝试以广探法的思想理解上述的匈牙利算法中的步骤 3。

例 5.1.3　图 5.3 中，设初始匹配 $M=\{(x_1,\ y_1),\ (x_3,\ y_4),\ (x_4,\ y_5)\}$。用匈牙利算法求其最大匹配的过程如下。

(1) $U=\{x_2\}$，$V=\phi$。

$\Gamma(U)=\{y_4,\ y_6\}$，$y_6 \in \Gamma(U)-V$，　且无标记。

所以　增广路 $P=(x_2,\ y_6)$。

$M=\{(x_1,\ y_1),\ (x_3,\ y_4),\ (x_4,\ y_5),\ (x_2,\ y_6)\}$。

(2) $U=\{x_5\}$，$V=\phi$。

$\Gamma(U)=\{y_5,\ y_6\}$，$y_5 \in \Gamma(U)-V$。

$U=\{x_5,\ x_4\}$，$V=\{y_5\}$。

$\Gamma(U)=\{y_5,\ y_6\}$，$y_6 \in \Gamma(U)-V$。

$U=\{x_5,\ x_4,\ x_2\}$，$V=\{y_5,\ y_6\}$。

$\Gamma(U)=\{y_5,\ y_6,\ y_4\}$，$y_4 \in \Gamma(U)-V$。

$U=\{x_5,\ x_4,\ x_2,\ x_3\}$，$V=\{y_5,\ y_6,\ y_4\}$。

$\Gamma(U)=\{y_5,\ y_6,\ y_4,\ y_2\}$，$y_2 \in \Gamma(U)-V$ 且无标记。

所以　有增广路 $P=(x_5,\ y_6,\ x_2,\ y_4,\ x_3,\ y_2)$。

$M=\{(x_1,\ y_1),\ (x_4,\ y_5),\ (x_5,\ y_6),\ (x_2,\ y_4),\ (x_2,\ y_3)\}$。

(3) $U=\{x_6\}$，$V=\phi$。

$\Gamma(U)=\{y_6\}$，$y_6 \in \Gamma(U)-V$。

$U=\{x_6,\ x_5\}$，$V=\{y_6\}$。

$\Gamma(U)=\{y_6,\ y_5\}$，$y_5 \in \Gamma(U)-V$。

$U=\{x_6,\ x_5,\ x_4\}$，$V=\{y_6,\ y_5\}$。

$\Gamma(U)=\{y_6,\ y_5\}$，$\Gamma(U)=V$。　给 x_6 标记2。结束。

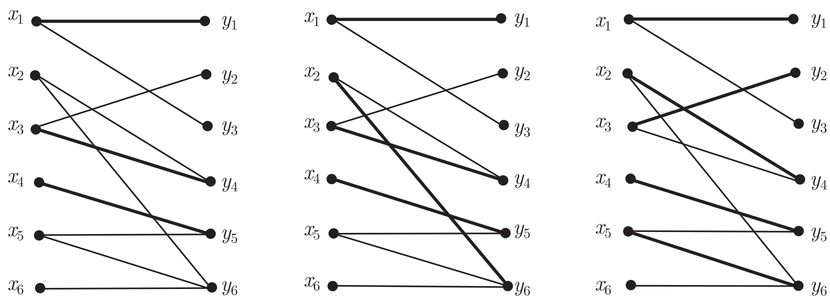

图 5.3

因此，其最大匹配是 $M=\{(x_1, y_1), (x_4, y_5), (x_5, y_6), (x_2, y_4), (x_3, y_2)\}$。

定理 5.1.2 最大匹配匈牙利算法的计算复杂性是 $O(mn)$,其中二分图 $G(X, Y, E)$ 中，$n=|X|$，$m=|E|$。

证明：初始匹配可以是空匹配，算法最多找 n 条增广路，每找一条增广路时，最多判断 m 条边，因此其计算复杂性是 $O(mn)$。

5.2 完全匹配

二分图 $G=(X, Y, E)$ 的最大匹配 M 包含的边数不会超过 $|X|$，若 $|M|=|X|$，则称 M 是完全匹配。特别地，如果 $|M|=|X|=|Y|$，则称 M 是完美匹配。直观地看，如果每个顶点 x 关联的边愈多，则最大匹配的边数可能愈大。例如图 5.4(a) 有完全匹配，而图 5.4(b) 没有完全匹配。那么满足什么条件 G 中就会有完全匹配呢？霍尔 (Hall) 定理给出了判别的标准。

