面向计算机科学的离散数学

图论—平面图与色数

苏航
suhangss@mail.tsinghua.edu.cn
清华大学 计算机系

第五章 匹配与网络流

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

首尾相接：道路、回路
非首尾相接：树，割集
继续寻找特殊情况？

两人一组：两两配对？
三三配对？

相异代表系

- 课程分配问题：每门课程都有几位教师可以讲授

课程	教师
1	A C F
2	C D E G
3	A C
4	A F
5	B E G
6	C F

组合复杂度！

- 不考虑一个教授讲一门课程的限制，所有可能性笛卡尔积=288种可能

相异代表系

- 设 $S_1, S_2, \cdots, S_n$ 是集合的有限序列，如果有 $x_i \in S_i, i=1,2,\cdots,n$，并且各个元素 x_i 都不相同，则称该序列是一个相异代表系。

- 例：$S_1=\{1,2,3\}$

$S_2=\{1,3\}$

$S_3=\{1,3\}$

$S_4=\{3,4,5\}$

$\Rightarrow$ 2, 1, 3, 4

相异代表系

- 课程分配问题：每门课程都有几位教师可以讲授

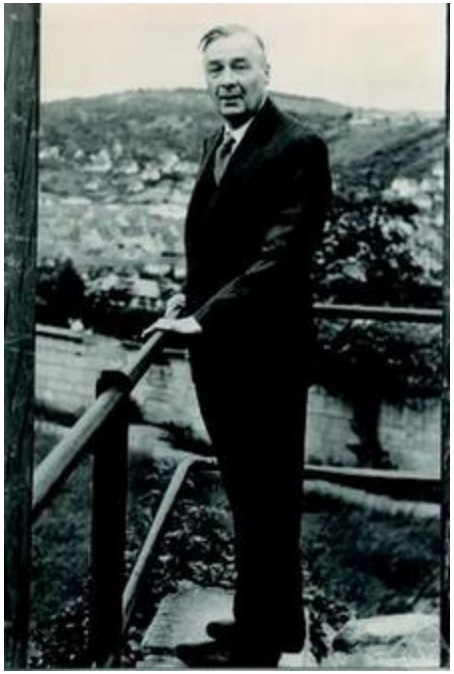
课程	教师	分配
1	A C F	A
2	C D E G	D
3	A C	C
4	A F	F
5	B E G	B
6	C F	?

有相异代表系么？

$$P_1 \cup P_3 \cup P_4 \cup P_6 = \{A, C, F\}$$

霍尔定理

- 英国数学家
- 对近世代数的发展做出了重要贡献
- 英国布雷奇莱庄园密码小组工作



霍尔定理

- 有限集合序列$S_1, S_2, \cdots, S_n$有相异代表系**当且仅当**对$\{1,2,\cdots,n\}$ 的任意子集 I，$S_i(i \in I)$ 的并集所包含的元素个数不少于集合I 所包含的元素的个数。

$$S_1 = \{A, C, E\},\ S_2 = \{A, B\},\ S_3 = \{B, E\}$$

I	S_i 并集	I	S_i 并集
Φ	Φ	1，2	A，B，C，E
1	A, C, E	1，3	A，B，C，E
2	A，B	2，3	A，B，E
3	B，E	1，2，3	A，B，C，E

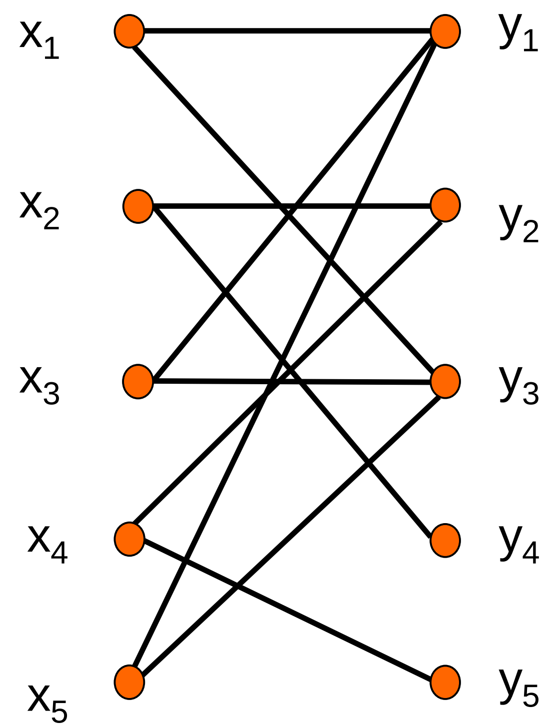
霍尔定理

- 如果对于n个集合，需要检查 2^n个子集
- 指数复杂度

课程	教师
1	A C F
2	C D E G
3	A C
4	A F
5	B E G
6	C F

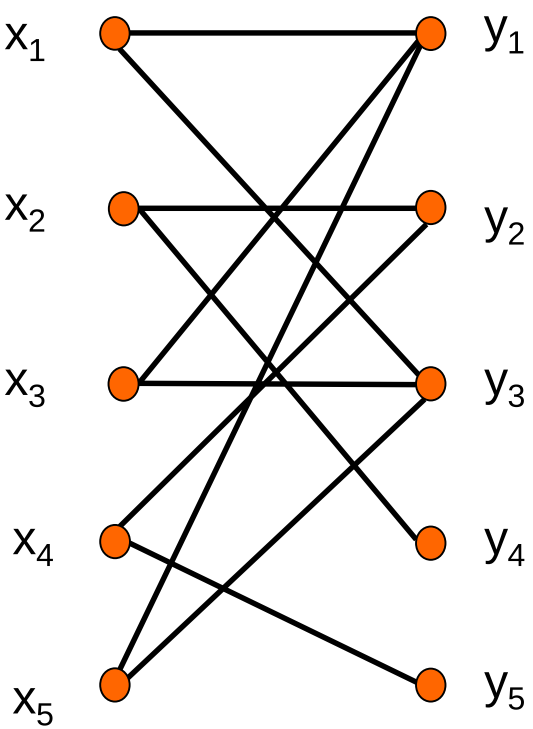
二分图的匹配

- 分配n个人来做m项工作
- 图论建模，G=(V, E)
  - 边(x_i, y_j)表示x_i可以从事y_j
- 约束条件
  - 每个人最多从事其中一项
  - 每项工作只能由其中一个人来承担
- 问怎样才能让更多的人安排上工作？
  - 该问题的图论建模？
- 资源分配问题



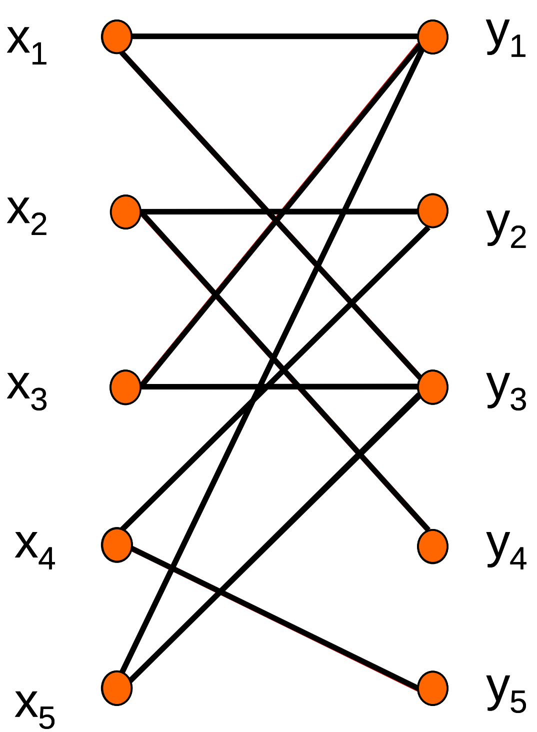
二分图的匹配

- **二分图（偶图）：如果顶级集V可以写成两个不相交的集合V_1和V_2的并集，使得边集E中的每条边都连接V_1中的一个元素和V_2中的一个元素**
- **定义5.1.1：匹配**
  - 令M是图G的边集，若M中任意两条边都没有共同顶点，则称M为G的一个匹配
- **最大匹配：包含边最多的匹配**



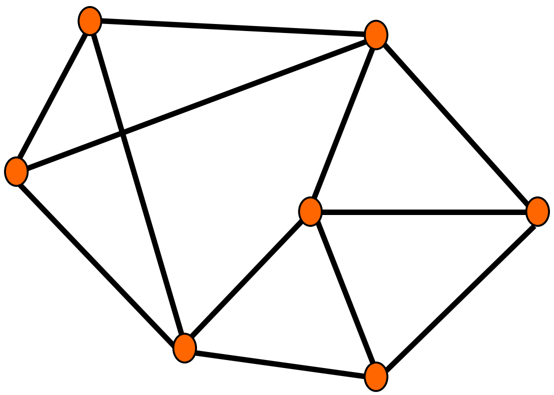
二分图的匹配(2)

- 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也不再允许其它人承担
- 如何描述呢？
  - 因此它相当于用一种颜色比如红色对G的边进行着色，保证每个结点最多只与一条红色边相关联
  - 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**



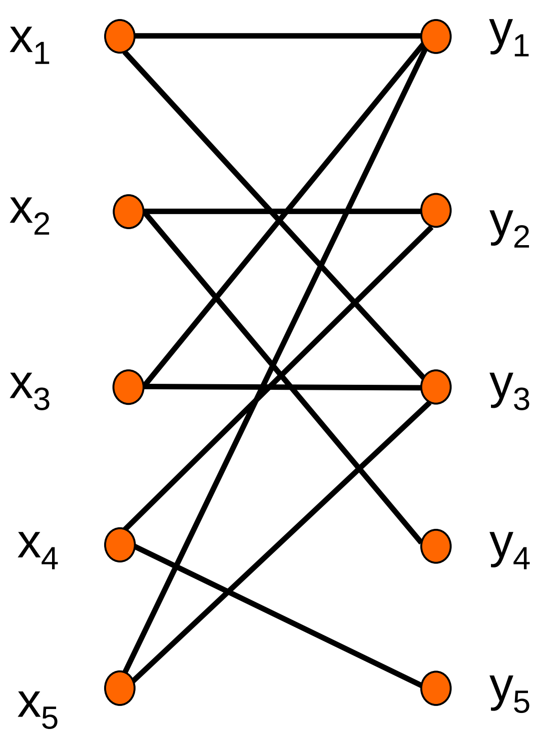
二分图的最大匹配(3)

- **例5.1.2**
  - 二战期间，盟军派飞行员到英国参加对德的空袭，每架飞机需要领航员和飞行员各1人
  - 有些人只会领航，一些人只会驾驶，也有两者均会的
  - 要求二人语言相通
  - 图论建模
    - 以点表示人
    - 边表示二人语言相通并且一人可领航一人可驾驶
  - 那么最多的编队方案就是计算G的一个最大匹配



二分图的最大匹配(4)

- 关键术语定义
  - 图中与M边相关联的结点称为饱和点，否则称为非饱和点
  - 饱和点：x_1, x_2, x_3, x_4, y_1, y_3, y_4, y_5
  - 非饱和点：x_5, y_2
- 最多的任务分配是什么？

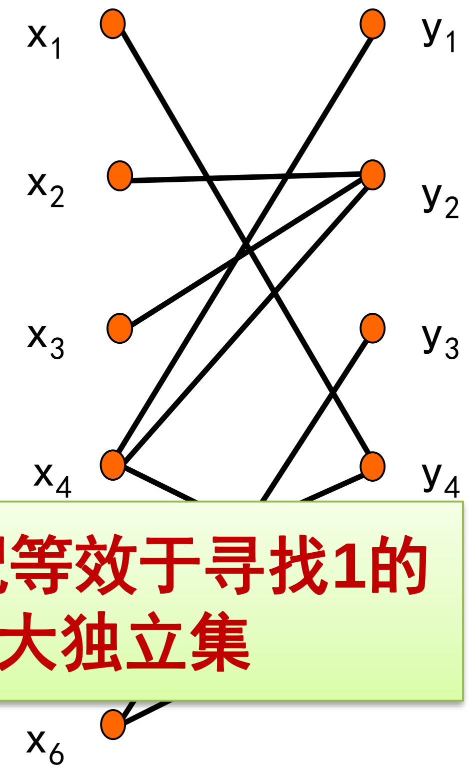


二分图的矩阵表示

- **二分图邻接矩阵**
  - 简化为|X|×|Y|二值矩阵
- **最大匹配是多少？**
  - A中不同行同列的非零元的最大数目
  - 如果矩阵A是p×q矩阵，则最大匹配数≤min(p, q)

