第五章 匹配与网络流 I

计算机系网络所：张小平

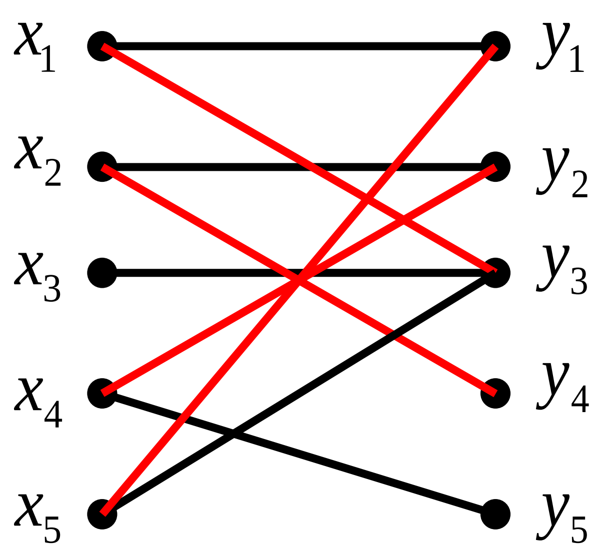
清華大學
Tsinghua University

匹配与网络流

- 匹配问题和网络流问题是图论中两类具有丰富实际应用背景的问题。
- 匹配问题涉及了二分图的最大匹配、最佳匹配问题。
- 网络流问题涉及网络流图及最大流、最小费用流等应用。

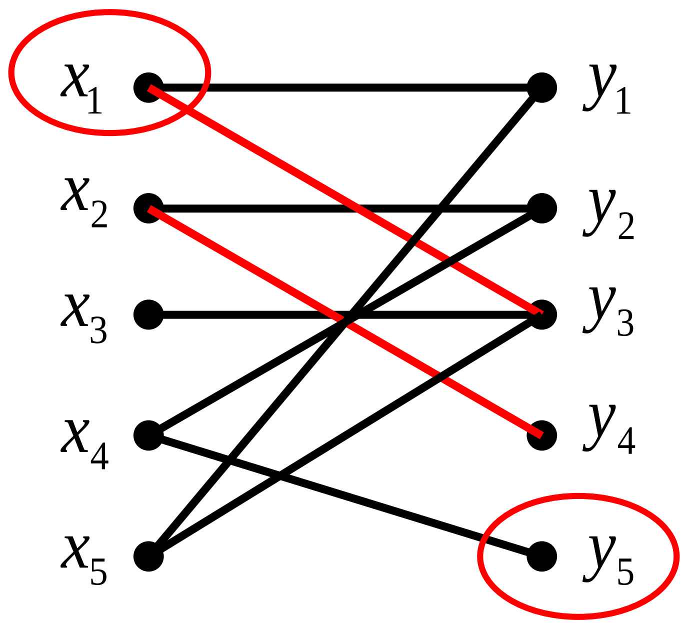
匹配问题

- 例：m项工作准备分配给n个人做，每个人可能会做其中某一项或几项工作，但每人只能承担一项工作。问如何分配才能让尽可能多的人安排上任务？
  - 如图所示，(x_1, y_3)表示 x_1 可做 y_3 工作
  - 问题相当于是对边进行着色，保证每个结点最多与一条着色边相关联
  - 这种着色边集合，就是一个**匹配**



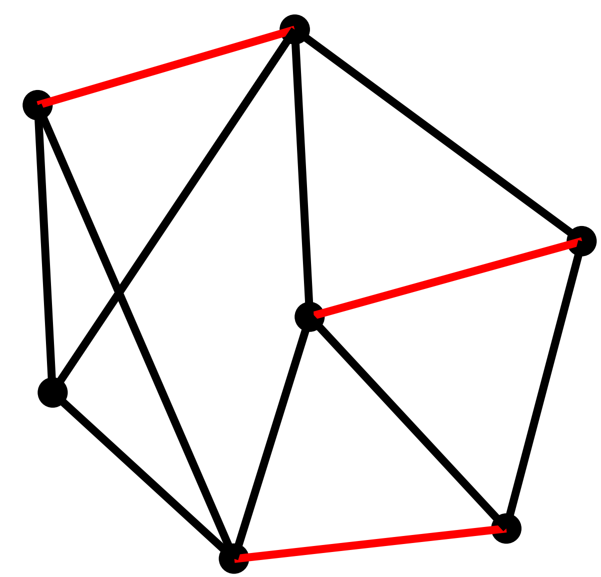
匹配问题

- 定义5.1.1 令M为图G的边子集，若M中任意两条边都没有共同的结点，则称M是G的一个**匹配**，其中与M的边关联的结点称为**饱和点**，否则称为**不饱和点**



匹配问题

- 例：二战期间，盟军许多飞行人员到英国参加对德空袭行动，当时每架飞机需要驾驶员和领航员各一人，二人需要语言相通。有人只会驾驶，有人只能领航，有人二者均可。问最多编队方案如何确定？
  - 图中边表示两人语言相通，且恰好一人会驾驶一人会领航
  - 则最多的编队方案可以转化求简单图G的一个最大匹配

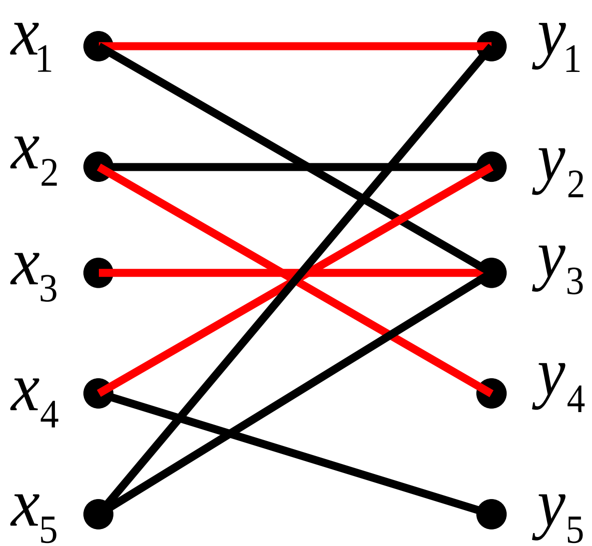


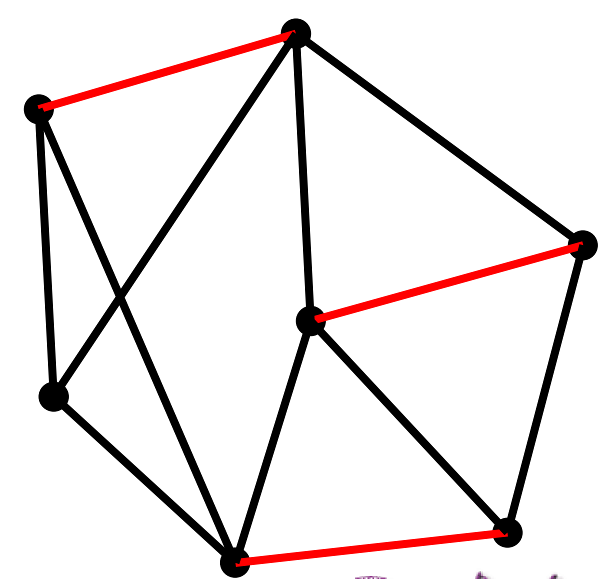
主要内容

- **5.1 二分图的最大匹配**
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

二分图的最大匹配

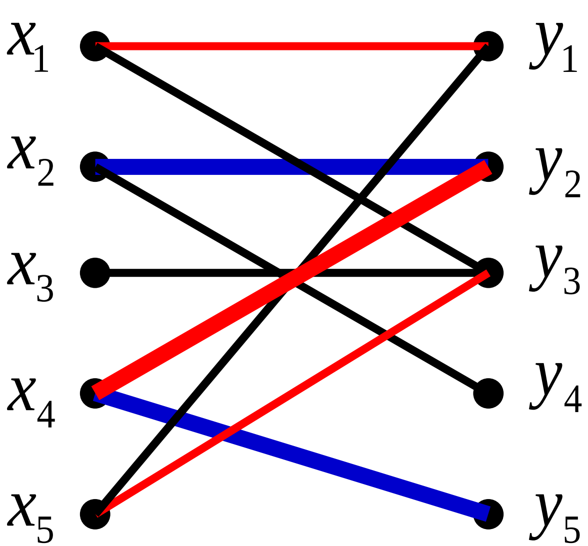
- 定义5.1.2 设M是G=(V,E)中的一个匹配，如果对G的任意匹配M’，都有 $|M|\geq|M'|$，就说M是G的一个**最大匹配**