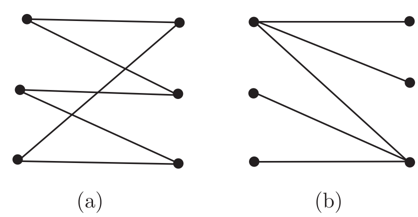

图 5.4

定理 5.2.1 在二分图 $G=(X, Y, E)$ 中，X 到 Y 存在完全匹配的充要条件是对于 X 的任意子集 A，恒有

$$|\Gamma(A)| \geqslant |A|$$

证明：必要性。若存在子集 $A \subseteq X$，使 $|A| > |\Gamma(A)|$，则 A 中的顶点无法全部匹配。因此，X 到 Y 不可能有完全匹配。充分性。假定 G 的一个最大匹配 M 不是完全匹配，一定存在顶点 $x_0 \in X$ 是关于 M 的非饱和点。如果 $\Gamma(x_0)=\varPhi$，则令 $A=\{x_0\}$，于是 $|\Gamma(A)|<|A|$，不满足条件。如果 $\Gamma(x_0) \neq \varPhi$，如图 5.5 所示，对某一个 $y_j \in \Gamma(x_0)$，若 y_j 关于 M 为非饱

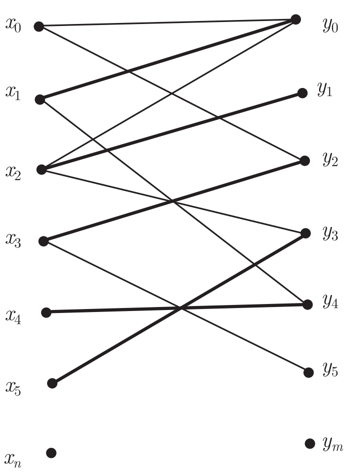

图 5.5

和点，则存在增广路 (x_0, y_j)，与 M 是最大匹配矛盾。因此，$y_j \in \Gamma(x_0)$ 都是关于 M 的饱和点。这样可以寻找以 x_0 为端点的相对于 M 的一切交互道路，记交互道中顶点 y_j 的集合为 Y_1，结点 x_i 的集合为 X_1，根据匹配的性质 Y_1 的顶点与 $X_1 - x_0$ 的顶点之间存在一一对应，于是 $|X_1| > |Y_1|$，即 $|X_1| > |\Gamma(X_1)|$。

推论 5.2.1　若二分图 $G = (X, Y, E)$ 的每个顶点 $x_i \in X$，都有 $d(x_i) \geqslant k$，每个顶点 $y_i \in Y$，都有 $d(y_j) \leqslant k$，那么 X 到 Y 存在完全匹配。

证明：对任意子集 $A \subseteq X$，设它的顶点总共与 m 条边关联，于是有 $m \geqslant k|A|$，这 m 条边又与 Y 中的 $|\Gamma(A)|$ 个顶点相关联，又有 $m \leqslant k|\Gamma(A)|$，因此 $|\Gamma(A)| \geqslant |A|$，由定理 5.2.1 即得。

例 5.2.1　在一个舞会上男女各占一半，假定每位男士都认识 k 位女士，每位女士也认识 k 位男士。那么一定可以安排得当，使每位都有认识的人作为舞伴。

证明：用顶点 x_i 表示每位男士，y_j 表示每位女士，互相认识者用边连接。于是得到二分图 $G = (X, Y, E)$，图中每个顶点 x_i 有 $d(x_i) = k$，y_j 有 $d(y_j) = k$。满足 $d(x_i) \geqslant k$，$d(y_j) \leqslant k$，由推论 5.2.1，X 到 Y 有完美匹配 M。M 就是一种安排方案。

二分图的完全匹配一定是最大匹配，而最大匹配不一定就是完全匹配，那么它们之间有什么内在联系呢?