A=
0 0 0 1 0
0 1 0 0 0
0 1
1 1
0 0 0 1 0
0 0 1 0 1

邻接矩阵



最大匹配等效于寻找1的最大独立集

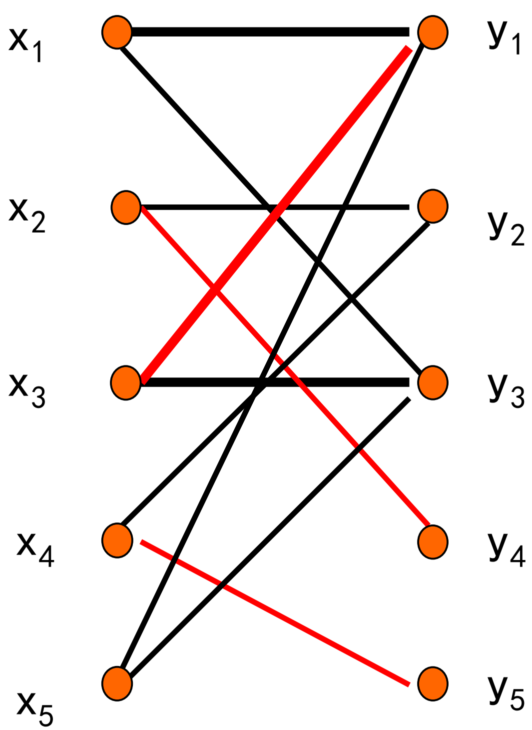
最大匹配是多少？
考虑x_2和x_3、x_1和x_5

如果没有两个元素在同一行/列，该集合是一个独立集

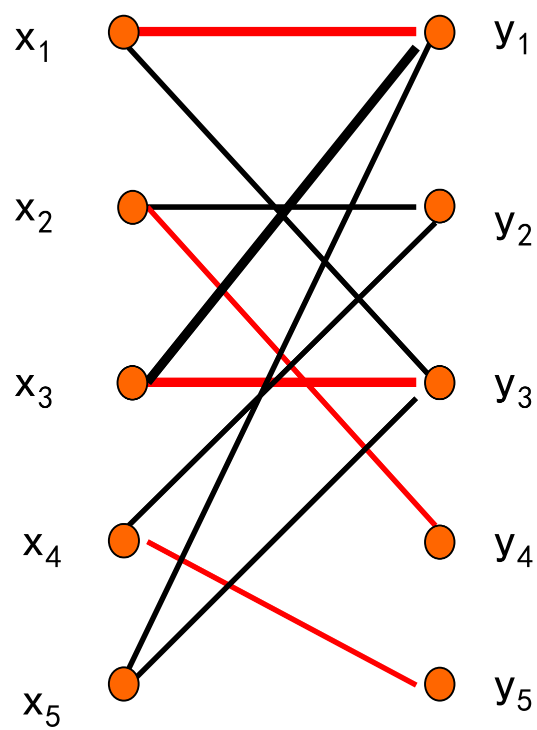
二分图的最大匹配(4)

- **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|≥|M'|，则称M为一个最大匹配

如何求解最大匹配？



黑红黑？

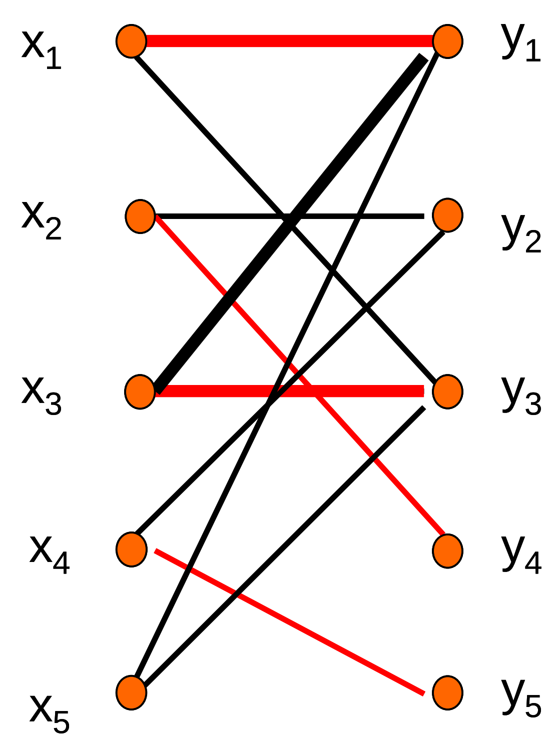


$x_1-y_3-x_3-y_1$？

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（ $x_1-y_1-x_3-y_3$ ）

二分图的最大匹配(5)

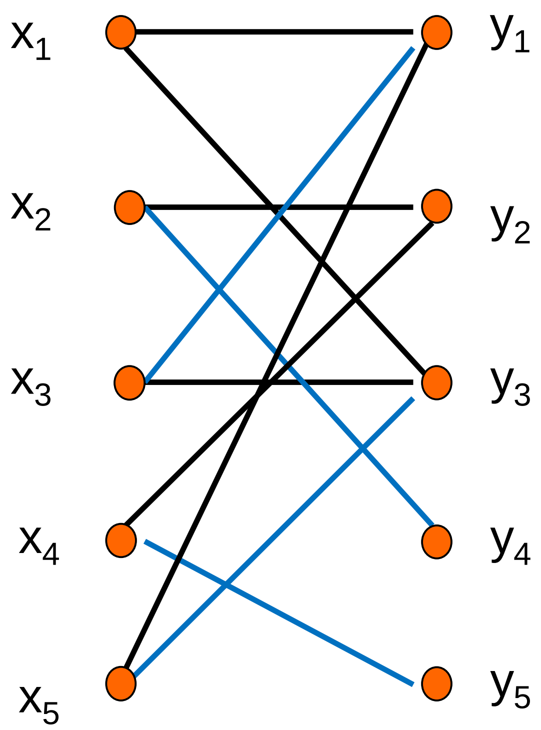
- **定义5.1.4**
  - 设P是G中关于匹配M的一条交互道路
  - 如果P的两个端点是关于M的非饱和点，那么它就称为可增广道路
  - x_1–y_1–x_3–y_3（黑–红–黑）



- **可增广道路的基本特点**
  - 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
  - $M'=P \oplus M$仍然是一个匹配
  - M'使P的两个端点变成了饱和点
  - $|M'|=|M|+1$，即M'是比M更大的匹配

二分图的最大匹配(6)

- **例**
  - 蓝色的边构成图的匹配M
  - 关于该匹配的饱和点是 $\{x_3, y_1, x_2, y_4, x_4, y_5, x_5, y_3\}$
  - 非饱和点是 $\{x_1, y_2\}$
  - 道路（x1，y1，x3，y3，x5）是一条交互道路，但不是可增广道路
  - 判定：M是否为最大匹配？



呼唤定理？

可增广道路存在性与最大匹配的关系？

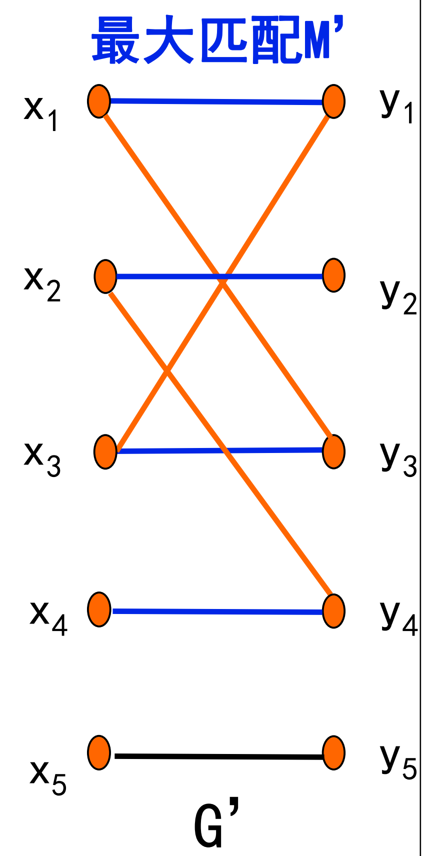
二分图的最大匹配(7)

- 定理5.1.1(Berge 1957)
  - 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(反证法)
    - 若存在M的可增广道路P，则M⊕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(反证法)
    - 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'⊕M，我们逐一分析G'中三种可能的连通支

二分图的最大匹配(8)

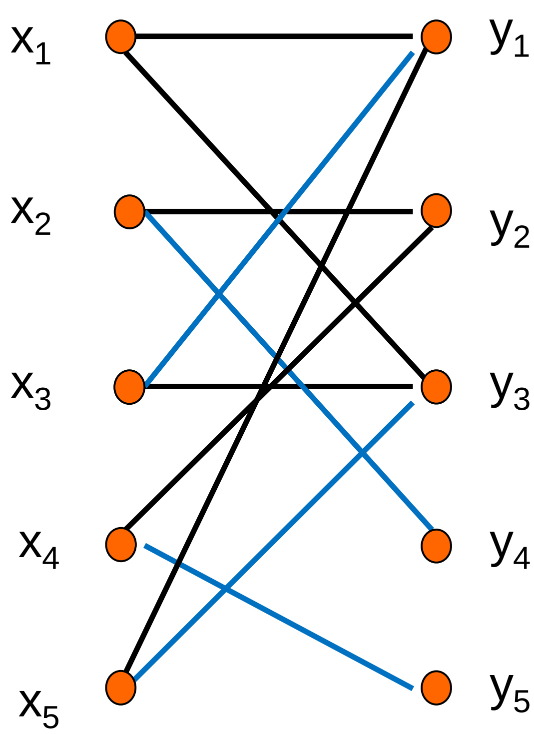
- 要证：M不是最大匹配则存在可增广道路；已有G’ = M’ ⊕M，且M’ 是最大匹配
- 孤立结点v_i（M’ 和M均为不连通的边集）
  - 关联v_i的边$(v_i, v_j) \in$ M’ ∩M，对M’ 和M的贡献相同
- 交互回路（例如x_1-y_1-x_3-y_3）
  - 该回路必为偶回路，其中属于M’ 和M的边数相同
- 交互道路
  - 若不存在增广道路，则|M’ |=|M|，与假设矛盾
  - 若只存在M关于M’ 的增广路/红蓝红，与M’ 是最大匹配矛盾
  - 由于|M’ |>|M|，故必定存在M’ 关于M的可增广交互道路

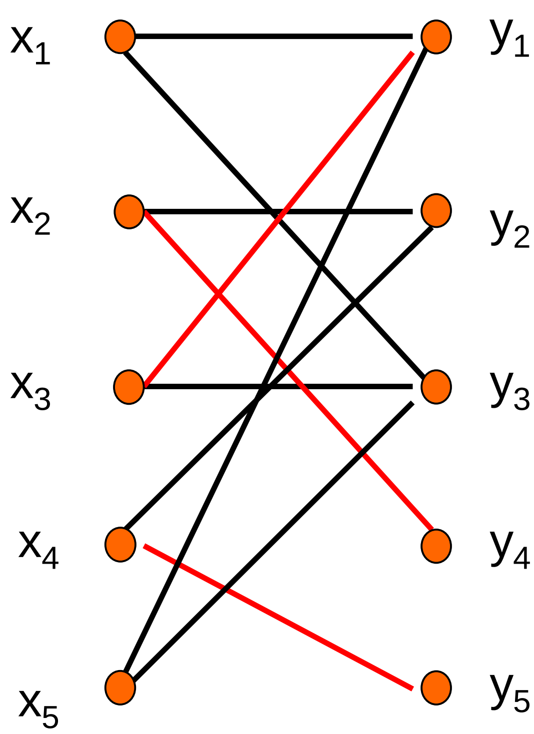
  x_4-y_4-x_2-y_2 ，即G中存在关于M的可增广道路



二分图的最大匹配(9)