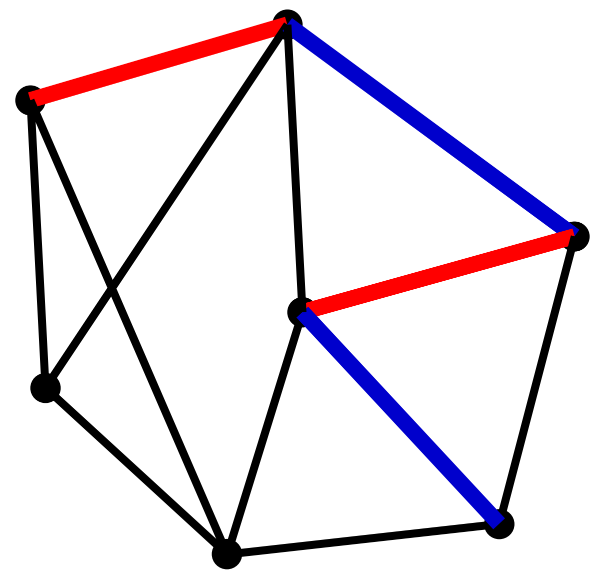




二分图的最大匹配

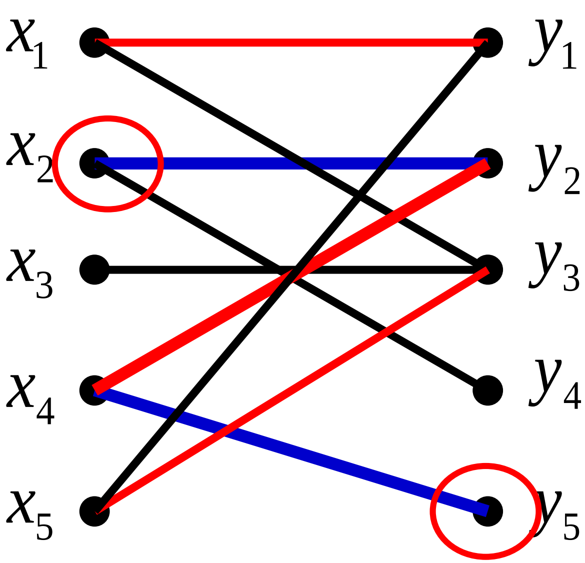
- 定义5.1.3 给定了G的一个匹配M，G中属于M与不属于M的边交替出现的道路称为**交互道路**

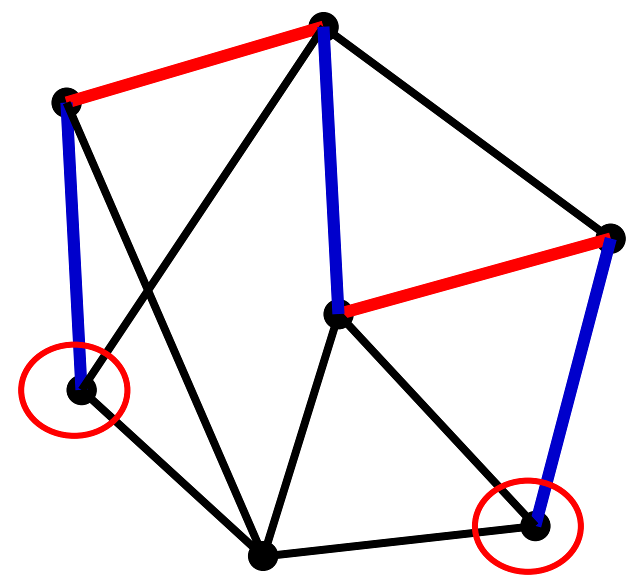




二分图的最大匹配

- 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P就称为**可增广道路**
  - $P \oplus M$ 仍然是G的一个匹配M'，且 $|M'| = |M| + 1$





二分图的最大匹配

- 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：

显然！

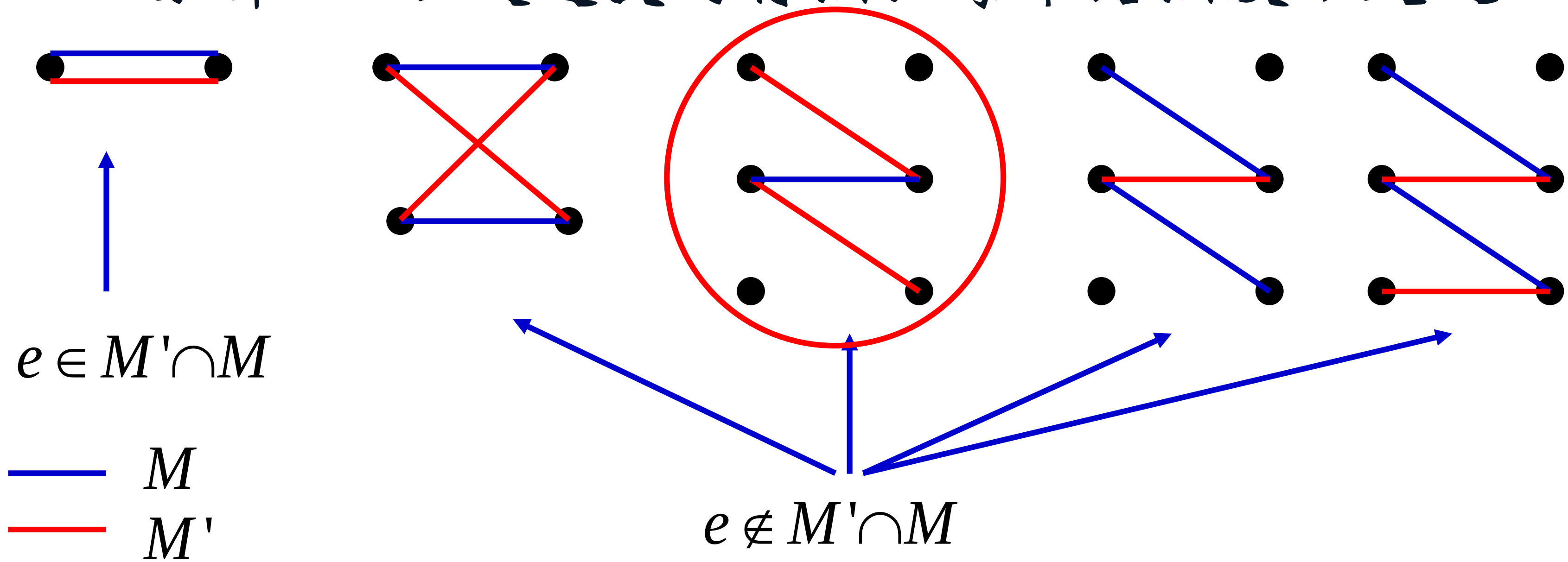
充分性

G中不存在关于M的可增广道路

⟶ M是G的最大匹配

二分图的最大匹配

- 假设匹配M不是G的最大匹配，则存在一个最大匹配M’，则必有$|M'|>|M|$，做 $G'=M'\oplus M$
- 分析G’各连通支的特点：每个结点度不超过2



- 由于$|M'|>|M|$，必定存在M’关于M的增广道路

二分图的最大匹配

- 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：

显然！

充分性（反证法）：

G中不存在关于M的可增广道路

→ M是G的最大匹配

证毕！

二分图的最大匹配

- 上述定理是最大匹配算法的基础，无论二分图还是一般图，都要依据此定理。

- 思考：
  - 如果给定一个二分图，如何找到它的最大匹配?
  - 假如能够找到可增广道路，即可找到更大匹配!

二分图的最大匹配

思考：

- 刚才的方法，是否一定可以找到二分图的最大匹配？
  - 每找一个结点，目标都是发现一条可增广道路
  - 寻找的方法实质是搜索过程，如果存在增广道路，则一定可以找到
  - 根据定理，不存在增广道路时，为最大匹配

二分图的最大匹配

- 匈牙利算法：
  - 经典的二分图最大匹配算法
- 算法描述：
  - 输入为二分图G=(X,Y,E)；初始匹配；
  - 结点标记为0：尚未搜索
  - 结点标记为1：饱和结点
  - 结点标记为2：无法扩大匹配的结点

Tsinghua University

二分图的最大匹配－匈牙利算法

1. 任给一初始匹配M，给饱和点“1”标记

2. 判断X中各结点是否都已经有非零标记

   2.1 是，M为最大匹配，结束

   搜索过的X结点集

   2.2 否，找一个“0”标记结点 $x_0 \in X$，$U \leftarrow \{x_0\}$，$V \leftarrow \Phi$

3. 判断集合U的邻点集 $\Gamma(U) = V$?

   U邻点集中搜索过的Y结点集

   3.1 是，给 x_0标记“2”，转2

   U邻点集中还未搜索过的Y结点集

   3.2 否，在$\Gamma(U)-V$中找一点 y_i，检查其是否标“1”