定理 5.2.2　在二分图 $G = (X, Y, E)$ 中，X 到 Y 最大匹配的边数是 $|X| - \delta(G)$，其中 $\delta(G) = \max\limits_{A \subseteq x} \delta(A)$，$\delta(A) = |A| - |\Gamma(A)|$，$\delta(A) \geqslant 0$。

证明略。

例 5.2.2　10 个人有 10 件不同的乐器，其中 3 人只会拉小提琴，其余 7 人每件乐器都会，若每人只用一件乐器，则由定理 5.2.2，最多只有 8 人能同时登台演出。

二分图是一个图，当然可以用邻接矩阵表示。由于其全部的边都跨越在 X 和 Y 之间，因此，可以将邻接矩阵进行简化，成为 $|X| \times |Y|$ 的一个矩阵。例如图 5.6 的邻接矩阵是

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

这样 G 中的最大匹配数 r 就是 $\boldsymbol{A}$ 中不在同行同列非零元的最多个数。如果矩阵 $\boldsymbol{A}$ 是 $p \times q$ 的，显然有 $r \leqslant \min(p, q)$。

另外，也可以适当地选取 $\boldsymbol{A}$ 的某些行和列，使这些行和列能盖住 $\boldsymbol{A}$ 中的全部非零元，这称为 $\boldsymbol{A}$ 的覆盖，当然如果盖住 $\boldsymbol{A}$ 的全部 p 行，或全部 q 列，就一定会盖住所有非零元，

但这不一定是最少选取的行与列，例如图 5.6 的矩阵 $\boldsymbol{A}$，如果盖住其第 4、6 行，第 2、4 列，就可以覆盖其全部非零元。因此，在矩阵 $\boldsymbol{A}$ 的全部覆盖中，一定存在最小覆盖，其覆盖数为 s，显然 $s \leqslant \min(p，q)$。

定理 5.2.3 设 r 是二分图 G 的最大匹配数，s 是其邻接矩阵的最小覆盖数，则有 $r = s$。

证明：因为每个不在同行同列的非零元需要一行或一列才能盖住，所以 r 个不在同行同列的非零元需要 r 行、列才能盖住，而 s 个不同的行、列盖住了矩阵 $\boldsymbol{A}$ 的全部非零元，自然也盖住了 r 个不在同行同列的非零元。因此 $s \geqslant r$。再证 $r \geqslant s$。不失一般性，设最小覆盖盖住了 $\boldsymbol{A}$ 的 c 行、d 列，即 $s = c + d$。设这 c 行对应的顶点子集是 X_c，其余为 $X - X_c$；d 列对应的顶点集是 Y_d，其余为 $Y - Y_d$。把矩阵 $\boldsymbol{A}$ 的行、列进行调整，如图 5.7 所示，显然 A_{11} 每行都被覆盖，A_{22} 每列都被覆盖，A_{12} 中每个元素既被行也被列所覆盖，而 $A_{21} = 0$。

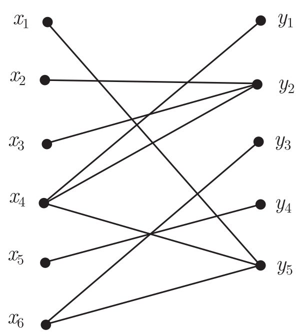

图 5.6

$$
\begin{array}{c}
X_c \\
X-X_c
\end{array}
\left[\begin{array}{c:c}
A_{11} & A_{12} \\
\hdashline
A_{21} & A_{22}
\end{array}\right]
$$
$$
\qquad Y-Y_d \qquad\qquad Y_d
$$

图 5.7

现证明 X_c 到 $Y - Y_d$ 存在完全匹配。在 A_{11} 中任取 $|V'|$ 行，这些行中的非零元至少分布在它的 $|V'|$ 个不同列上。否则，不覆盖这 $|V'|$ 行，而覆盖这些更少的列，$\boldsymbol{A}$ 中的所有非零元仍然全部覆盖，这时所用的覆盖数比原先要少，与原来是最小覆盖矛盾。这就是说，对 X_c 的任意子集 V'，它在 $Y - Y_d$ 中的邻接点集是 $\Gamma(V')$，总有 $|\Gamma(V')| \geqslant |V'|$。根据定理 5.2.1，$X_c$ 到 $Y - Y_d$ 存在完全匹配 M_1，$|M_1| = c$。同理，Y_d 到 $X - X_c$ 也存在完全匹配 M_2，$|M_2| = d$，$M_1 \cup M_2$ 仍然是 G 的一个匹配，$|M_1 \cup M_2| = c + d = s$。因此，$G$ 的最大匹配数 $r \geqslant s$。