- **例**
  - 蓝色边构成的匹配
    - 不存在可增广道路
    - 是最大匹配
  - 红色边构成的匹配
    - 不是最大匹配
    - 存在可增广道路





从判定到求解：最大匹配？

考虑红色图三种可能性：x_2– ；x_1y_3 ；$x_1y_1x_3y_3$

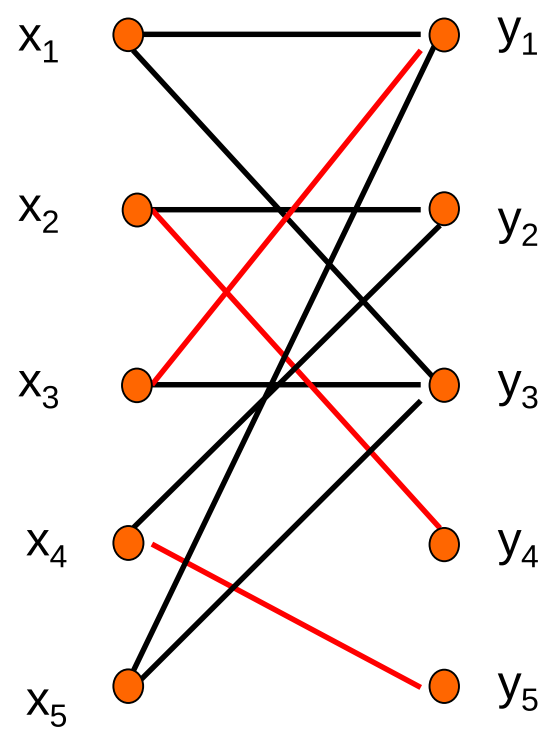
二分图的最大匹配(10)

- 计算二分图的最大匹配算法匈牙利算法
  - 输入为二分图G=(X, Y, E);
  - 结点标记
    - 0: 表示尚未搜索
    - 1: 表示饱和点
    - 2: 表示无法扩大匹配的点（仅对X中的结点）
  - 1、任给初始匹配M，给饱和点“1”标记
  - 2、判断X各结点是否都已有非零标记
    - 2.1 是，M是最大匹配，结束
    - 2.2 否，找一个“0”标记点$x_0 \in X$，开始本次搜索

      令$U \leftarrow \{x_0\}$，为本次搜索的X结点集

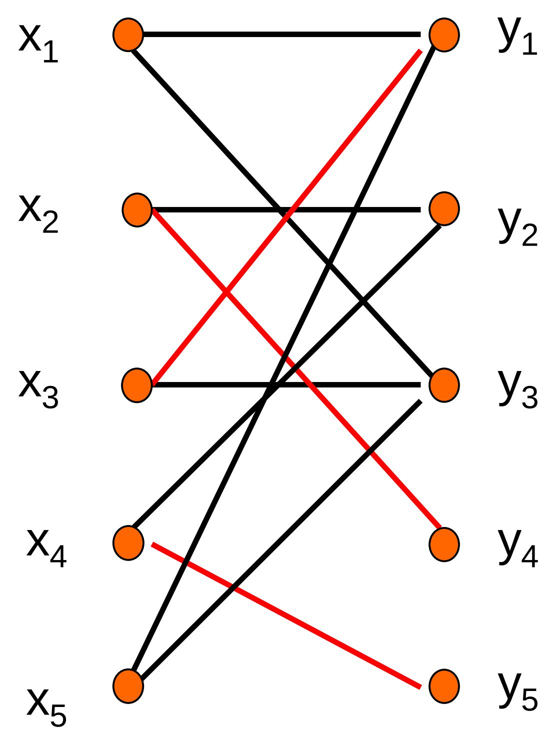
      令$V \leftarrow \Phi$，为本次已检查过的Y结点集

$x_1y_1x_3y_3$



二分图的最大匹配(11)

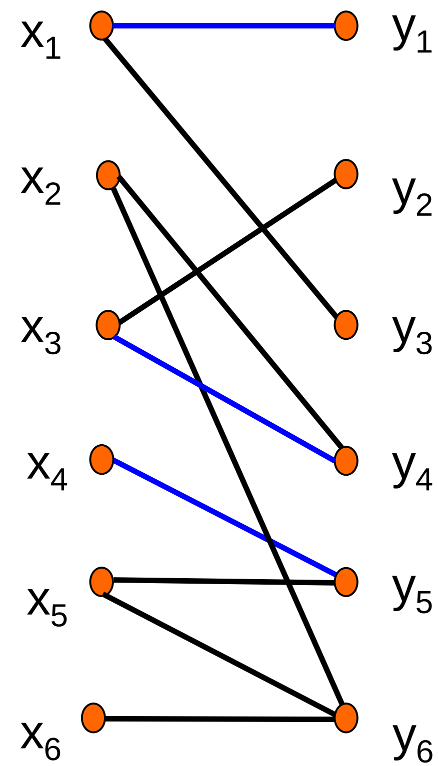
- 3、判断集合U的邻接点集Γ(U)=V?
  - 3.1 是，x_0无扩大匹配，标x_0 "2"，转2
  - 3.2 否，在Γ(U)-V中找一点y_i，判断y_i是否标记" 1"(即饱和点)
    - (1) 非饱和点（如y_3）：
      存在从x_0到y_j的可增广路P
      令M ← M ⊕ P，给x_0，y_i标记1
      转2
    - (2)饱和点（如y_1）：
      则存在x_i使得边(x_i，y_i) ∈ M
      令 U ← U + {x_i}, V ← V + {y_i}, 转3



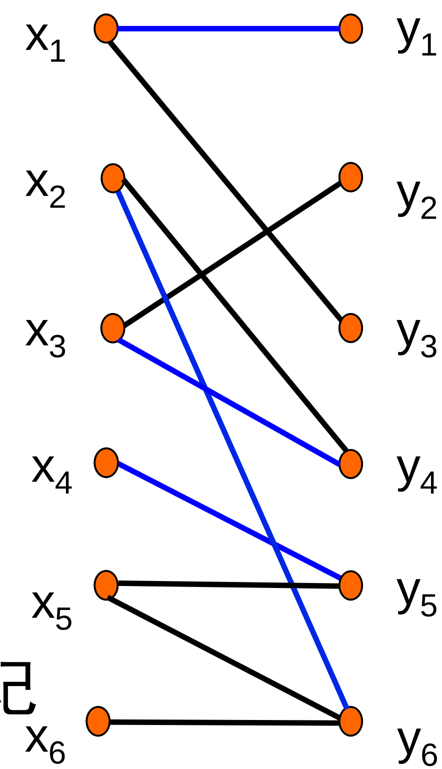
向右非匹配边
向左匹配边

二分图的最大匹配(12)

- 例5.1.3 设初始匹配：
  $M=\{(x_1,y_1),(x_3,y_4),(x_4,y_5)\}$
- 用匈牙利算法求最大匹配
  - 进行节点标记
  - $U=\{x_2\}$，$V=\Phi$（为本次已访问过的Y结点集）
    $\Gamma(U)=\{y_4, y_6\}$，$y_6\in\Gamma(U)-V$，且无标记
  - 得到增广路$P=(x_2, y_6)$
  - 更改节点标记
  - $M=\{(x_1,y_1),(x_3,y_4),(x_4,y_5),(x_2,y_6)\}$



二分图的最大匹配(13)



- $U=\{x_5\}$，访问过的y节点集$V=\Phi$

  $\Gamma(U)=\{y_5, y_6\}$， $y_5\in\Gamma(U)-V$，不可增广
- $U=\{x_5, x_4\}$， $V=\{y_5\}$

  $\Gamma(U)=\{y_5, y_6\}$， $y_6\in\Gamma(U)-V$
- $U=\{x_5, x_4, x_2\}$， $V=\{y_5, y_6\}$

  $\Gamma(U)=\{y_5, y_6, y_4\}$， $y_4\in\Gamma(U)-V$
- $U=\{x_5, x_4, x_2, x_3\}$， $V=\{y_5, y_6, y_4\}$

  $\Gamma(U)=\{y_5, y_6, y_4, y_2\}$， $y_2\in\Gamma(U)-V$ 且无标记
- ∴增广路$P=(x_5, y_6, x_2, y_4, x_3, y_2)$

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

二分图的最大匹配(14)

- (3) $U=\{x_6\}$， $V=\Phi$

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

- $U=\{x_6, x_5\}$， $V=\{y_6\}$

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

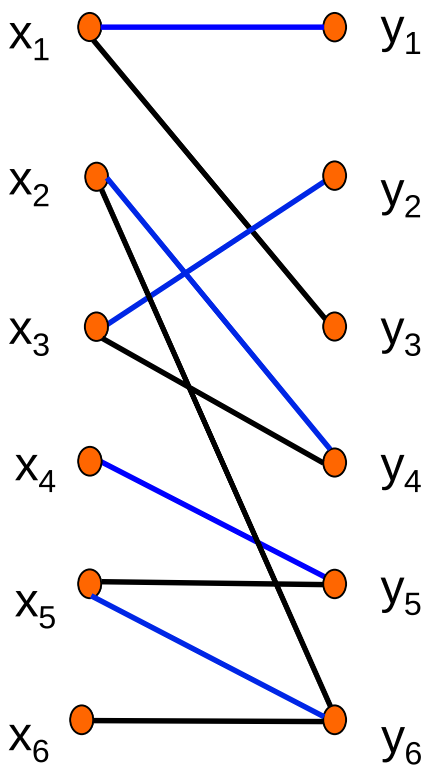
- $U=\{x_6, x_5, x_4\}$， $V=\{y_6, y_5\}$

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

- 给x_6标记2。结束。
- 因此其最大匹配是

  $M=\{(x_1, y_1), (x_4, y_5), (x_5, y_6), (x_2, y_4), (x_3, y_2)\}$

x_2?



二分图的最大匹配(15)

- 定理5.1.2
  - 最大匹配的匈牙利算法，计算复杂度为O(mn)，其中n是二分图G中X的结点数
- 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)

独立集算法

- **通过简单穷举办法得到最大匹配的方法复杂度过高**
- **转化为在0-1矩阵种求解1的最大独立集合**
  - 从某个1的独立集开始得到新的独立集
  - 最大独立集或者比原独立集所包含的1多一个

$$
\begin{array}{cc}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{bmatrix}
1^* & 0 & 1 & 1 \\
0 & 1^* & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

增加任何
一个“1”
不再独立

独立集算法

- **标记和扫描法**
  - 某行/列被标记，在同一次循环中不会被再次标记
  - 某行/列被扫描，在同一次循环中不会被再次扫描
  - 某行/列被扫描之前，必须被标记
- **从标记所有不带*的1的列开始，并用“#”标记**

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ \\
\end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\left[\begin{array}{cccc}
1^* & 0 & 1 & 1 \\
0 & 1^* & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0
\end{array}\right] \\
\begin{array}{cccc} & & \# & \# \end{array}
\end{array}
$$

独立集算法

- 扫描每个被标记的列，寻找不带*的1
- 在C列第一行中不带*的1
- 用C标记这个行指示不带*的1是在C列发现的
- 在C列下面做已验讫的标记√

$$
\begin{array}{ccc}
 & \begin{array}{cccc} A & B & C & D \end{array} & \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1^* & 0 & 1 & 1 \\ 0 & 1^* & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} C \\ \\ \\ \\ \end{array} \\
 & \begin{array}{cccc} & & \#\checkmark & \# \end{array} &
\end{array}
$$

独立集算法

- 扫描D列，在第一行发现不带*的1，但该行已经被标记，在D列下面做一个已验讫的标记
- 所有标记的列都已经扫描，开始行的扫描
- 只有第一行被标记，**寻找带*的1**
- 标记第一列，并在该行后面标记已验讫

$$
\begin{array}{ccc}
 & \begin{array}{cccc} A & B & C & D \end{array} & \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1^* & 0 & 1 & 1 \\ 0 & 1^* & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} C\checkmark \\ \\ \\ \\ \end{array} \\
 & \begin{array}{cccc} 1 & & \#\checkmark & \# \end{array} &
\end{array}
$$