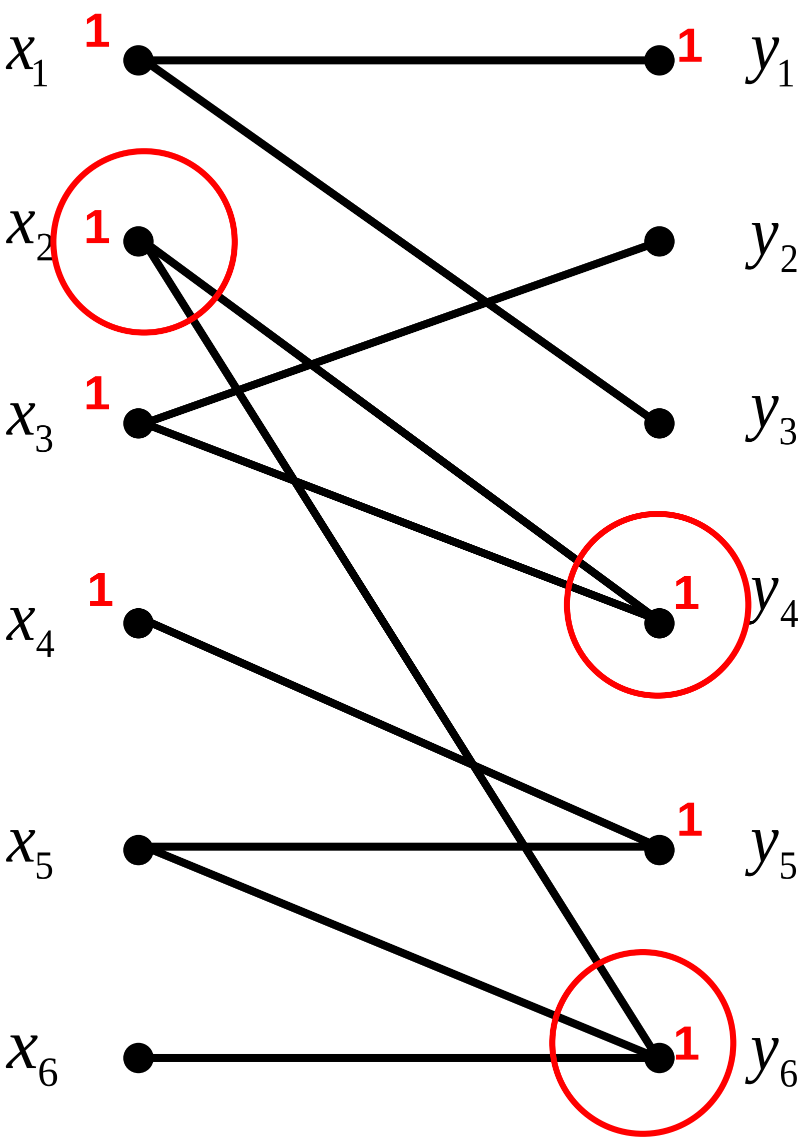
      3.2.1 是，则说明有边$(y_i, z) \in M$。令

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

      3.2.2 否，则说明找到了从x_0到y_i的增广路P，令

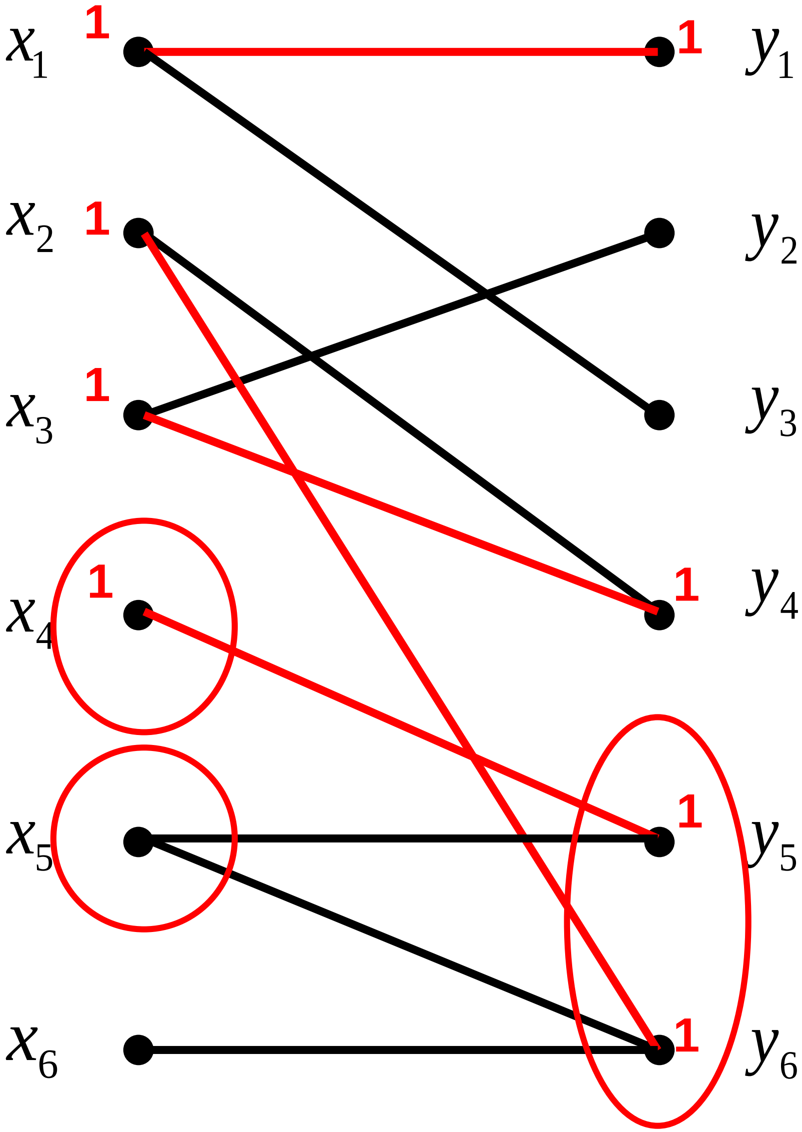
      $M \leftarrow M \oplus P$，给 x_0，y_i 标“1”，转2