定理 5.2.3 不但揭示了匹配与覆盖之间的关系，而且也是最佳匹配算法的基本依据之一。

5.3 最佳匹配及其算法

5.1 节和 5.2 节讨论的都是边权为 1 的匹配问题，如果边权是非负实数，而且存在多个完全匹配，那么其中权和最大或最小的完全匹配就叫作最佳匹配。

例 5.3.1　5 项工作由 5 个人完成，如下所示。

$$\boldsymbol{C}=\begin{bmatrix}3&4&6&4&9\\6&4&5&3&8\\7&5&3&4&2\\6&3&2&2&5\\8&4&5&4&7\end{bmatrix}$$

其中 c_{ij} 表示 i 从事工作 j 的利润，如果每个人只做一项工作，那么最大的利润就应该是 $\max\Sigma c_{ij}$，c_{ij} 不在相同的行与列。

假如 c_{ij} 表示 i 从事工作 j 的成本，那么最小的成本应该是 $\min\Sigma c_{ij}$，c_{ij} 不在相同的行与列。

> ☞启发与思考
>
> 实际问题中的情况一般更为复杂，如工作数与人数可能不等，允许一部分工作不被完成，有的人可能做不了某些工作，或是做某些工作的利润为负（即导致亏损）。该例题实际上已将问题充分简化，使得工作数与人数相等，利润均存在且非负。这样一来，便能找到一个较为简单的算法解决该问题，而将较为复杂的情况化归为该算法能解决的情况也是简单的，如添加一些虚拟的点和边来解决两边点数不同的问题，给所有边权加上一个固定值解决有负权边的问题等。

显然这种最佳匹配就是二分图的最大权或最小权匹配。在讨论最佳匹配时，二分图 $G=(X，Y，E)$ 满足条件 $|X|=|Y|$。

我们先介绍一个利用最小覆盖取代最大匹配的最大权匹配算法。

最大权匹配算法（已知利润矩阵 $\boldsymbol{C}$）。

1. 在矩阵 $\boldsymbol{C}$ 的每行选一最大值作为本行的界值 $l(x_i)$，每列的界值 $l(y_j)=0$。构造矩阵 $\boldsymbol{B}=(b_{ij})_{n\times n}$，其中 $b_{ij}=l(x_i)+l(y_j)-c_{ij}$。
2. 在 $\boldsymbol{B}$ 中对 0 元素进行最小覆盖，覆盖数为 r。

   2.1 若 $r=n$，转 4。

   2.2 在未覆盖的元素中选最小非零元，设值为 δ。

   若 x_i 行、y_j 列均已覆盖，则 $b_{ij}\leftarrow b_{ij}+\delta$。

   若 x_i 行、y_j 列均未覆盖，则 $b_{ij}\leftarrow b_{ij}-\delta$。
3. 修改界值

   若 x_i 行没覆盖，令 $l(x_i)\leftarrow l(x_i)-\delta$；

   若 y_j 列已覆盖，令 $l(y_j)\leftarrow l(y_j)+\delta$。

   删除覆盖标记，转 2。
4. $\sum(l(x_i)+l(y_j))$ 即是最大权，结束。

例 5.3.2　求例 5.3.1 表中的最大利润。

解：首先得到矩阵 $\boldsymbol{B}$，界值已在表的两旁标出，最小覆盖是 1、5 两列，$\delta=2$。

$$\begin{array}{cc}
 & \begin{array}{ccccc}\downarrow & & & & \downarrow\end{array} \\
\begin{array}{c}9\\8\\7\\6\\8\end{array} & \begin{bmatrix}6&5&3&5&0\\2&4&3&5&0\\0&2&4&3&5\\0&3&4&4&1\\0&4&3&4&1\end{bmatrix} \\
 & \begin{array}{ccccc}0&0&0&0&0\end{array}
\end{array}$$

$r<n$，$\boldsymbol{B}$ 中没覆盖的元素均减 δ；修改界值，结果如下。这时一个最小覆盖是第 1、5 列，第 3 行。$\delta=1$。

$$\begin{array}{cc}
 & \begin{array}{ccccc}\downarrow & & & & \downarrow\end{array} \\
\begin{array}{c}7\\6\\5\\4\\6\end{array} & \begin{bmatrix}6&3&1&3&0\\2&2&1&3&0\\0&0&2&1&5\\0&1&2&2&1\\0&2&1&2&1\end{bmatrix} \\
 & \begin{array}{ccccc}2&0&0&0&2\end{array}
\end{array}$$