独立集算法

- 所有标记的行都已经扫描，进行列扫描
- A列做了标记但是没有被扫描，扫描找到不带*的1
- 在第三行中用A列标记

$$
\begin{array}{cc}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \left[\begin{array}{cccc} 1^* & 0 & 1 & 1 \\ 0 & 1^* & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \begin{array}{c} C\checkmark \\ \\ A \\ \\ \end{array} \\
 & \begin{array}{cccc} 1\checkmark & & \#\checkmark & \# \end{array}
\end{array}
$$

独立集算法

- 关键转折步骤！
- 扫描标记的第3行，没有发现带*的1
- 节点3是一个非饱和点！用“！”标记，表明是可以改进初始1的独立集

$$
\begin{array}{cc}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \left[\begin{array}{cccc} 1^* & 0 & 1 & 1 \\ 0 & 1^* & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \begin{array}{c} C\checkmark \\ \\ A! \\ \\ \end{array} \\
 & \begin{array}{ccccc} 1\checkmark & & \#\checkmark & \# \end{array}
\end{array}
$$

独立集算法

- 第三行被标记为A，所以在A列3行的1画圆圈
- 由于该列被被标记为1，在1行A列带*的1外面画一个圆圈
- 第一行被标记为C，在1行C列画一个圆圈
- C列被#标记，停止（找到非饱和点！）

$$
\begin{array}{cc c}
 & \begin{array}{cccc} A & B & C & D \end{array} & \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1^* & 0 & 1 & 1 \\ 0 & 1^* & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} C\checkmark \\ \\ A! \\ \\ \end{array} \\
 & \begin{array}{cccc} 1\checkmark & & \#\checkmark & \# \end{array} &
\end{array}
$$

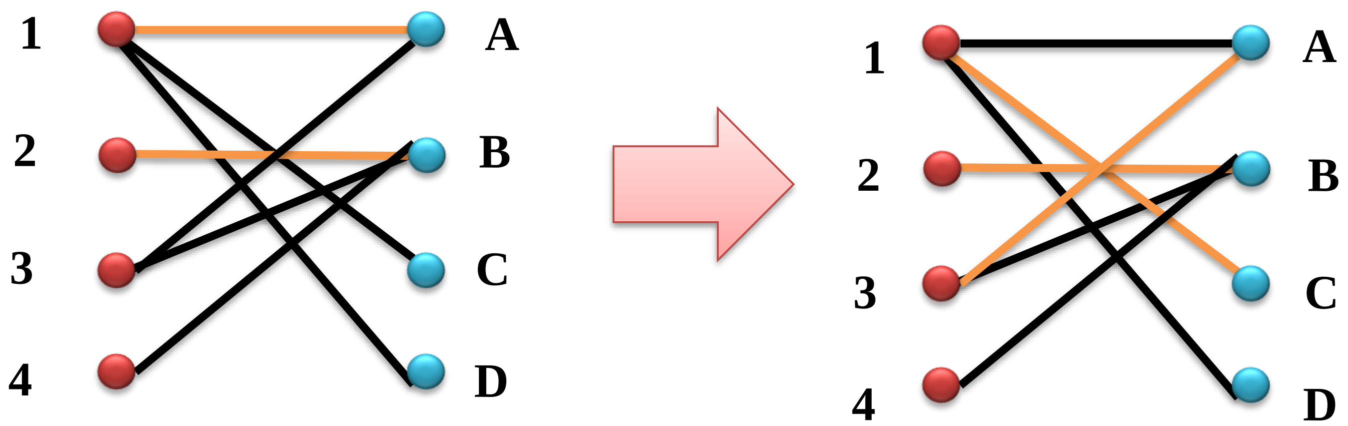
独立集算法

- 反转所有带圆圈的1上的*，找到一个更大的独立集
- 从2个元素增加到3个元素

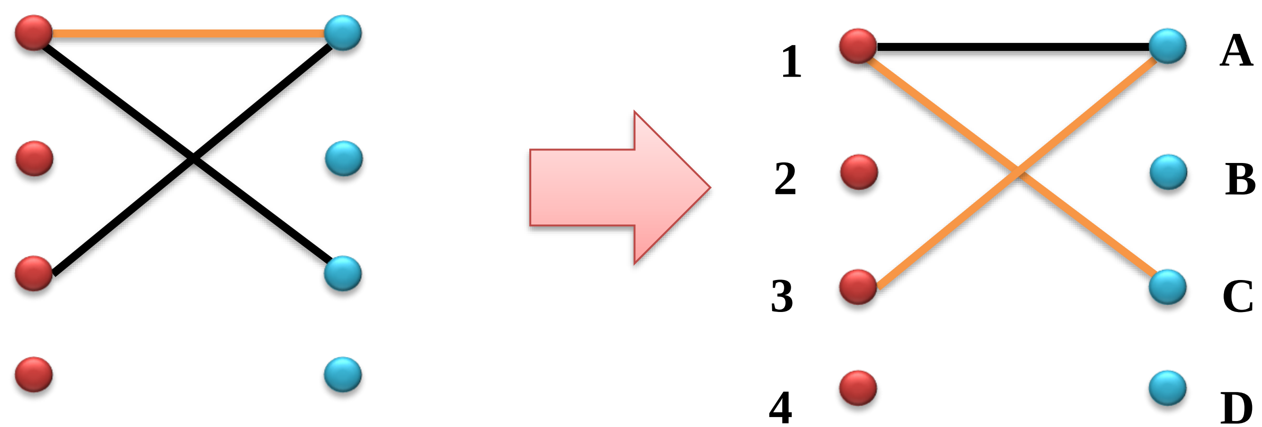
$$\begin{array}{ccc}
 & \begin{matrix} A & B & C & D \end{matrix} & \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 0 & 1^* & 1 \\ 0 & 1^* & 0 & 0 \\ 1^* & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & \begin{matrix} C\checkmark \\ \\ A! \\ \\ \end{matrix} \\
 & \begin{matrix} 1\checkmark & & \#\checkmark & \# \end{matrix} &
\end{array}$$

独立集算法

◈ 从图的角度理解

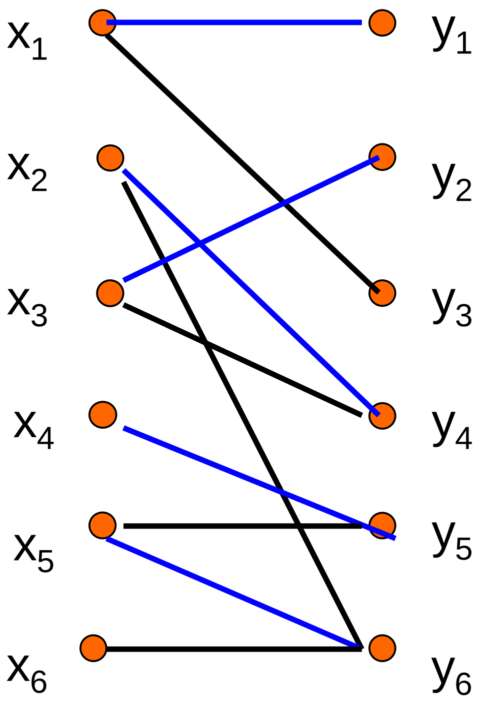


◈ 简单道路的观点



二分图的最大匹配(16)

- 最大匹配之后呢？



特例情况?

X是否能全都匹配?

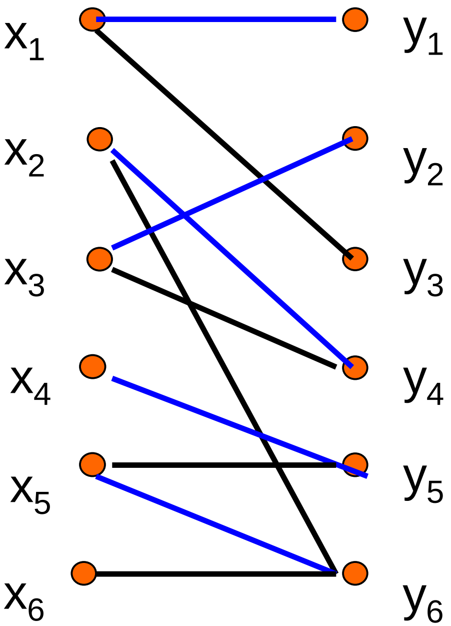
Y是否能全都匹配?

第五章 匹配与网络流

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

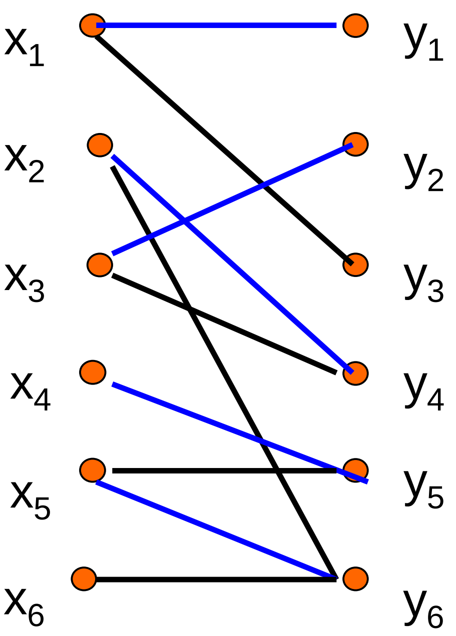
完全匹配

- **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
- **完美匹配**
  - 如果|M|=|X|=|Y|，则称M是完美匹配
- 完全匹配的存在性？
- **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为：
  - 对于X的任意子集A，恒有|Γ(A)|≥|A|



完全匹配(2)

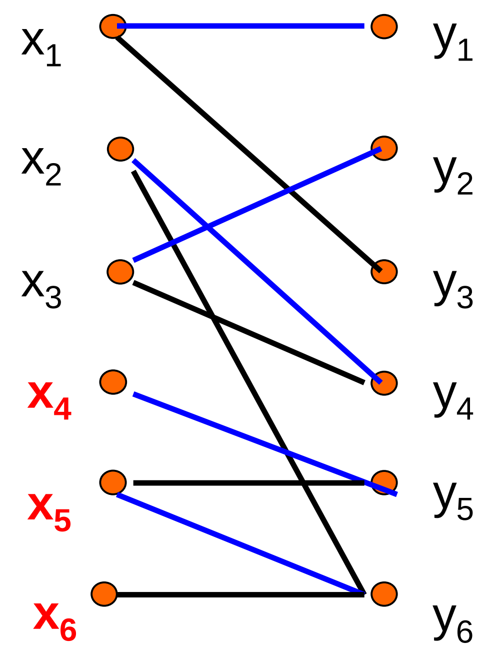
- 证明(存在完全匹配的充要条件：任意子集A有|Γ(A)|≥|A|)
  - 必要性(反证法)
    - 若存在子集A ⊆ X，使|A| >|Γ(A)|
    - 则A中的结点无法全部匹配
    - 因此X到Y不可能有完全匹配
  - 充分性 (反证法，构造A)
    - 要证：不存在完全匹配，则存在A满足|Γ(A)|<|A|)
    - 假定G的一个最大匹配M不是完全匹配
    - 一定存在结点x_0∈X是关于M的非饱和点(如x_6)
    - 如果Γ(x_0)=Φ，则令A={x_0}，于是|Γ(A)|< |A|，不满足条件



完全匹配(3)

- 证(续)(要证存在子集|A| >|Γ(A)|)
  - 若Γ(x_0) ≠ Φ，对某y_i ∈ Γ(x_0)
  - 若y_i关于M为非饱和点，则存在增广路(x_0，y_i)，与M是最大匹配矛盾
  - 因此y_i ∈ Γ(x_0)都是关于M的饱和点，但是x_0y_i为非匹配边
  - 这样可以寻找以x_0为端点的相对于M的一切交互道路
  - 记交互道路中结点y_i的集合为Y_1，结点x_i的集合为X_1(新增的x与y一一对应)
  - 根据匹配的性质，Y_1结点与X_1-x_0的结点之间存在一一对应，于是$|X_1|>|Y_1|$，即$|X_1|>|Γ(X_1)|$

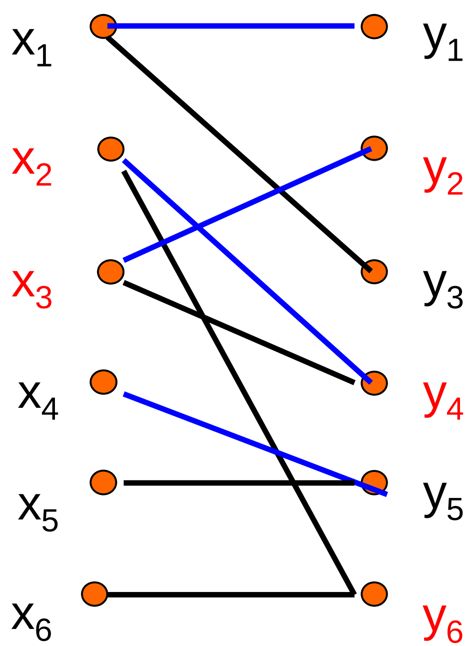
y_6饱和吗?