例：设二分图G初始匹配为 $M=\{(x_1,y_1),(x_3,y_4),(x_4,y_5)\}$ 求其最大匹配



$U \leftarrow \{x_2\},\ \ V \leftarrow \Phi$

$\Gamma(U)=\{y_4, y_6\}$

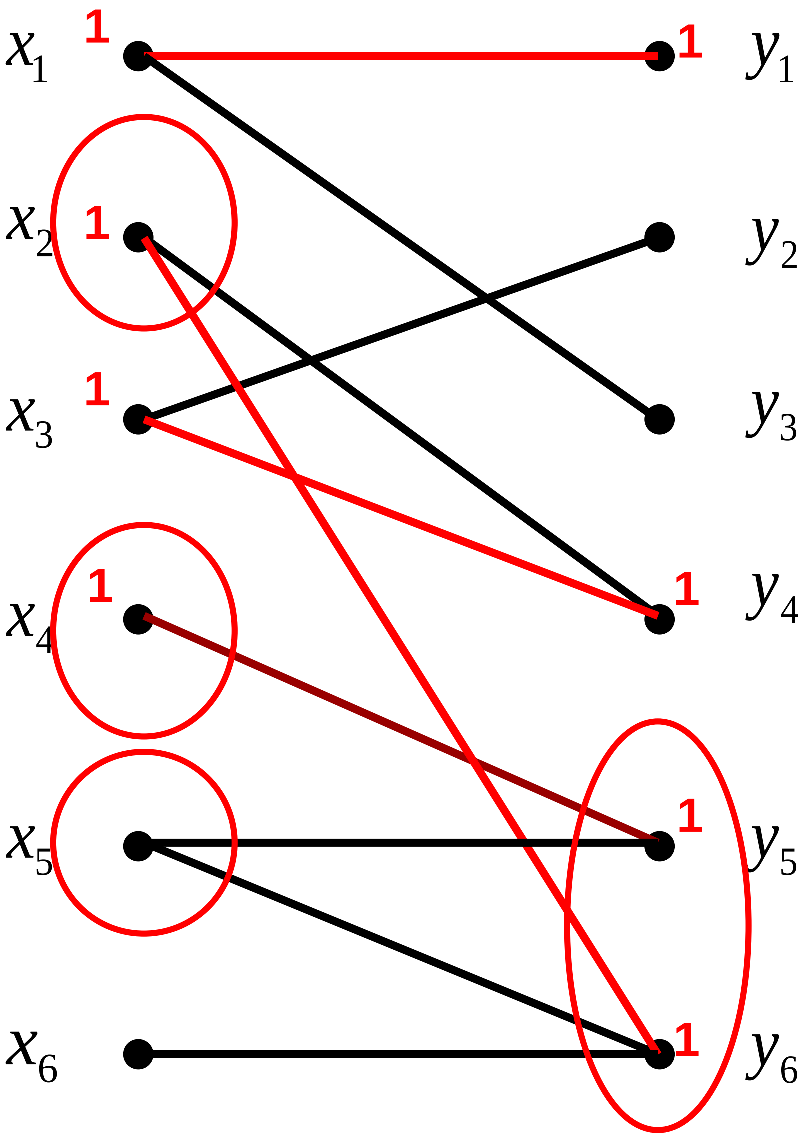


$$U \leftarrow \{x_5\}, \quad V \leftarrow \Phi$$

$$\Gamma(U) = \{y_5, y_6\}$$

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

$$U \leftarrow \{x_5, x_4\}, \quad V \leftarrow \{y_5\}$$

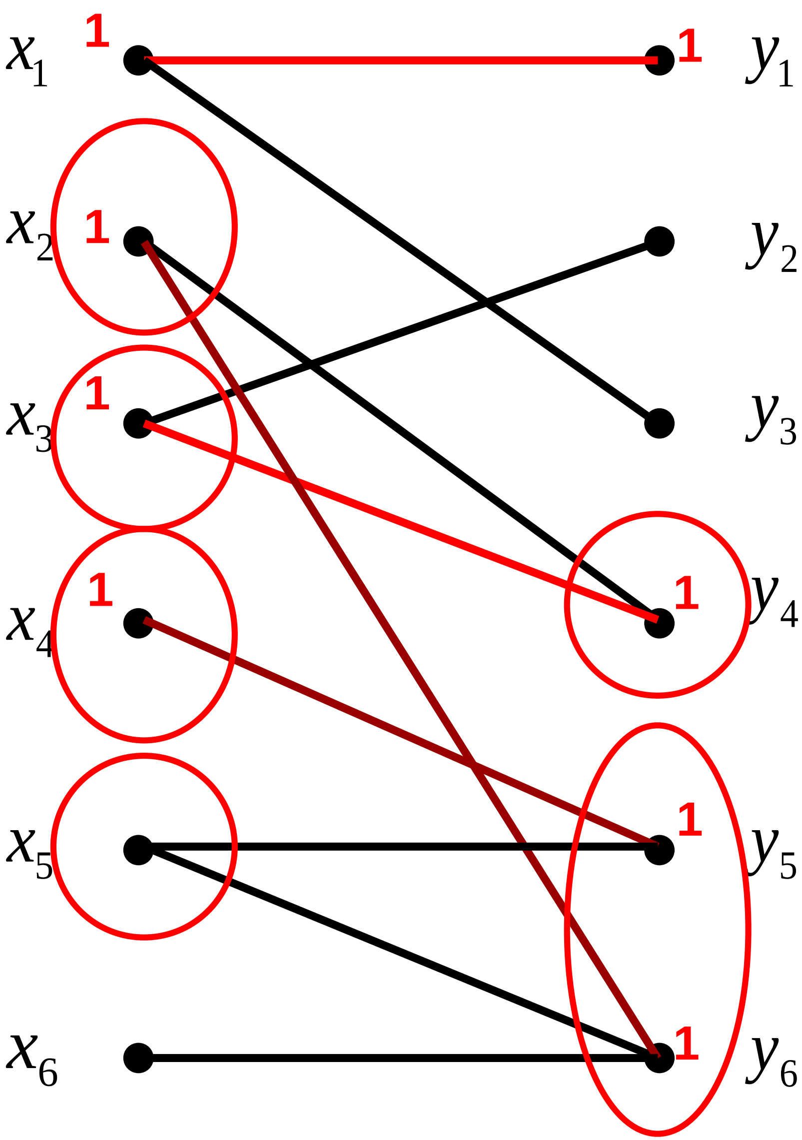


$$U \leftarrow \{x_5, x_4\}, \;\; V \leftarrow \{y_5\}$$

$$\Gamma(U) = \{y_5, y_6\}$$

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

$$U \leftarrow \{x_5, x_4, x_2\}, \;\; V \leftarrow \{y_5, y_6\}$$

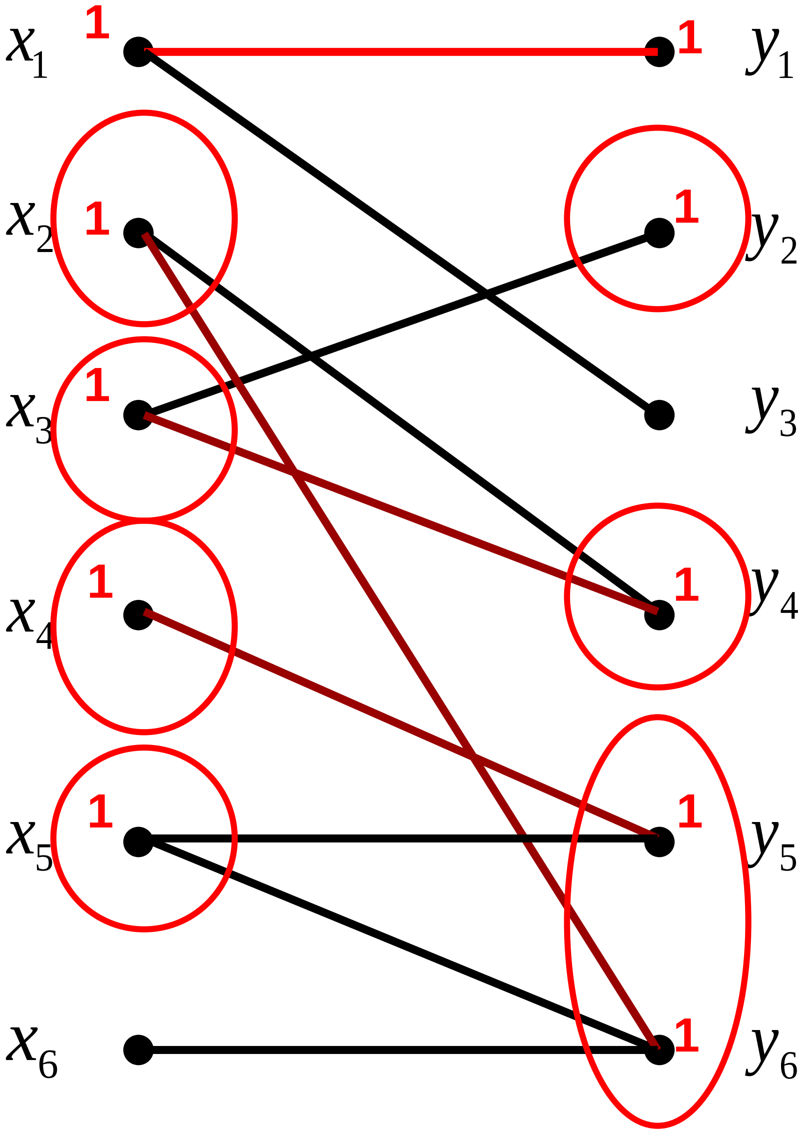


$$U \leftarrow \{x_5, x_4, x_2\}, \quad V \leftarrow \{y_5, y_6\}$$

$$\Gamma(U) = \{y_5, y_6, y_4\}$$

$$\Gamma(U) - V = \{y_4\}$$

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

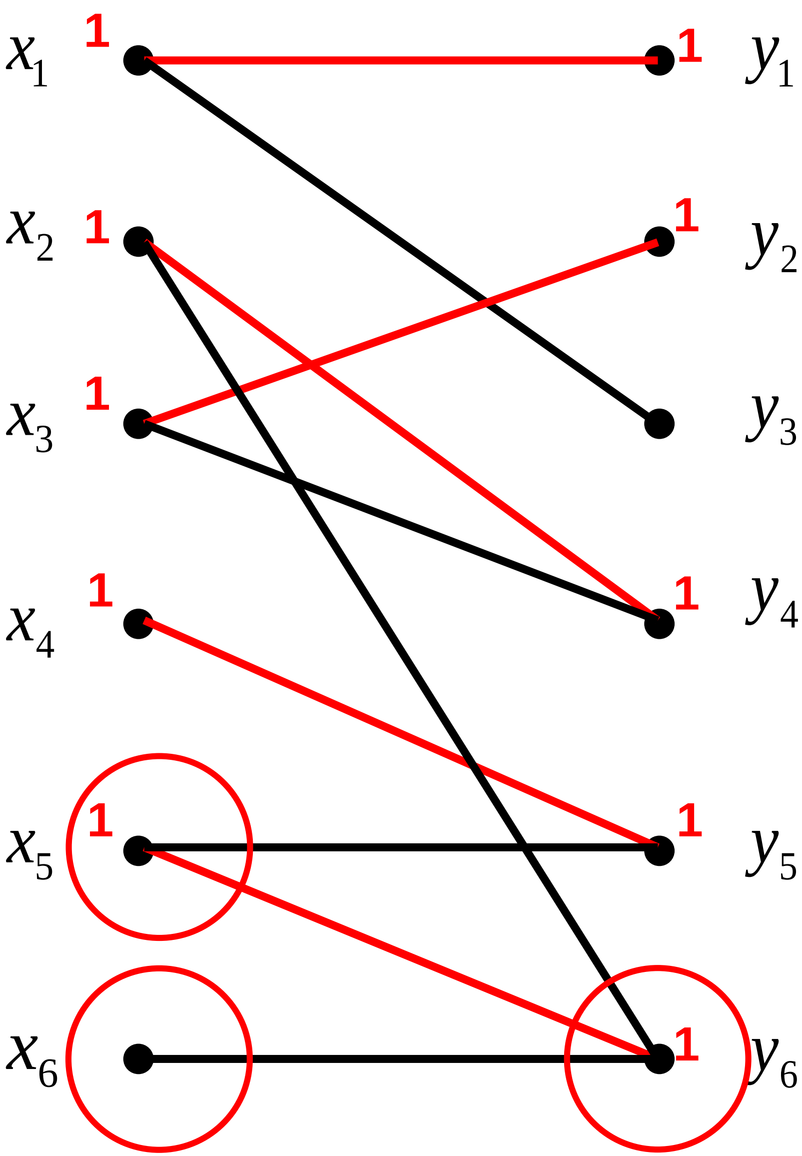


$$U \leftarrow \{x_5, x_4, x_2, x_3\}, \quad V \leftarrow \{y_5, y_6, y_4\}$$

$$\Gamma(U) = \{y_5, y_6, y_4, y_2\}$$

$$\Gamma(U) - V = \{y_2\}$$

$$M \leftarrow M \oplus P$$