$r<n$，$\boldsymbol{B}$ 中没覆盖元素减 1，双重覆盖元加 1。修改界值，这时一个最小覆盖是第 1、2、3、5 列。$\delta=1$。

$$\begin{array}{cc}
 & \begin{array}{ccccc}\downarrow & \downarrow & \downarrow & & \downarrow\end{array} \\
\begin{array}{c}6\\5\\5\\3\\5\end{array} & \begin{bmatrix}6&2&0&2&0\\2&1&0&2&0\\1&0&2&1&6\\0&0&1&1&1\\0&1&0&1&1\end{bmatrix} \\
 & \begin{array}{ccccc}3&0&0&0&3\end{array}
\end{array}$$

$r<n$，$\boldsymbol{B}$ 中没覆盖元素减 1，修改界值，这时一个最小覆盖是第 3、4、5 行，第 3、5 列，最小覆盖数 $r=n$。一个最大权匹配方案是 $\{c_{13},\ c_{25},\ c_{34},\ c_{42},\ c_{51}\}$，$\Sigma(l(x_i)+l(y_i))=29$。结束。

$$\begin{array}{ccc}
 & \begin{array}{ccccc} & & \downarrow & & \downarrow\end{array} & \\
\begin{array}{c}5\\4\\4\\2\\4\end{array} & \begin{bmatrix}6&2&0&1&0\\2&1&0&1&0\\1&0&2&0&6\\0&0&1&0&1\\0&1&0&0&1\end{bmatrix} & \begin{array}{c}\\ \\ \leftarrow\\ \leftarrow\\ \leftarrow\end{array} \\
 & \begin{array}{ccccc}4&1&1&0&4\end{array} &
\end{array}$$

定理 5.3.1　算法的结果是矩阵 $\boldsymbol{C}$ 的最大权匹配。

证明：所选取的矩阵 $\boldsymbol{B}$ 满足

$$b_{ij}=l\left(x_i\right)+l\left(y_j\right)-c_{ij}\geqslant 0 \tag{5-1}$$

设 W 是 $\boldsymbol{C}$ 的最大权匹配权和，一定有

$$\sum\left(l\left(x_i\right)+l\left(y_j\right)\right)\geqslant \max\sum c_{ij}=W \tag{5-2}$$

如果等式成立，那么一定存在 n 个不在同行同列的 c_{ij}，满足 $c_{ij}=l(x_i)+l(y_j)$，或者说 $\boldsymbol{B}$ 中有 n 个不在同行同列的 0 元素，处于这 n 个位置的 c_{ij} 构成了最大权匹配。如果最多存在 $k(<n)$ 个这样的 0 元素，式 (5-2) 就不可能相等，设此时的最小覆盖盖住了 c 行 d 列，其对应的顶点集为 X_c 和 Y_d，由定理 5.2.3，$k=c+d<n$。令 $\boldsymbol{B}$ 中没被覆盖的最小元是 $\delta(>0)$，按照算法在修改界值时，对没覆盖的各行，令 $l^*(x_i)=l(x_i)-\delta$；对覆盖的各列，令 $l^*\left(y_j\right)=l\left(Y_j\right)+\delta$，其余界值不变。为了保证式 (5-1) 不变，即修改界值后仍需满足

$$b_{ij}^*=l^*\left(x_i\right)+l^*\left(y_j\right)-c_{ij}\geqslant 0$$

就应该对 b_{ij} 的不同位置分别进行如下处理。

(1) 在覆盖的行和列的交叉点位置，

$$b_{ij}^*=l^*\left(x_i\right)+l^*\left(y_j\right)-c_{ij}=l\left(x_i\right)+l\left(y_j\right)+\delta-c_{ij}=b_{ij}+\delta$$

(2) 没被覆盖，$b_{ij}^*=l\left(x_i\right)-\delta+l\left(y_j\right)-c_{ij}=b_{ij}-\delta$。

(3) 只被行覆盖，$b_{ij}^*=l\left(x_i\right)+l\left(y_j\right)-c_{ij}=b_{ij}$。

(4) 只被列覆盖，$b_{ij}^*=l\left(x_i\right)-\delta+l\left(y_j\right)+\delta-c_{ij}=b_{ij}$。