对非完全匹配，构造子集|A|，满足|A| >|Γ(A)|。证毕充分性

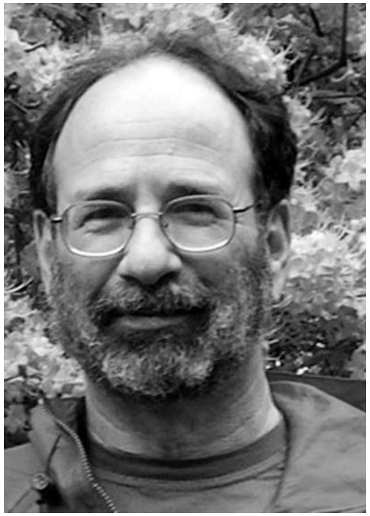
完全匹配(4)

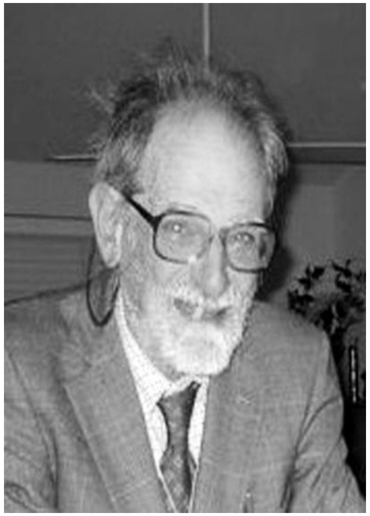
- **推论5.2.1**
  - 若二分图G=(X,Y,E)的每个结点$x_i \in X$，都有$d(x_i) \geq k$，每个结点$y_j \in Y$，都有$d(y_j) \leq k$，
    则X到Y存在完全匹配
- **证明**
  - 对任意子集$A \subseteq X$
  - 设A共与m条边相关联，由$d(x_i) \geq k$
    且边不重复，于是$m \geq k|A|$
  - 这m条边又与Y中的$|\Gamma(A)|$个结点相关联，由$d(y_j) \leq k$，又有$m \leq k|\Gamma(A)|$
  - 因此$|\Gamma(A)| \geq |A|$，由定理5.2.1即得



稳定匹配理论和市场设计实践

- 2012年诺贝尔经济学奖
  - 美国经济学家阿尔文·罗思和劳埃德·沙普利
  - 如何尽可能适当地匹配不同市场主体。
  - 这样的匹配如何尽可能有效地完成？什么样的方法对什么样的人群有益？





劳埃德·沙普利:
著名的飞虎队的队员

完全匹配(5)

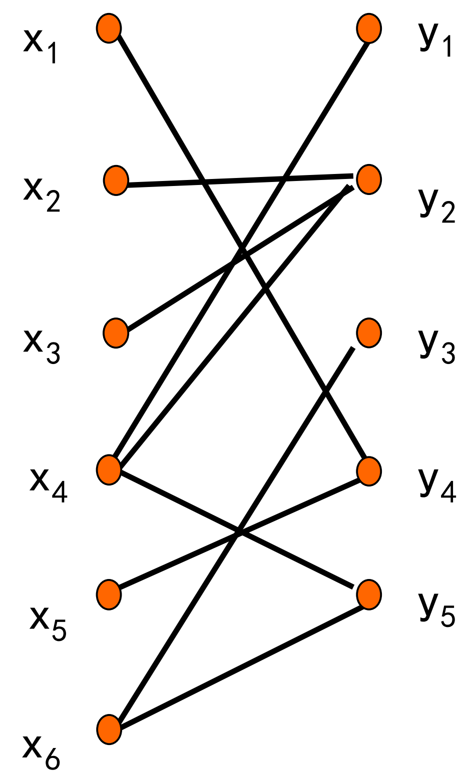
- 例5.2.1
  - 在一个舞会上男女各占一半，假定每位男士都认识k位女士，每位女士都认识k位男士，那么一定可以安排得当，使每位都有认识的人作为舞伴
- 证明（图论建模）
  - 图论建模：用结点x_i表示每位男士，结点y_j表示每位女士，互相认识者用边连之，则存在完美匹配
  - 于是得到二分图G=(X,Y,E)，图中每个x_i 结点有 $d(x_i)=k$，每个y_j 结点有$d(y_j)=k$
  - 满足$d(x_i)\geq k$，$d(y_j)\leq k$
  - 由推论5.2.1，X到Y有完全匹配M(即完美匹配)
  - M就是一种安排方案

二部图的矩阵表示

- **二分图邻接矩阵**
  - 简化为|X|×|Y|二值矩阵
- **最大匹配是多少？**
  - A中不同行同列的非零元的最大数目
  - 如果矩阵A是p×q矩阵，则最大匹配数≤min(p,q)

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

二分图邻接矩阵



最大匹配是多少？考虑x_2和x_3、x_1和x_5

计算机可否有快速算法？

覆盖

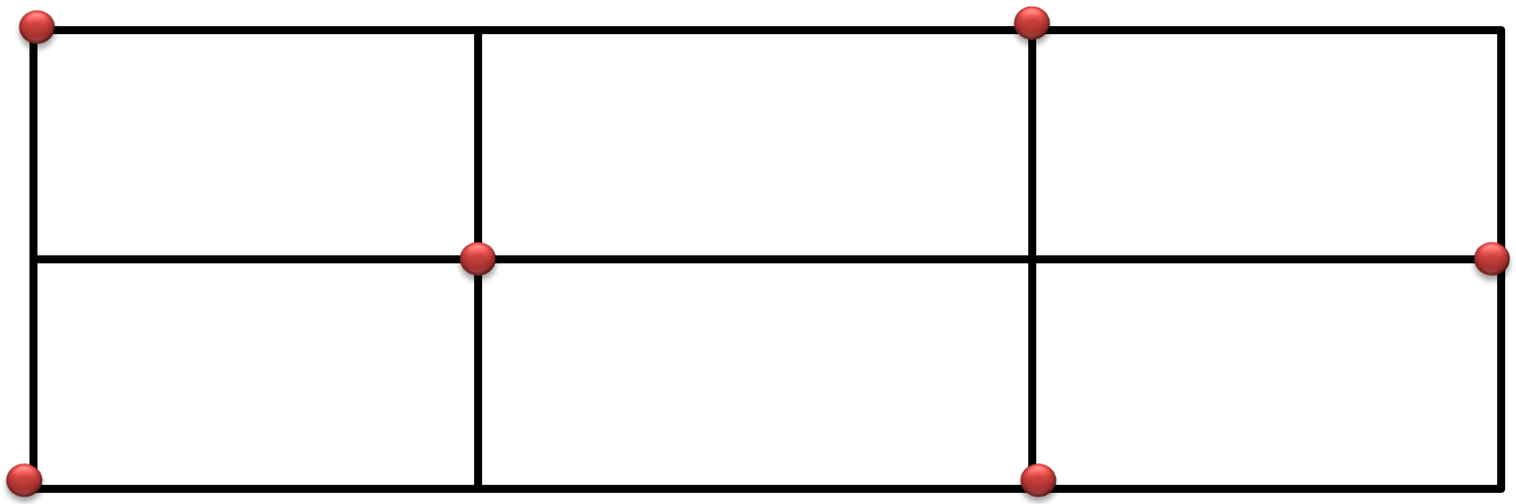
- **覆盖**
  - 适当地选取A的某些行和列，使这些行和列能盖住A的全部非零元，称为A的覆盖
  - 覆盖数为所选取的行和列的个数
  - 矩阵A中如果盖住第4、6行，第2、4列，就可以覆盖其全部非零元，覆盖数为4

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

- **最小覆盖**
  - 如果选取最少的行与列就能覆盖A的全部非零元，则称这样的覆盖为最小覆盖
  - 在矩阵A的各种覆盖中，一定存在最小覆盖，其覆盖数为s，显然s<=min(p,q)
  - 二分图的最大匹配数r，与其邻接矩阵的最小覆盖数s相等

覆盖

- 某公司希望在市区的若干十字路口设置广告摊位，使得市区中所有人距离最近摊位的距离不超过1个街区，该公司希望设置尽可能少摊位达到这个要求

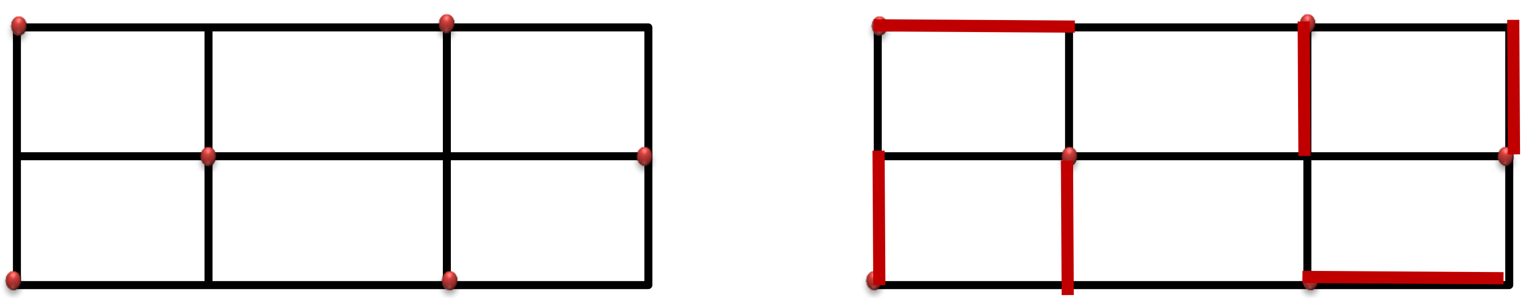


最小覆盖

- **设一个图有匹配M和一个覆盖C，那么 $\|M\| \leq \|C\|$，如果 $\|M\| = \|C\|$，那么M是一个最大匹配，C是一个最小覆盖**
- **证明：**
  - 根据覆盖的定义，图中的每条边，特别是M中的每条边与C中的某个顶点关联。
  - 如果e在M中，设v(e)是C中关联e的顶点。
  - 根据匹配的定义，匹配中的两条边e_1和 e_2 不能共享顶点
  - 则C中的顶点至少和M中的边一样多，因此 $\|M\| \leq \|C\|$

最小覆盖

- 证明续
  - 假设 $\|M\| = \|C\|$，如果M不是一个最大的匹配，那么存在匹配M’使得 $\|M'\| > \|M\| = \|C\|$ 矛盾！
  - 同理，如果C不是一个最小的覆盖，则存在一个顶点少于M的覆盖，$\|M\| = \|C\| > \|C'\|$ 矛盾！



第五章 匹配与网络流

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

完全匹配存在性，

唯一性？然后呢？

H道路，旅行商？

最佳匹配及其算法

- **从旅行商问题想到了什么?**
  - 前两节讨论了最简单的匹配（边权为1 ），边权不等于1的一般情况如何呢?

玩命简化

- **最佳匹配**
  - 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匹配就叫最佳匹配
- **例5.3.1**
  - 5项工作由5个人完成
  - i从事工作j的利润或成本矩阵
  - 某人不能做某事?