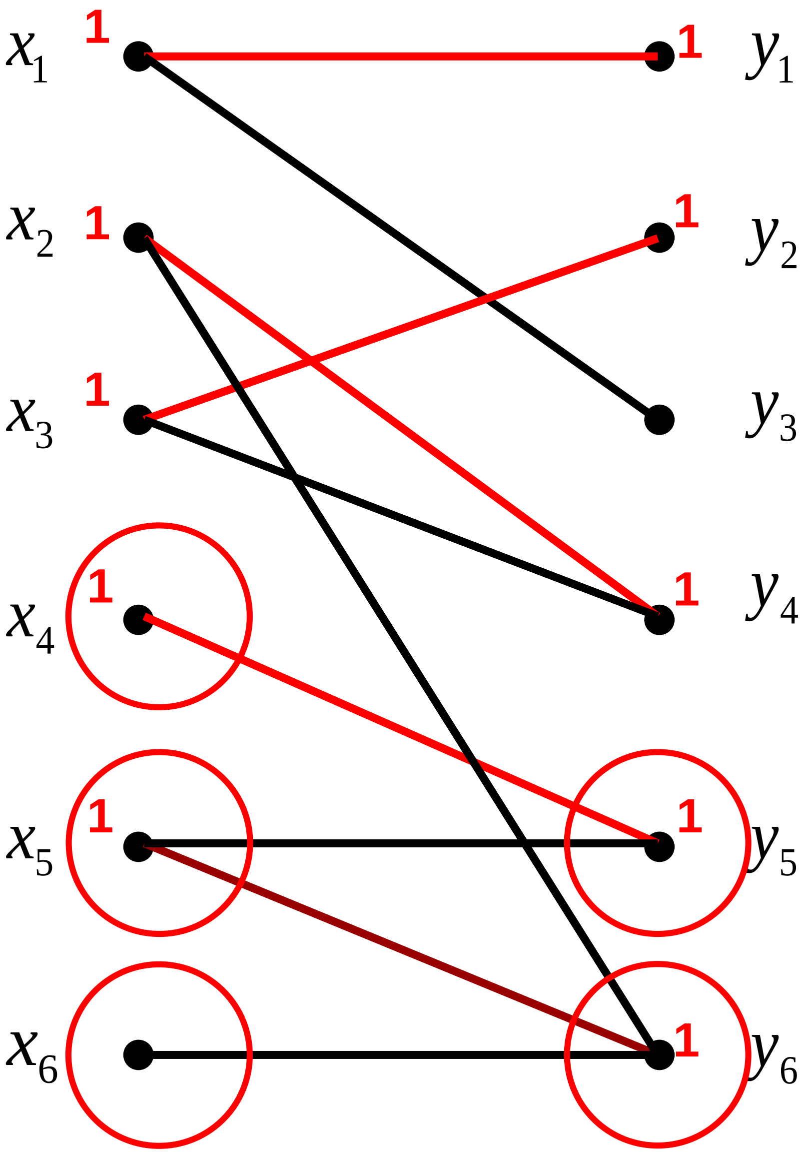


$$U \leftarrow \{x_6\}, \;\; V \leftarrow \Phi$$

$$\Gamma(U) = \{y_6\}$$

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

$$U \leftarrow \{x_6, x_5\}, \;\; V \leftarrow \{y_6\}$$

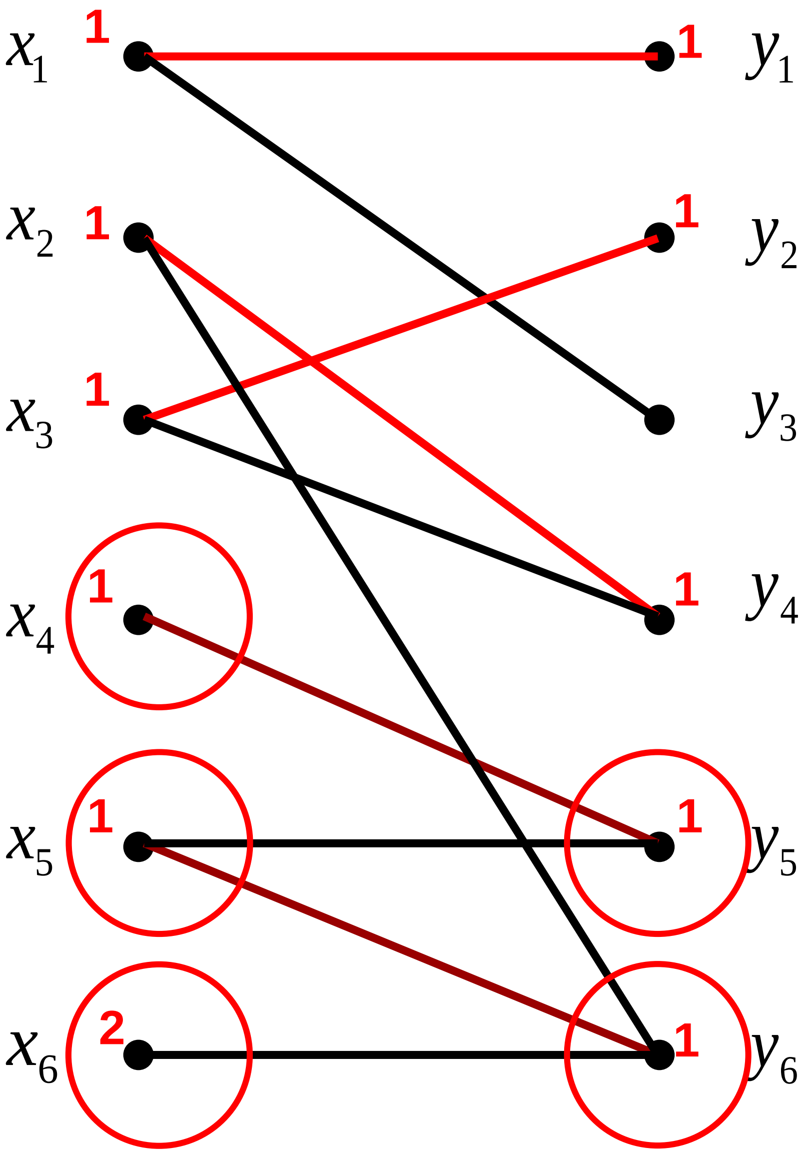


$$U \leftarrow \{x_6, x_5\}, \quad V \leftarrow \{y_6\}$$

$$\Gamma(U) = \{y_6, y_5\}$$

$$\Gamma(U) - V = \{y_5\}$$

$$U \leftarrow \{x_6, x_5, x_4\}, \quad V \leftarrow \{y_6, y_5\}$$



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

$$\Gamma(U) = \{y_6, y_5\}$$

$$\Gamma(U) - V = \phi$$

说明无法搜索到可增广路径

1965年匈牙利人Edmonds提出

并以他的祖国名字命名

二分图的最大匹配-小结

- 基本概念：
  - 匹配、(不)饱和点、交互道路、可增广道路、最大匹配
  - 判断最大匹配的充要条件
  - 求二分图的最大匹配：匈牙利算法

主要内容

- 5.1 二分图的最大匹配
- **5.2 完全匹配**
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

完全匹配

- 例：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。

  问：是否可以安排得当，使每位都有认识的人做舞伴

完全匹配

- 二分图G=(X,Y,E)的最大匹配M所包含的边数为|M|（显然不会超过|X|）
  - 若|M|=|X|，则称M为完全匹配
  - 若|M|=|X|=|Y|，则称M为完美匹配

思考：

- 二分图如何才会具有完全匹配？

完全匹配

- 定理5.2.1(Hall定理) 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$\Gamma(A) \geq A$