在上述每种情况下，$b_{ij}^*\geqslant 0$ 成立。综上，在对界值和元素 b_{ij} 修改之后，式 (5-1) 和式 (5-2) 继续保持成立。但新的界值之和

$$\sum\left(l^*\left(x_i\right)+l^*\left(y_j\right)\right)=\sum\left(l\left(x_i\right)+l\left(y_j\right)\right)-\delta(n-c)+\delta d$$

而

$$-\delta(n-c)+\delta d=\delta(c+d-n)<0$$

即界值之和下降。由于 δ 选值最小，因此它是最小下降。界值以及 b_{ij} 调整后，$\boldsymbol{B}$ 中出现了新的 0 元素，将可能增加最小覆盖数。经过若干次迭代之后，界值之和将恰好等于最大权匹配值。

如果矩阵 $\boldsymbol{C}$ 表示成本矩阵，那么它的最小权匹配或最小成本也就容易计算了。一种方法是确定一个 n 阶矩阵 $\boldsymbol{Q}=(q_{ij})$，其中 q_{ij} 是一个大于或等于 $\max c_{ij}$ 的常数 a，令 $\boldsymbol{C}'=\boldsymbol{Q}-\boldsymbol{C}$，则 $c'_{ij}+c_{ij}=a$。这样矩阵 $\boldsymbol{C}$ 的最小成本对应了 $\boldsymbol{C}'$ 的最大利润。对 $\boldsymbol{C}'$ 调用最大权匹配算法就容易计算 $\boldsymbol{C}$ 的最小成本。另一种方法类似于最大权匹配算法的思路。首先选每行的最小元为界值，满足 $b_{ij}=c_{ij}-l\left(x_i\right)-l\left(y_j\right)\geqslant 0$。然后不断最小地增加界值，直至存在 n 个不在同行同列值为 0 的 b_{ij} 出现。

例 5.3.3 求 $\boldsymbol{C}$ 的最小成本。

$$\boldsymbol{C}=\begin{bmatrix}7&6&4&6&1\\4&6&5&7&2\\3&5&7&6&8\\4&7&8&8&5\\2&6&5&6&3\end{bmatrix}$$

解：选用 $a=10$，得到矩阵 $\boldsymbol{C}'=\left(c'_{ij}\right)$，$c'_{ij}=10-c_{ij}$。$\boldsymbol{C}'$ 恰是例 5.3.2 计算的矩阵，因此 $\boldsymbol{C}$ 的最小成本是

$$5\times 10-\max\sum c'_{ij}=21$$

相应的一个最小权匹配方案是 $\{c_{13}，c_{25}，c_{34}，c_{42}，c_{51}\}$。

如果直接对 $\boldsymbol{C}$ 进行计算，过程可以如下。

首先得到矩阵 $\boldsymbol{B}$，界值已标出，满足 $b_{ij}=c_{ij}-l\left(x_i\right)-l\left(y_j\right)$，最小覆盖是第 1、5 列，$\delta=2$。

$$\begin{array}{c c}
 & \begin{array}{ccccc}\downarrow&&&&\downarrow\end{array}\\
\begin{array}{c}1\\2\\3\\4\\2\end{array} & \begin{bmatrix}6&5&3&5&0\\2&4&3&5&0\\0&2&4&3&5\\0&3&4&4&1\\0&4&3&4&1\end{bmatrix}\\
 & \begin{array}{ccccc}0&0&0&0&0\end{array}
\end{array}$$

$r<n$。$\boldsymbol{B}$ 中没覆盖的元素均减 δ，行、列都覆盖的加 δ，修改界值。没覆盖的行界值加 δ，覆盖的列界值减 δ。这时一个最小覆盖是第 1、5 列，第 3 行，$\delta=1$。

$$\begin{array}{c c c}
 & \begin{array}{ccccc}\downarrow&&&&\downarrow\end{array} & \\
\begin{array}{c}3\\4\\5\\6\\4\end{array} & \begin{bmatrix}6&3&1&3&0\\2&2&1&3&0\\0&0&2&1&5\\0&1&2&2&1\\0&2&1&2&1\end{bmatrix} & \begin{array}{c}\\\\\leftarrow\\\\\\\end{array}\\
 & \begin{array}{ccccc}-2&0&0&0&-2\end{array} &
\end{array}$$

$r<n$。$\boldsymbol{B}$ 中没覆盖的元素均减 δ，修改界值如下，这时一个最小覆盖是第 1、2、3、5 列，$\delta=1$。