权值为0或无穷大

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

匈牙利算法

- **工时分配**

	工人1	工人2	工人3	工人4
工作A	3	6	3	5
工作B	7	3	5	8
工作C	5	2	8	6
工作D	8	3	6	4

匈牙利算法

- 设计独立集的元素之和尽量小

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 3^* & 6 & 3 & 5 \\ 7 & 3^* & 5 & 8 \\ 5 & 2 & 8^* & 6 \\ 8 & 3 & 6 & 4^* \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 3 & 6 & 3 & 5^* \\ 7 & 3^* & 5 & 8 \\ 5^* & 2 & 8 & 6 \\ 8 & 3 & 6^* & 4 \end{bmatrix} \end{array}$$

$$\sum x^* = 18 \qquad\qquad \sum x^* = 19$$

匈牙利算法

- 设计独立集的元素之和尽量小
- 第一行每个元素减去该行的最小值
- 不会改变相对大小关系！

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0^* & 3 & 0 & 2 \\ 7 & 3^* & 5 & 8 \\ 5 & 2 & 8^* & 6 \\ 8 & 3 & 6 & 4^* \end{bmatrix} \end{array} \qquad \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0 & 3 & 0 & 2^* \\ 7 & 3^* & 5 & 8 \\ 5^* & 2 & 8 & 6 \\ 8 & 3 & 6^* & 4 \end{bmatrix} \end{array}$$

$$\sum x^* = 15 \qquad\qquad \sum x^* = 16$$

匈牙利算法

- 对于所有的行和列都先后执行同样的操作
- 减去该行和该列的最小值

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 2 \\ 4 & 0 & 2 & 5 \\ 3 & 0 & 6 & 4 \\ 5 & 0 & 3 & 1 \end{bmatrix} \end{array}
\Rightarrow
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 1 \\ 4 & 0 & 2 & 4 \\ 3 & 0 & 6 & 3 \\ 5 & 0 & 3 & 0 \end{bmatrix} \end{array}
$$

应用独立集算法，寻找0的独立集

匈牙利算法

- 0的最大独立集含有3个元素，存在最小覆盖

$$
\begin{array}{c|cccc|c}
 & 1 & 2 & 3 & 4 & \\
A & 0^* & 3 & 0 & 1 & C\checkmark \\
B & 4 & 0^* & 2 & 4 & \\
C & 3 & 0 & 6 & 3 & \\
D & 5 & 0 & 3 & 0^* & \\
 & 1\checkmark & & \#\checkmark & &
\end{array}
$$

匈牙利算法

- 不在覆盖中的最小元素是2
- 不在覆盖中的元素减2
- 既在行覆盖又在列覆盖中元素加2
- 只在一个覆盖中的元素保持不变

$$
\begin{array}{c c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0^* & 3 & 0 & 1 \\ 4 & 0^* & 2 & 4 \\ 3 & 0 & 6 & 3 \\ 5 & 0 & 3 & 0^* \end{bmatrix} & \begin{array}{c} C\checkmark \\ \\ \\ \\ \end{array} \\
 & \begin{array}{cccc} 1\checkmark & & \#\checkmark & \end{array} &
\end{array}
$$

匈牙利算法

◆ 找出含有4个0的独立集，并在原矩阵找出对应元素

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} &
\begin{bmatrix} 0^* & 5 & 0 & 3 \\ 2 & 0 & 0^* & 4 \\ 1 & 0^* & 4 & 3 \\ 3 & 0 & 1 & 0^* \end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} &
\begin{bmatrix} 3^* & 6 & 3 & 5 \\ 7 & 3 & 5^* & 8 \\ 5 & 2^* & 8 & 6 \\ 8 & 3 & 6 & 4^* \end{bmatrix}
\end{array}
$$

$$\sum x^* = 14$$

匈牙利算法

- **步骤1 减小矩阵**
  - 将每行中的各元素减去该行的最小元
  - 将每列中的各元素减去该列的最小元
- **步骤2 确定一个0的最大独立集**
  - 在矩阵中找出一个0的最大独立集
- **步骤3 如果$|S| < n$，扩大独立集**

匈牙利算法

- **循环** $|S| < n$
  - 找出矩阵的一个0的最小覆盖
  - 设 k 是不在这个覆盖的行（列）上的最小矩阵元素
  - 将不在覆盖的行（列）上的每个元素减去 k
  - 将既在这个覆盖的行中又在这个覆盖列中的每个元素加上 k
  - 用一个新的0的最大独立集替代 $|S|$
- **步骤四 输出集合** S

最佳匹配及KM算法(2)

- 若C_{ij}表示i从事工作j的利润
  - 如果每个人只从事一项工作
  - 那么最大利润就应该是
  - C_{ij}不在相同的行与列
- 若C_{ij}表示i从事工作j成本
  - 那么最小的成本应该是
  - C_{ij}不在相同的行与列
- 这种最佳匹配就是二分图的最大权或是最小权匹配
- 在讨论最佳匹配时，二分图G=(X,Y,E)满足条件|X|=|Y|

$$\max \sum C_{ij}$$

$$\min \sum C_{ij}$$

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

最佳匹配及KM算法(3)

- 最大权匹配算法(已知利润矩阵C)
  - 1. 在C的每行中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$B=(b_{ij})_{n\times n}$，其中$b_{ij}= l(x_i)+ l(y_j)- c_{ij}$
  - 2. 在B中对0元素进行最小覆盖，覆盖数为r
    - 2.1 若r=n，转4
    - 2.2 在未覆盖的元素中选最小非零元δ
      计算B矩阵
      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$b_{ij}\leftarrow b_{ij}-\delta$

最佳匹配及KM算法(4)

- 最大匹配算法(续)
  - 3. 修改界值

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

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

    删除覆盖标记 转2
  - 4. $\sum(l(x_i)+l(y_j))=29$ 即为最大权，结束

最佳匹配及KM算法(5)

- **例5. 3. 2**
  - 已知利润矩阵，求二分图的最佳匹配(最大利润)
- **解**
  - 计算界值(表的两旁标出)
  - 计算矩阵B
  - 求最小覆盖
    - 如1、5两列
  - 在未覆盖的元素中选最小非零元，即 δ =2

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

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

最佳匹配及KM算法(6)

- **解(续)**
  - 覆盖数r=2<n， δ =2
  - 产生新B:未覆盖的元素均减δ（最小值）
  - 修改界值
    - 若x_i行没覆盖，则$l(x_i) \leftarrow l(x_i)-\delta$
    - 若y_j列已覆盖，则$l(y_j) \leftarrow l(y_j)+\delta$
  - 重新求最小覆盖
    - 如第1、5两列和第3行
  - 在未覆盖的元素中选最小非零元，即 δ =1

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

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

最佳匹配及KM算法(7)

- **解(续)**
  - 覆盖数r<n ， δ =1
  - B中没覆盖的元素均减1
  - 双重覆盖元加1
  - 修改界值
    - 没覆盖的x_i行：$l(x_i) \leftarrow l(x_i)-\delta$
    - 已覆盖的y_j列：$l(y_j) \leftarrow l(y_j)+\delta$
  - 重新求最小覆盖
    - 如第1，2，3，5列
  - 在未覆盖的元素中选最小非零元，即 δ =1

$$
\begin{array}{c}
\begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 6 \end{array}
\left[\begin{array}{ccccc}
6 & 3 & 1 & 3 & 0 \\
2 & 2 & 1 & 3 & 0 \\
0 & 0 & 2 & 1 & 5 \\
0 & 1 & 2 & 2 & 1 \\
0 & 2 & 1 & 2 & 1
\end{array}\right] \\
\begin{array}{ccccc} 2 & 0 & 0 & 0 & 2 \end{array}
\end{array}
$$

$$
\begin{array}{c}
\begin{array}{c} 6 \\ 5 \\ 5 \\ 3 \\ 5 \end{array}
\left[\begin{array}{ccccc}
6 & 2 & 0 & 2 & 0 \\
2 & 1 & 0 & 2 & 0 \\
1 & 0 & 2 & 1 & 6 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{array}\right] \\
\begin{array}{ccccc} 3 & 0 & 0 & 0 & 3 \end{array}
\end{array}
$$

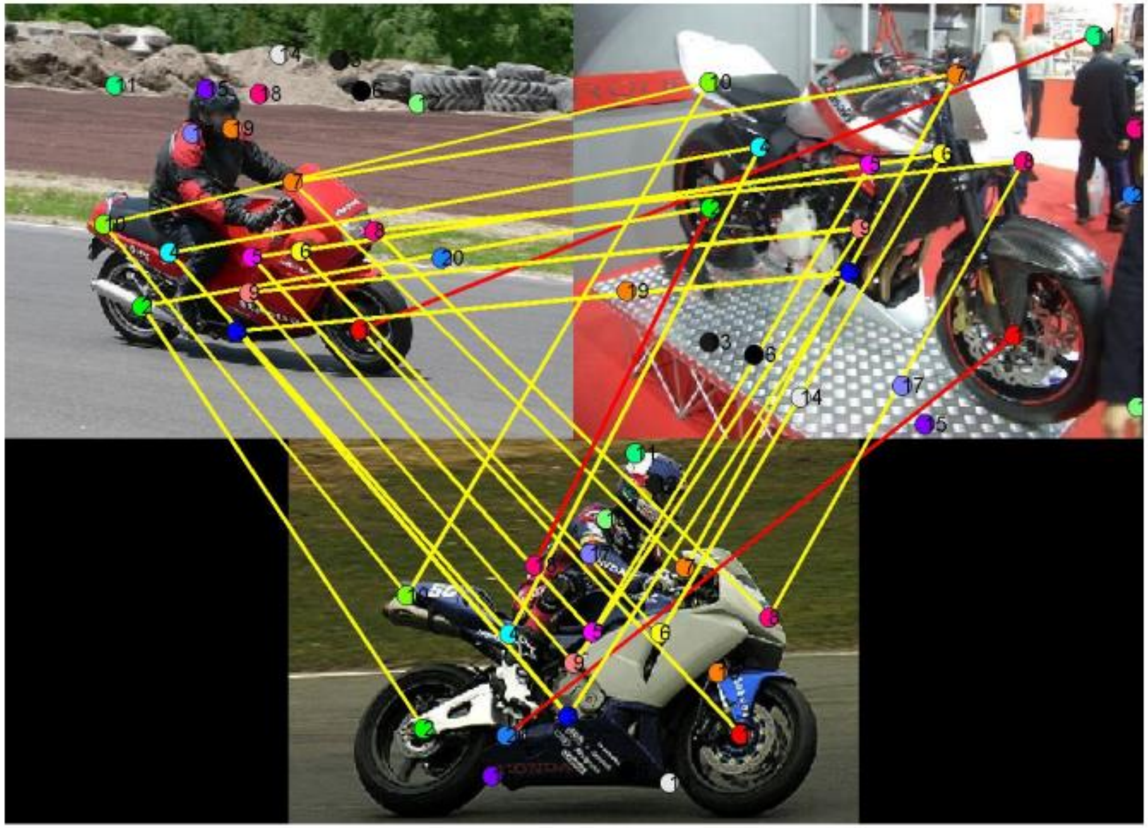
最佳匹配及KM算法(8)

- 覆盖数r<n， δ =1
- B中没覆盖的元素均减1
- 修改界值
  - 没覆盖的x_i行：$l(x_i) \leftarrow l(x_i)-\delta$
  - 已覆盖的y_j列：$l(y_j) \leftarrow l(y_j)+\delta$
- 重新求最小覆盖
  - 如第3、4、5行和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}}
- 最大权 $\sum(l(x_i)+l(y_j))=29$
- **定理5.3.1：上述算法（最大权匹配算法）的结果是矩阵C的最大权匹配**