证明：

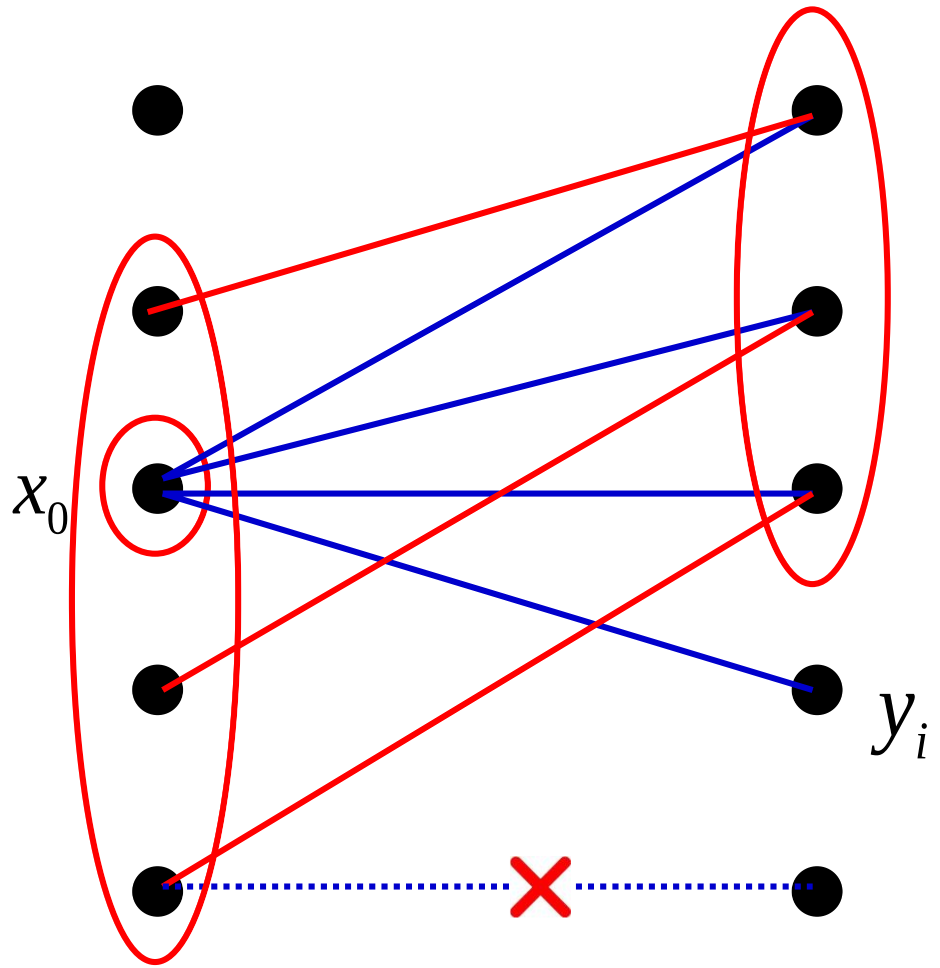
必要性： 存在完全匹配 → $\Gamma(A) \geq A$

– 假定存在子集$A \subseteq X$，使$\Gamma(A) < A$

– 则A中结点一定无法全部匹配，因此X到Y不可能有完全匹配

充分性：　$\Gamma(A) \geq A$ → 存在完全匹配

反证法：任何一个最大匹配M，它都不是完全匹配



这些交互道路中，一定不存在可增广道路！

交互道中分属于X和Y的结点集合为X'和Y'，　$\Gamma(X') = Y'$　$|\Gamma(X')| < |X'|$

充分性： $\Gamma(A) \ge A$ → 存在完全匹配

- 假设某最大匹配M不是完全匹配，$x_0 \in X$ 为非饱和点
- 假如 $\Gamma(x_0) = \Phi$，则令 $A = x_0$，存在 $\Gamma(A) < A$，矛盾
- 若 $\Gamma(x_0) \ne \Phi$，若存在 $y_i \in \Gamma(x_0)$，且 y_i 为非饱和点，则与M是最大匹配矛盾
- 因此，对于所有 $y_i \in \Gamma(x_0)$，必定全部是饱和点
- 根据匈牙利算法的搜索过程，可找到一切以 x_0 为起点的相对于M的交互道路；这些交互道路中，一定不存在可增广道路，否则与M是最大匹配矛盾
- 则交互道中分属于X和Y的结点集合为X'和Y'，必然有 $\Gamma(X') = Y'$ 且 $|\Gamma(X')| < |X'|$

证毕！

完全匹配

- 定理5.2.1(Hall定理) 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$\Gamma(A) \geq A$

完全匹配

- 推论5.2.1 若二分图G=(X,Y,E)的每个结点$x_i \in X$，都有 $d(x_i) \geq k$，每个结点 $y_i \in Y$，都有$d(y_i) \leq k$，那么X到Y存在完全匹配

证明：

- 对于任意子集$A \subseteq X$，设它的结点总共与m条边关联，于是可知$m = \sum_{x_i \in A} d(x_i)$
- 这m条边同时与Y中的$|\Gamma(A)|$个结点关联，于是可知 $m \leq k \cdot |\Gamma(A)|$
- 因此必然有$\Gamma(A) \geq A$

证毕！

Tsinghua University

完全匹配

- 例：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。

  问：是否可以安排得当，使每位都有认识的人做舞伴

完全匹配

思考：

- 对于二分图来说：
  - 完全匹配　　　　　最大匹配
  - 最大匹配　　　　　完全匹配
  - 最大匹配与完全匹配会有什么样的差别？

Tsinghua University

完全匹配

- 定理5.2.2 在二分图G=(X,Y,E)中，X到Y的最大匹配的边数是 $|X|-\delta(G)$

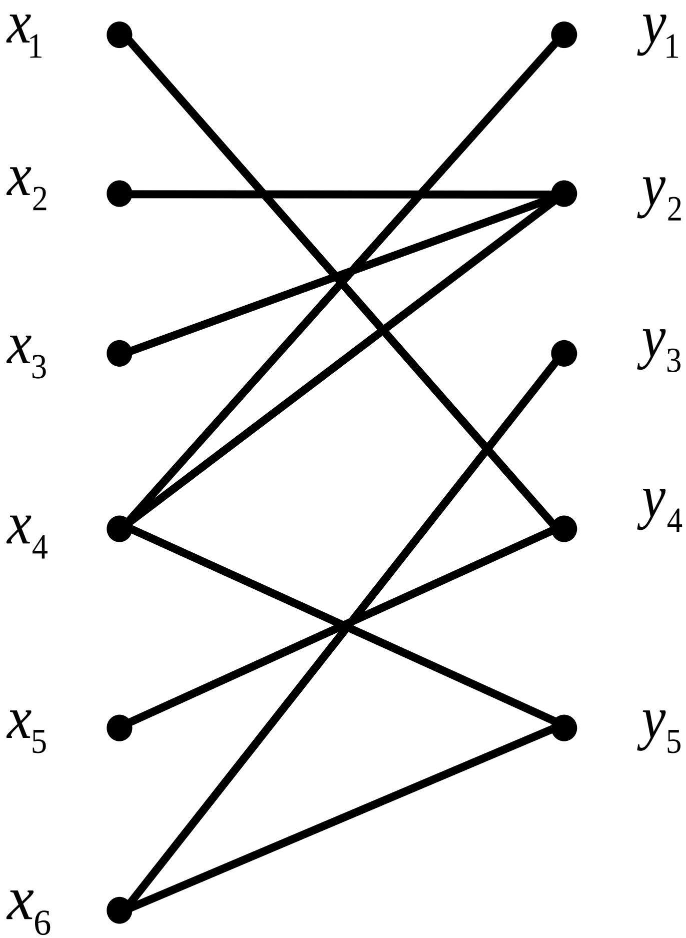
$$\delta(G)=\max_{A\subseteq X}\delta(A),\ \ \delta(A)=|A|-|\Gamma(A)|,\ \ \delta(A)\geq 0$$

完全匹配

思考：

- 如何通过代数的方法分析二分图的匹配问题？

- 二分图的代数表示方法：
  - 邻接矩阵

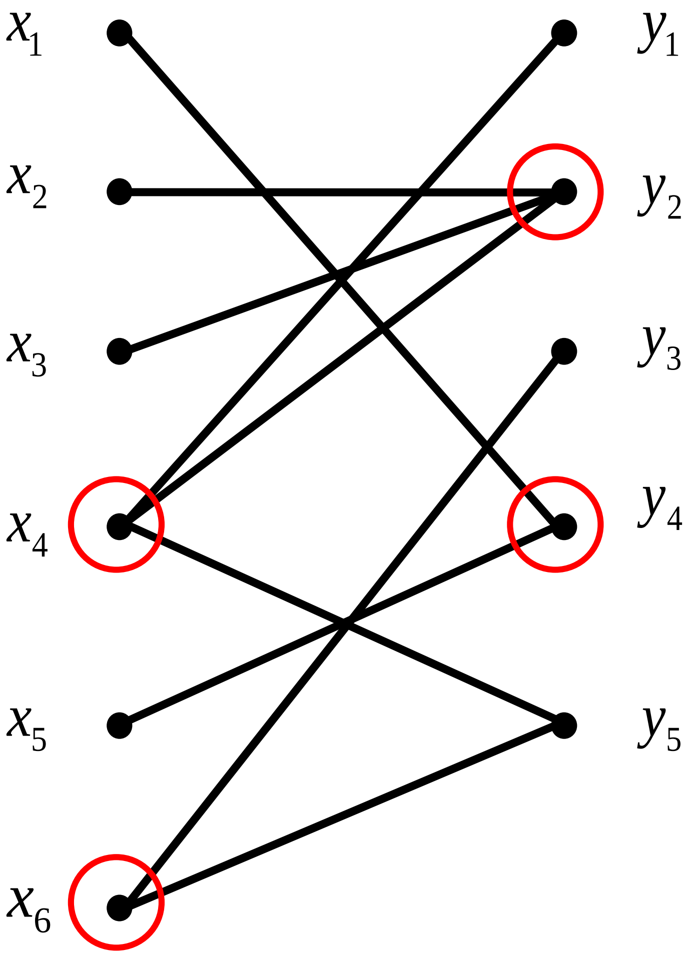


$$\begin{array}{cc}
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{array} &
\left[\begin{array}{c|c}
\mathbf{0} & \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{matrix} \\ \hline
\begin{matrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{matrix} & \mathbf{0}
\end{array}\right] \\
 & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & y_1 & y_2 & y_3 & y_4 & y_5 \end{matrix}
\end{array}$$

显然，最大匹配数r，是邻接矩阵中不在同行同列中非零元的最多个数

$$r \le \min\left(|X|,|Y|\right)$$

$$\begin{array}{cc}
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} &
\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\
 & \begin{matrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{matrix}
\end{array}$$



$$\begin{array}{c}
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array}
\left[\begin{array}{ccccc}
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1
\end{array}\right] \\
\begin{array}{ccccc} \quad y_1 & y_2 & y_3 & y_4 & y_5 \end{array}
\end{array}$$

适当选取某些行和列，可以包含A中全部非零元，这称之为A的覆盖

对于能覆盖非零元最少的行、列数和，称为它的最小覆盖数，记为S

$$s \le \min\left(\left|X\right|,\left|Y\right|\right)$$

完全匹配

思考：

- 最大匹配数 r 是A中不在同行同列非零元最多的个数
- 最小覆盖数 s 是A中能够涵盖所有非零元的最少行列数之和
- r 和 s 会有什么内在的联系？

完全匹配

- 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

证明：

(1) 证$s \geq r$（最小覆盖数≥最大匹配数）

  - 由于r个不同行、不同列的非零元需要至少r个行、列进行覆盖
  - 而s个行、列可以盖住所有的非零元
  - 因此$s \geq r$

完全匹配

(2) 证$r \geq s$（最大匹配数≥最小覆盖数）

- 不失一般性，设A的最小覆盖盖住了A的c行、d列，即s = c + d
- 设此c行对应的结点子集是X_c，其余为$X - X_c$
- 设d列对应的结点子集是Y_d，其余为$Y - Y_d$
- 将矩阵A进行行、列变换，得到如下形式矩阵

$$
\begin{array}{r c}
 & \begin{array}{cc} \quad & \quad \end{array} \\
\begin{array}{r} X_c \\ \\ X - X_c \end{array} & \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \\
 & \begin{array}{cc} Y - Y_d & \qquad Y_d \end{array}
\end{array}
$$

$$
\begin{array}{c|cc}
 & Y-Y_d & Y_d \\
\hline
\begin{array}{c} X_c \\ X-X_c \end{array} & \multicolumn{2}{c}{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}
\end{array}
$$