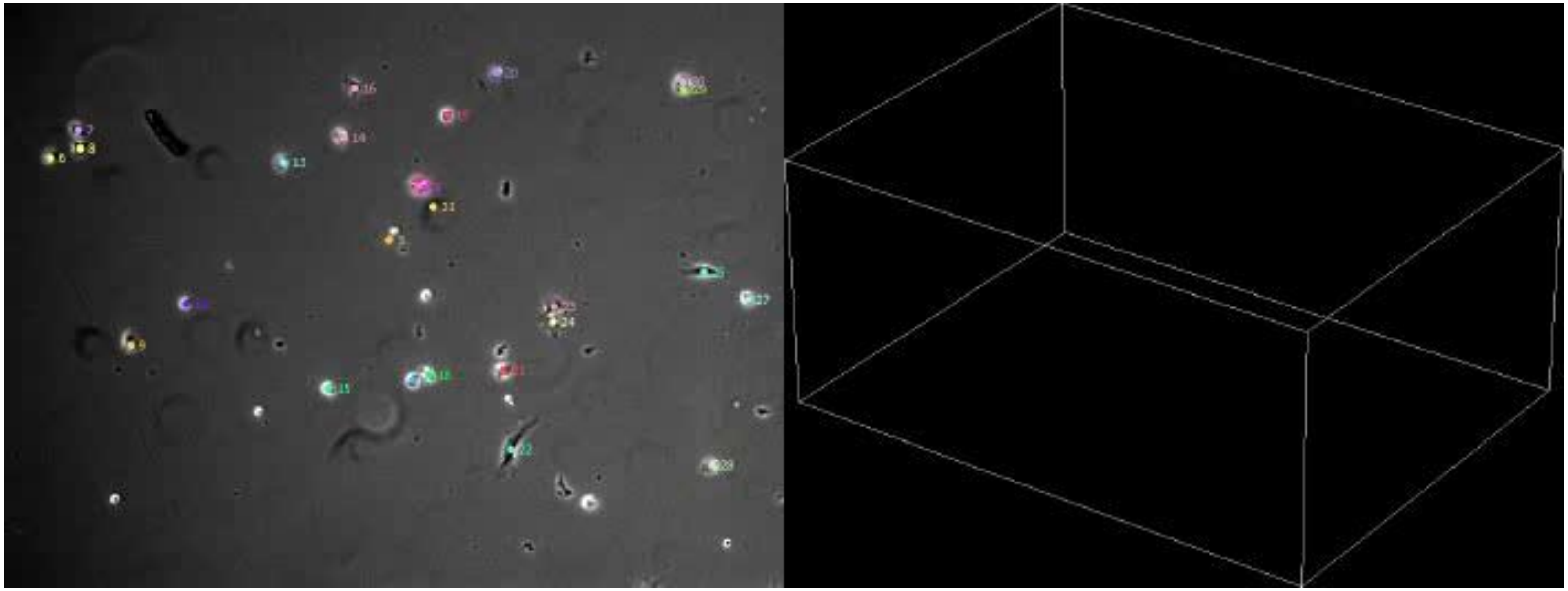
$$
\begin{array}{c}
\begin{array}{c} 6 \\ 5 \\ 5 \\ 3 \\ 5 \end{array}
\begin{bmatrix}
6 & 2 & 0 & 2 & 0 \\
2 & 1 & 0 & 2 & 0 \\
1 & 0 & 2 & 1 & 6 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix} \\
\begin{array}{ccccc} 3 & 0 & 0 & 0 & 3 \end{array}
\end{array}
$$

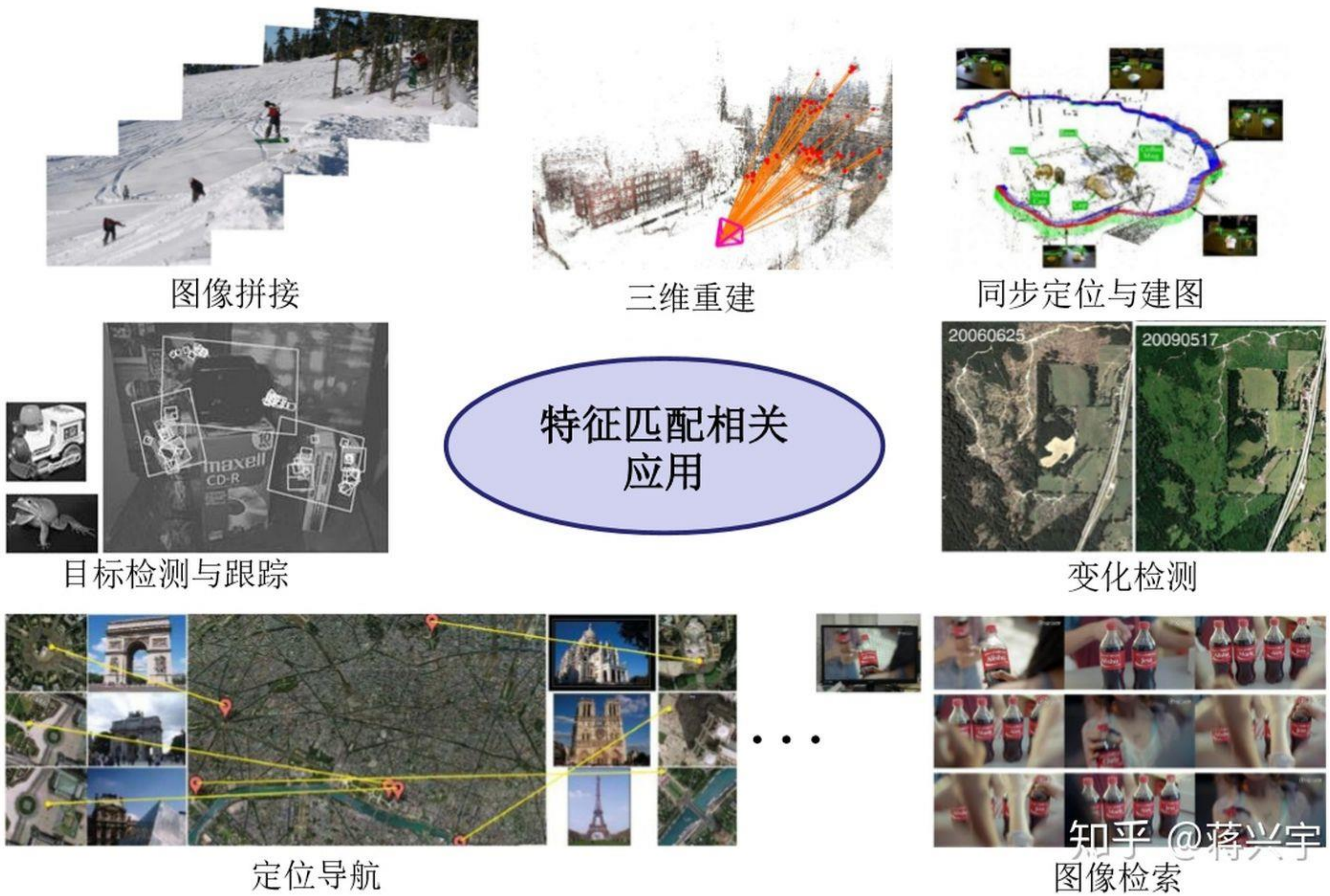
$$
\begin{array}{c}
\begin{array}{c} 5 \\ 4 \\ 4 \\ 2 \\ 4 \end{array}
\begin{bmatrix}
6 & 2 & 0 & 1 & 0 \\
2 & 1 & 0 & 1 & 0 \\
1 & 0 & 2 & 0 & 6 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1
\end{bmatrix} \\
\begin{array}{ccccc} 4 & 1 & 1 & 0 & 4 \end{array}
\end{array}
$$