在$G'=\left(X_c, Y-Y_d, E'\right)$中，$\forall V' \subseteq X_c, \left|\Gamma\left(V'\right)\right| \geq \left|V'\right|$

从 X_c 到 $Y-Y_d$ 存在完全匹配 M_1，且$\left|M_1\right|=c$

在$G''=\left(Y_d, X-X_c, E''\right)$中，$\forall V'' \subseteq Y_d, \left|\Gamma\left(V''\right)\right| \geq \left|V''\right|$

从Y_d 到 $X-X_c$ 存在完全匹配 M_2，且$\left|M_2\right|=d$

显然，$M'=M_1+M_2$仍然是A的一个匹配，且$|M'|=c+d=s\leq r$

- 观察A_{11}，如果在A_{11}中任取p行，这p行在 $Y-Y_d$ 列中，需要至少p列才能覆盖其中非零元
  - 否则若存在q列$(q<p)$能够覆盖这p行的全部非零元，则将原覆盖中此p行替换为q列，将得到A的更小覆盖，与前提矛盾。
- 这意味着对于 X_c 的任意子集V'，它在$Y-Y_d$中的邻接结点集为 $\Gamma(V')$，必有 $|\Gamma(V')|\ge|V'|$ 。根据Hall定理，X_c 到 $Y-Y_d$ 存在完全匹配M_1，且 $|M_1|=c$
- 同理，从Y_d到$X-X_c$ 存在完全匹配M_2，且 $|M_2|=d$
- 显然， $M_1\cup M_2$ 仍然是A的一个匹配且 $|M_1\cup M_2|=c+d=s$
- 因此r是最大匹配数，必有 $r\geq s$

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

证毕！

完全匹配

- 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 $r = s$

Tsinghua University

完全匹配-小结

- 基本概念：
  - 完全匹配，完美匹配

- 二分图存在完全匹配的充要条件：Hall定理

- 二分图完全匹配同最小覆盖的关系

主要内容

- 5.1 二分图的最大匹配
- 5.2 完全匹配
- **5.3 最佳匹配及其算法**
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

最佳匹配及其算法

- 二分图中最大匹配、完全匹配是在边权值为1的情况下的匹配问题
- 如果边权值不为1时，匹配问题往往要寻找权值总和最大或最小情况
- 如果边权为非负实数，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做最佳匹配

最佳匹配及其算法

- 例：5项工作由5个人完成，如果用c_{ij}表示i从事工作j的利润，则形成如下矩阵：

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

如果限定每个人只能从事一项工作，那么最大利润应该是

$\max \sum c_{ij}$ ，c_{ij} 不在同行同列

Tsinghua University

最佳匹配及其算法

- 最佳匹配问题的实质
  - 二分图的最大权匹配问题

- 最大权匹配算法（已知利润矩阵C）：

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

1. 在矩阵C的每行选一最大值作为本行的界值$l(x_i)$
每列的界值初值给定为0
构造矩阵B，$B=\left(b_{ij}\right)_{n\times n}$ $b_{ij}=l(x_i)+l(y_j)-c_{ij}$

2. 在B中对0元素进行最小覆盖，覆盖数为r
2.1 若覆盖数r＝n，则 $\sum\left(l(x_i)+l(y_j)\right)$ 即为最大权，结束
2.2 在未覆盖的元素中选取最小非零元δ
对于双重覆盖元, $b_{ij}=b_{ij}+\delta$
对于未覆盖元, $b_{ij}=b_{ij}-\delta$

3. 修改界值
若 x_i 行没被覆盖，则 $l(x_i)=l(x_i)-\delta$
若 y_j 列已被覆盖，则 $l(y_j)=l(y_j)+\delta$
删除覆盖标记，转2

1. 在矩阵C的每行选一最大值作为本行的界值$l(x_i)$

每列的界值初值给定为0

构造矩阵B，$B=\left(b_{ij}\right)_{n\times n}$ $\underline{b_{ij}=l(x_i)+l(y_j)-c_{ij}}$

$$\begin{array}{cc} \begin{array}{c} l(x_1)=9 \\ l(x_2)=8 \\ l(x_3)=7 \\ l(x_4)=6 \\ l(x_5)=8 \end{array} & \begin{bmatrix} 3 & 3 & 6 & 4 & 9 \\ 6 & 4 & 5 & 3 & 8 \\ 7 & 5 & 3 & 4 & 2 \\ 6 & 3 & 2 & 2 & 5 \\ 8 & 4 & 5 & 4 & 7 \end{bmatrix} \\ & \underbrace{\begin{matrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix}}_{l(y_j)=0} \end{array} \Longrightarrow B=\begin{bmatrix} 6 & 6 & 3 & 5 & 0 \\ 2 & 4 & 3 & 5 & 0 \\ 0 & 2 & 4 & 3 & 5 \\ 0 & 3 & 4 & 4 & 1 \\ 0 & 4 & 3 & 4 & 1 \end{bmatrix}$$

2. 在B中对0元素进行最小覆盖，覆盖数为r

2.1 若覆盖数r = n，则 $\sum\left(l(x_i)+l(y_j)\right)$ 即为最大权，结束

2.2 在未覆盖的元素中选取最小非零元δ

对于双重覆盖元，$b_{ij}=b_{ij}+\delta$

对于未覆盖元，$b_{ij}=b_{ij}-\delta$

$$\begin{array}{l} l(x_1)=9 \\ l(x_2)=8 \\ l(x_3)=7 \\ l(x_4)=6 \\ l(x_5)=8 \end{array}
\begin{array}{c}\begin{bmatrix} \boxed{6} & 6 & 3 & 5 & \boxed{0} \\ \boxed{2} & 4 & 3 & 5 & \boxed{0} \\ \boxed{0} & 2 & 4 & 3 & \boxed{5} \\ \boxed{0} & 3 & 4 & 4 & \boxed{1} \\ \boxed{0} & 4 & 3 & 4 & \boxed{1} \end{bmatrix} \\ \underbrace{\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}}_{l(y_j)=0} \end{array}
\longrightarrow
\begin{array}{l} l(x_1)=9 \\ l(x_2)=8 \\ l(x_3)=7 \\ l(x_4)=6 \\ l(x_5)=8 \end{array}
\begin{array}{c}\begin{bmatrix} \boxed{6} & 4 & 1 & 3 & \boxed{0} \\ \boxed{2} & 2 & 1 & 3 & \boxed{0} \\ \boxed{0} & 0 & 2 & 1 & \boxed{5} \\ \boxed{0} & 1 & 2 & 2 & \boxed{1} \\ \boxed{0} & 2 & 1 & 2 & \boxed{1} \end{bmatrix} \\ \underbrace{\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}}_{l(y_j)=0} \end{array}$$

3. 修改界值

若 x_i 行没被覆盖，则 $l(x_i)=l(x_i)-\delta$

若 y_j 列已被覆盖，则 $l(y_j)=l(y_j)+\delta$

删除覆盖标记，转2

$$
\begin{array}{rc}
\begin{array}{r}
l(x_1)=7\\
l(x_2)=6\\
l(x_3)=5\\
l(x_4)=4\\
l(x_5)=6
\end{array}
&
\begin{bmatrix}
6 & 4 & 1 & 3 & 0\\
2 & 2 & 1 & 3 & 0\\
0 & 0 & 2 & 1 & 5\\
0 & 1 & 2 & 2 & 1\\
0 & 2 & 1 & 2 & 1
\end{bmatrix}\\
 & \underbrace{\mathbf{2}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{2}}_{l(y_j)}
\end{array}
$$

2. 在B中对0元素进行最小覆盖，覆盖数为r

2.1 若覆盖数r = n，则 $\sum\left(l(x_i)+l(y_j)\right)$ 即为最大权，结束

2.2 在未覆盖的元素中选取最小非零元δ

对于双重覆盖元，$b_{ij}=b_{ij}+\delta$

对于未覆盖元，$b_{ij}=b_{ij}-\delta$

$$
\begin{array}{cc}
\begin{array}{c} l(x_1)=7 \\ l(x_2)=6 \\ l(x_3)=5 \\ l(x_4)=4 \\ l(x_5)=6 \end{array} &
\begin{bmatrix} 6 & 4 & 1 & 3 & 0 \\ 2 & 2 & 1 & 3 & 0 \\ 0 & 0 & 2 & 1 & 5 \\ 0 & 1 & 2 & 2 & 1 \\ 0 & 2 & 1 & 2 & 1 \end{bmatrix} \\
 & \underbrace{\begin{matrix} \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{2} \end{matrix}}_{l(y_j)}
\end{array}
\longrightarrow
\begin{array}{cc}
\begin{array}{c} l(x_1)=7 \\ l(x_2)=6 \\ l(x_3)=5 \\ l(x_4)=4 \\ l(x_5)=6 \end{array} &
\begin{bmatrix} 6 & 3 & 0 & 2 & 0 \\ 2 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 1 & 6 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \\
 & \underbrace{\begin{matrix} \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{2} \end{matrix}}_{l(y_j)}
\end{array}
$$

3. 修改界值

若 x_i 行没被覆盖，则 $l(x_i)=l(x_i)-\delta$

若 y_j 列已被覆盖，则 $l(y_j)=l(y_j)+\delta$

删除覆盖标记，转2

$$
\begin{array}{r}
l(x_1)=6 \\
l(x_2)=5 \\
l(x_3)=5 \\
l(x_4)=3 \\
l(x_5)=5 \\
\\
\end{array}
\begin{array}{c}
\begin{bmatrix}
6 & 3 & 0 & 2 & 0 \\
2 & 1 & 0 & 2 & 0 \\
1 & 0 & 2 & 1 & 6 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1
\end{bmatrix} \\
\underbrace{\mathbf{3} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{3}}_{l(y_j)}
\end{array}
$$

2. 在B中对0元素进行最小覆盖，覆盖数为r

2.1 若覆盖数r = n，则 $\sum\left(l(x_i)+l(y_j)\right)$ 即为最大权，结束

2.2 在未覆盖的元素中选取最小非零元δ

对于双重覆盖元，$b_{ij}=b_{ij}+\delta$

对于未覆盖元，$b_{ij}=b_{ij}-\delta$

$$
\begin{array}{l}
l(x_1)=6\\ l(x_2)=5\\ l(x_3)=5\\ l(x_4)=3\\ l(x_5)=5
\end{array}
\underbrace{\begin{bmatrix}
\boxed{6} & \boxed{3} & \boxed{0} & 2 & \boxed{0}\\
2 & 1 & 0 & 2 & 0\\
1 & 0 & 2 & 1 & 6\\
0 & 0 & 1 & 1 & 1\\
0 & 1 & 0 & 1 & 1
\end{bmatrix}}_{\substack{\mathbf{3\ \ 0\ \ 0\ \ 0\ \ 3}\\ l(y_j)}}
\Rightarrow
\begin{array}{l}
l(x_1)=6\\ l(x_2)=5\\ l(x_3)=5\\ l(x_4)=3\\ l(x_5)=5
\end{array}
\underbrace{\begin{bmatrix}
6 & 3 & 0 & 1 & 0\\
2 & 1 & 0 & 1 & 0\\
1 & 0 & 2 & 0 & 6\\
0 & 0 & 1 & 0 & 1\\
0 & 1 & 0 & 0 & 1
\end{bmatrix}}_{\substack{\mathbf{3\ \ 0\ \ 0\ \ 0\ \ 3}\\ l(y_j)}}
$$

3. 修改界值

若 x_i 行没被覆盖，则 $l(x_i)=l(x_i)-\delta$

若 y_j 列已被覆盖，则 $l(y_j)=l(y_j)+\delta$

删除覆盖标记，转2

$$
\begin{array}{l c}
l(x_1)=5 & \\
l(x_2)=4 & \\
l(x_3)=4 & \\
l(x_4)=2 & \\
l(x_5)=4 &
\end{array}
\begin{bmatrix}
\boxed{6} & \boxed{3} & \boxed{0} & 1 & \boxed{0} \\
\boxed{2} & \boxed{1} & \boxed{0} & 1 & \boxed{0} \\
\boxed{1} & \boxed{0} & \boxed{2} & 0 & \boxed{6} \\
\boxed{0} & \boxed{0} & \boxed{1} & 0 & \boxed{1} \\
\boxed{0} & \boxed{1} & \boxed{0} & 0 & \boxed{1}
\end{bmatrix}
$$

$$
\underbrace{\mathbf{4}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{0}\quad \mathbf{4}}_{l(y_j)}
$$

2. 在B中对0元素进行最小覆盖，覆盖数为r

2.1 若覆盖数r = n，则 $\sum\left(l(x_i)+l(y_j)\right)$ 即为最大权，结束

此时，可以得到一个最大权匹配方案：

$$\{c_{13}, c_{25}, c_{34}, c_{42}, c_{51}\}$$

$$\sum\left(l(x_i)+l(y_j)\right)=29$$

$$\begin{array}{c} l(x_1)=5 \\ l(x_2)=4 \\ l(x_3)=4 \\ l(x_4)=2 \\ l(x_5)=4 \end{array}
\begin{bmatrix} 6 & 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 6 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\underbrace{\mathbf{4} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{4}}_{l(y_j)}$$

1. 在矩阵C的每行选一最大值作为本行的界值$l(x_i)$
每列的界值初值给定为0

构造矩阵B，$B=(b_{ij})_{n\times n}$ $b_{ij}=l(x_i)+l(y_j)-c_{ij}$ **构造矩阵**

2. 在B中对0元素进行最小覆盖，覆盖数为r

2.1 若覆盖数r = n，则 $\sum(l(x_i)+l(y_j))$ 即为最大权，结束

2.2 在未覆盖的元素中选取最小非零元δ **最小覆盖调整元素**

对于双重覆盖元，$b_{ij}=b_{ij}+\delta$

对于未覆盖元，$b_{ij}=b_{ij}-\delta$

3. 修改界值

若 x_i 行没被覆盖，则 $l(x_i)=l(x_i)-\delta$

若 y_j 列已被覆盖，则 $l(y_j)=l(y_j)+\delta$ **修改界值**

删除覆盖标记，转2

最佳匹配及其算法

- 定理5.3.1 最大权匹配算法的结果是矩阵C的最大权匹配。

证明：

- 算法最初所选取的矩阵B，满足

$$b_{ij} = l(x_i) + l(y_j) - c_{ij} \geq 0$$

假如，我们在B每一次调整的过程中，能够保持这个关系，会有什么现象？

$$b_{ij}=l(x_i)+l(y_j)-c_{ij}\geq 0 \rightarrow l(x_i)+l(y_j)\geq c_{ij}$$

$$\begin{array}{c|c}
\begin{matrix} l(x_1) \\ l(x_2) \\ l(x_3) \\ l(x_4) \\ l(x_5) \end{matrix} & \begin{bmatrix} 3 & 3 & 6 & 4 & 9 \\ 6 & 4 & 5 & 3 & 8 \\ 7 & 5 & 3 & 4 & 2 \\ 6 & 3 & 2 & 2 & 5 \\ 8 & 4 & 5 & 4 & 7 \end{bmatrix} \\
 & \begin{matrix} l(y_1) & l(y_2) & l(y_3) & l(y_4) & l(y_5) \end{matrix}
\end{array}$$

观察所有不同行、列的C_{ij}之和

$$\max\sum c_{ij} = \sum\left(l(x_i)+l(y_j)\right)$$

最佳匹配及其算法

- 算法最初所选取的矩阵B，满足

$$b_{ij} = l(x_i) + l(y_j) - c_{ij} \geq 0$$

- 我们在B每一次调整的过程中，保持这个关系
- 在 $l(x_i)$和$l(y_j)$ 的选择上，最初可选取一个上限值，但是保证在每一次运算后这个上限值单调递减，直至收敛至最小值

  (当b_{ij}恰好在各不同行、列都有0值时)
- 那么，究竟如何在运算过程中即保持上述关系，又确保上限值单调递减？

清華大學 Tsinghua University

最佳匹配及其算法

- $l(x_i)$和$l(y_j)$的初值选取可确保是上限
- 不失一般性，我们不妨设某次运算之后，盖住了B矩阵的c行、d列
- 如果算法没有结束，则 $c+d<n$
- 此时，δ是未覆盖的最小元

- 对于b_{ij}来说

若x_i行没被覆盖，则 $l(x_i)=l(x_i)-\delta$

若y_j列已被覆盖，则 $l(y_j)=l(y_j)+\delta$

b_{ij}二次覆盖：$b_{ij}^{*}=l^{*}(x_i)+l^{*}(y_j)-c_{ij}=\underline{l(x_i)}+\underline{l(y_j)+\delta}-c_{ij}=b_{ij}+\delta>0$

b_{ij}只列覆盖：$b_{ij}^{*}=l^{*}(x_i)+l^{*}(y_j)-c_{ij}=\underline{l(x_i)-\delta}+\underline{l(y_j)+\delta}-c_{ij}=b_{ij}\geq 0$

b_{ij}只行覆盖：$b_{ij}^{*}=l^{*}(x_i)+l^{*}(y_j)-c_{ij}=l(x_i)+l(y_j)-c_{ij}=b_{ij}\geq 0$

b_{ij}未覆盖：$b_{ij}^{*}=l^{*}(x_i)+l^{*}(y_j)-c_{ij}=l(x_i)-\delta+l(y_j)-c_{ij}=b_{ij}-\delta\geq 0$

- 此时，

$$\begin{aligned}\sum\left(l^{*}(x_i)+l^{*}(y_j)\right)&=\sum\left(l(x_i)+l(y_j)\right)-(n-c)\cdot\delta+d\cdot\delta\\&=\sum\left(l(x_i)+l(y_j)\right)-(n-c-d)\cdot\delta\end{aligned}$$

- 界值和单调减，直至最小值

证毕！

最佳匹配及其算法

- 最小权匹配算法（已知成本矩阵C）：

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

最佳匹配及其算法-小结

- 最大权匹配算法
- 最大权匹配算法正确性证明过程
- 最小成本算法（转化为最大权匹配算法）

主要内容

- 5.1 二分图的最大匹配
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

作业

- 课后：1，2，3，8
- 选作：5，7

- 练习题：有6个字符串be，ef，ab，cd，bc，ad，问是否可以用串中一个字母代表该串而不产生混淆？若可以，请给出一个方案。

Tsinghua University