图匹配问题



基于图匹配的多目标跟踪

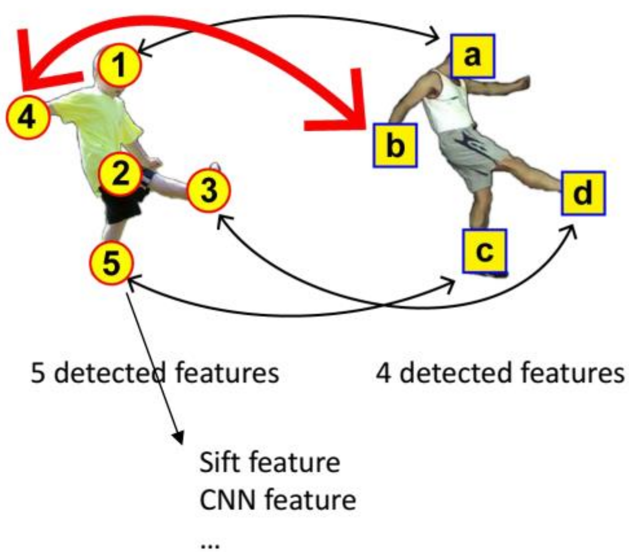


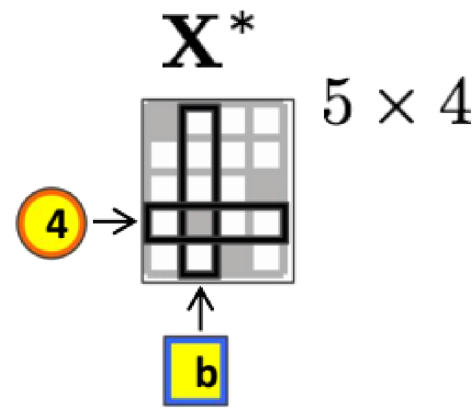

图像拼接
三维重建
同步定位与建图
目标检测与跟踪
maxell
CD-R
特征匹配相关应用
20060625
20090517
变化检测
定位导航
图像检索

图匹配建模

- 利用分配矩阵对图匹配进行数学建模
- 二值0-1矩阵，每行（列）不超过1个1，每列（行）只有1个1

节点之间的一一映射



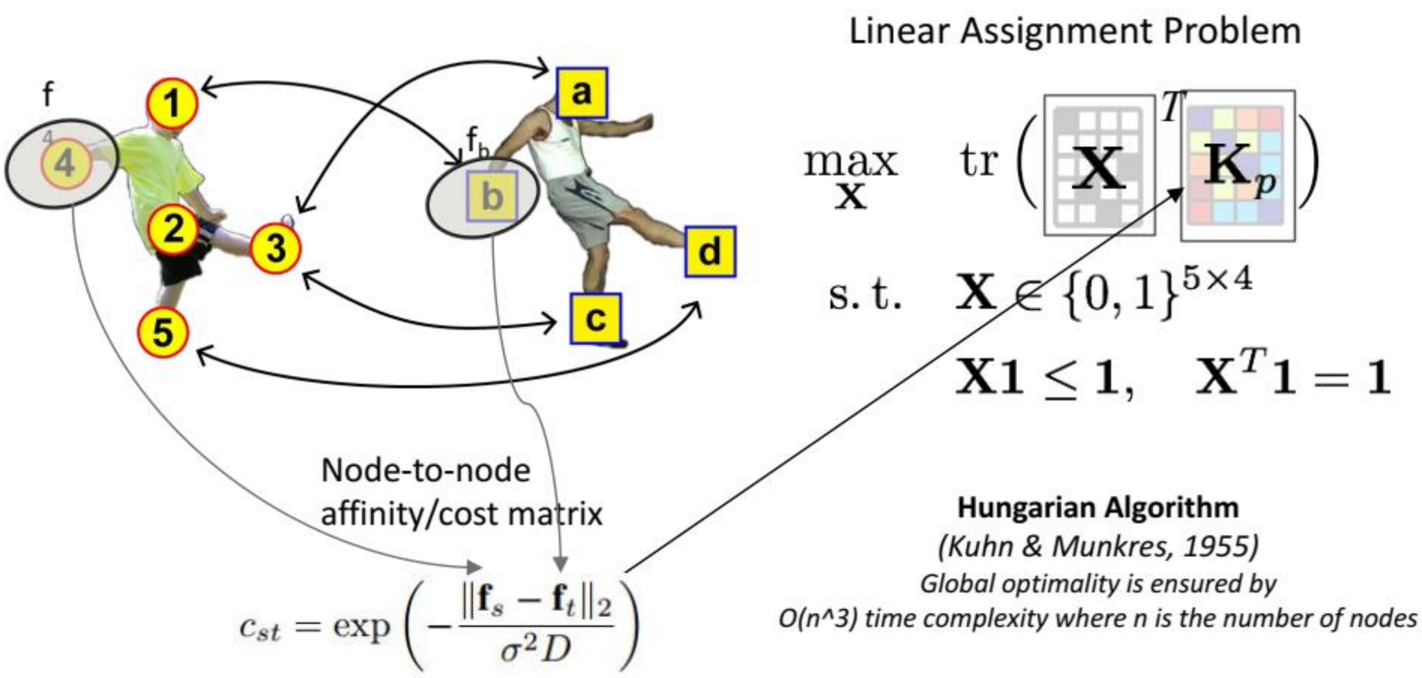


$$\mathbf{X} \in \{0,1\}^{5\times 4}$$

$$\mathbf{X1} \leq \mathbf{1}, \quad \mathbf{X}^T\mathbf{1} = \mathbf{1}$$

图匹配建模

◆ 结合点点相似度矩阵和分配矩阵



$$\max_{\mathbf{X}} \quad \mathrm{tr}\left(\mathbf{X}^T \mathbf{K}_p\right)$$

$$\text{s.t.} \quad \mathbf{X} \in \{0,1\}^{5\times 4}$$

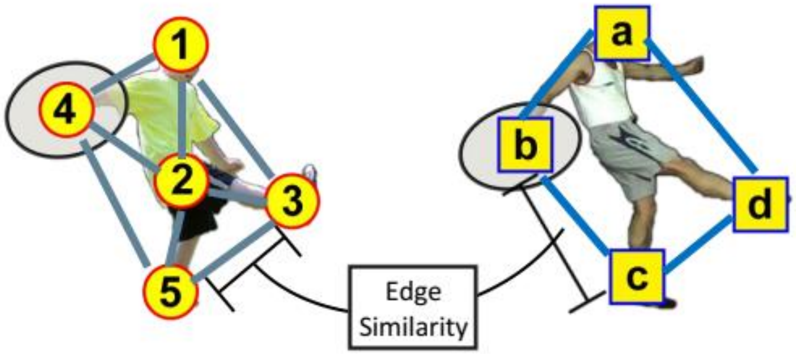
$$\mathbf{X}\mathbf{1} \leq \mathbf{1}, \quad \mathbf{X}^T\mathbf{1} = \mathbf{1}$$

$$c_{st} = \exp\left(-\frac{\|\mathbf{f}_s - \mathbf{f}_t\|_2}{\sigma^2 D}\right)$$

- S. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching,” IEEE Transaction on PAMI, 1996

图匹配建模

- 节点相似度可以拓展为边的匹配
- 子集内部节点有关联



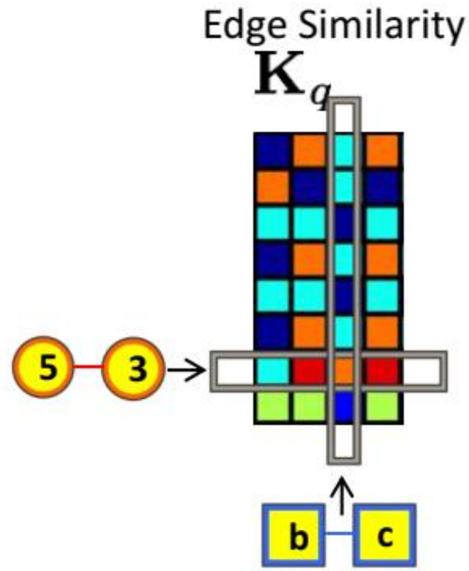
1st-order Feature (eg. Local Texture)

Feature Matching (linear)

+

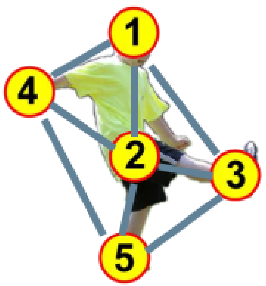
2nd-order Feature (eg, Edge Length)

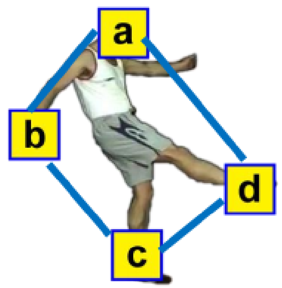
Graph Matching (quadratic)



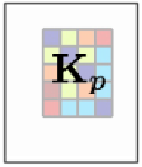
图匹配

- 综合考虑节点相似度和边的相似度

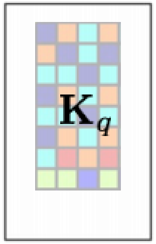




Node Similarity



Edge Similarity



Affinity maximization

Steven Gold

A. Rangarajan

Node Compatibility

Edge Compatibility

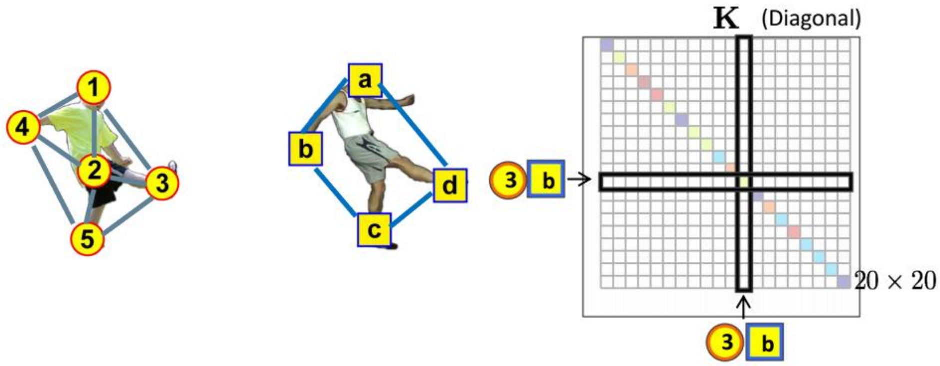
$$\max_{\mathbf{X}} \sum_{i_1 i_2} x_{i_1 i_2}\kappa^p_{i_1 i_2} + \sum_{\substack{(i_1,j_1)\in E_1\\(i_2,j_2)\in E_2}} x_{i_1 i_2}x_{j_1 j_2}\kappa^q_{c(i_1,j_1)c(i_2,j_2)}$$

Node-Edge Index

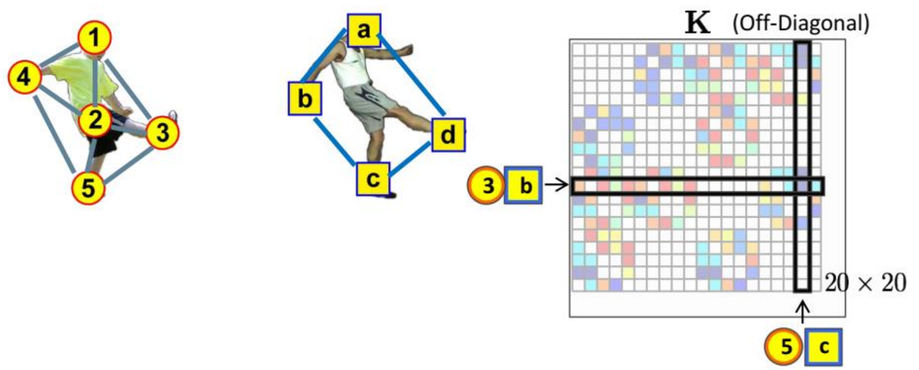
$$\mathrm{s.t.}\quad \mathbf{X}\in\{0,1\}^{5\times 4},\quad \mathbf{X}\mathbf{1}\leq\mathbf{1},\quad \mathbf{X}^T\mathbf{1}=\mathbf{1}$$

广义相似度矩阵

- 顶点间相似度（假想自环）

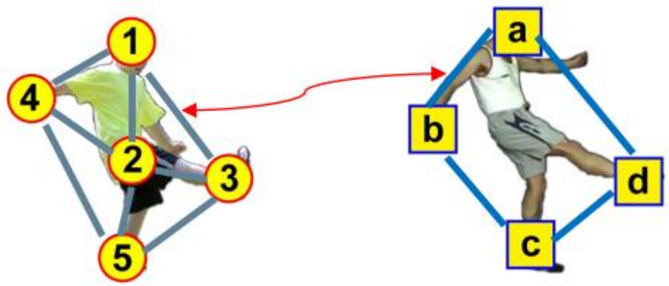


- 边相似度



基于广义相似度矩阵的图匹配建模

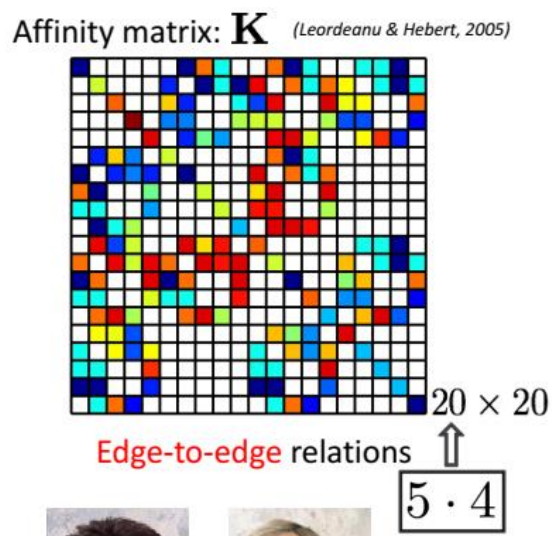
◆ 图匹配建模



$$\max_{\mathbf{X}} \quad \mathrm{vec}(\mathbf{X})^T \, \mathbf{K} \, \mathrm{vec}(\mathbf{X})$$

$$\text{s.t.} \quad \mathbf{X} \in \{0,1\}^{5\times 4}$$

$$\mathbf{X}\mathbf{1} \leq \mathbf{1}, \quad \mathbf{X}^T\mathbf{1} = \mathbf{1}$$



M. Leordeanu

M. Hebert

- M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in ICCV, 2005

图匹配的谱近似算法

◆ 谱近似算法

$$\max_{\mathbf{X}} \ \mathrm{vec}(\mathbf{X})^T \mathbf{K} \, \mathrm{vec}(\mathbf{X})$$

$$\text{s.t.} \ \cancel{\mathbf{X} \in \{0,1\}^{5\times 4}}$$

$$\cancel{\mathbf{X}\mathbf{1} \le \mathbf{1}, \quad \mathbf{X}^T\mathbf{1} = \mathbf{1}}$$

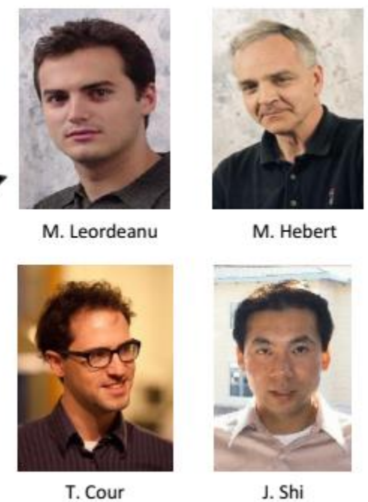
$$\|\mathrm{vec}(\mathbf{X})\|_2^2 = 1$$

Faster

Not Tight

Not Discrete

Spectral Method is Faster



- M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in ICCV, 2005
- P. S. T. Cour and J. Shi, “Balanced graph matching,” in NIPS, 2006

基于谱分析的优化

- **有约束优化问题求解，令** $\mu_1 = vec(X)$

$$\mu_1 = \arg\max_{\mu_1} \mu_1^\top K \mu_1 \quad s.t., \quad \mu_1^T \mu_1 = 1$$

- **利用拉格朗日法求解**
  - 特征值问题 $K\mu_1 = \lambda_1 \mu_1$
  - μ_1 的解是 K 最大的特征对应的特征向量
- **特征值分解**（Eigendecomposition），又称**谱分解**（Spectral decomposition）

总结

- **二分图的匹配**
  - 独立集算法
  - 最大匹配等于最小覆盖
- **完全匹配**
  - 霍尔定理
- **最佳匹配**
  - 匈牙利算法
  - KM算法
  - 谱分析算法

Questions?