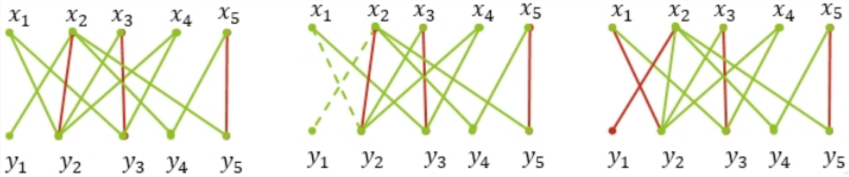
本次课程提纲：偶图的匹配

- 匈牙利算法
- 最优匹配算法
- 应用

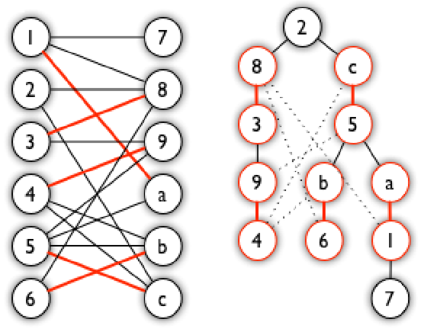
匈牙利算法

- 输入：偶图 $G = (X, Y)$
- 输出：饱和 X 的匹配
- 第 0 步：若 $|X| > |Y|$，停止；否则，任取一个匹配 M
- 第 1 步：若 M 饱和 X，输出 M，停止；否则，取 M 中一个 M 非饱和点 x，记 $S = \{x\}$，$T = \emptyset$
- 第 2 步：若 $N(S) \subseteq T$，不存在饱和 X 的匹配，停止；否则，取 $y \in N(S) - T$
- 第 3 步：若 y 是 M 饱和的，设 $yz \in M$，$S \leftarrow S \cup \{z\}$，$T \leftarrow T \cup \{y\}$，转第 2 步；否则，找一条 M 可扩路 $P(x, y)$，$M \leftarrow M \Delta E(P)$，转第 1 步



构造可扩路

- 基于广度或深度优先搜索



- 匈牙利算法复杂性：$O(|V| * |E|)$，$O(n^3)$

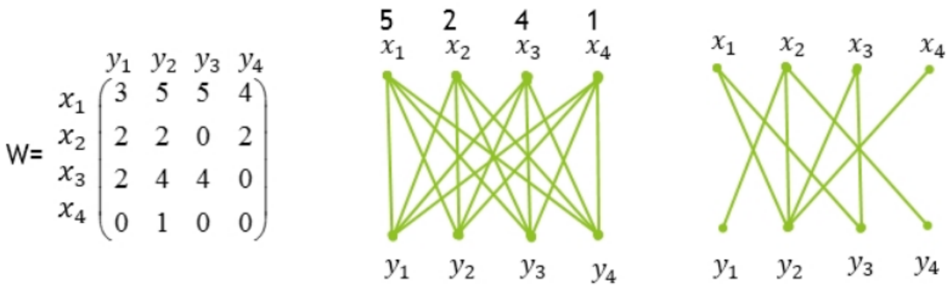
最优匹配

- 问题：设 $G=(X,Y)$ 是边赋权完全偶图，$X=\{x_1,\cdots,x_n\}$，$Y=\{y_1,\cdots,y_n\}$，$w_{ij}=w(x_iy_j)$，求一个具有最大权值的完美匹配
- 可行顶点标号
  - 若对任意的 $x\in X, y\in Y$，有 $l(x)+l(y)\geq w(xy)$，称 l 是 G 的可行顶点标号
- 对于任意 G，均存在可行顶点标号

$$l(x)=\max_{y\in Y} w(xy)$$

$$l(y)=0$$

- 设 l 是 G 的可行顶点标号，令 $E_l = \{xy \in E(G) | l(x) + l(y) = w(xy)\}$，称 G 的生成子图 $G_l = G[E_l]$ 为 G 对应于 l 的相等子图



图G的各边的权　　图G的平凡标号　　对应的相等子图

最优匹配

定理

设 l 是赋权完全偶图 G 的可行顶点标号，若相等子图 G_l 有完美匹配 M^*，则 M^* 是 G 的最优匹配

证明

- 设 M^* 是 G_l 的完美匹配：$w(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in V(G)} l(v)$
- 设 M 是 G 的任一完美匹配：$w(M) = \sum_{e \in M} w(e) \le \sum_{v \in V(G)} l(v)$
- 故 $w(M^*) \ge w(M)$，即 M^* 是 G 的最优匹配

如果找到一种可行顶点标号，对应的相等子图有完美匹配，则求出了 G 的最优匹配

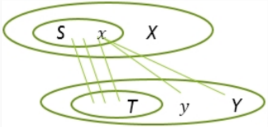
最优匹配 Kuhn-Munkres 算法

- 第 0 步：基于任意可行标号 l，求出相等子图 G_l，任选 G_l 一个匹配 M
- 第 1 步：若 M 是完美匹配，算法终止；否则，令 x 是一个 M 非饱和顶点，置 $S=\{x\}$，$T=\emptyset$
- 第 2 步：若 $T\subset N_{G_l}(S)$，转第 3 步；否则有 $N_{G_l}(S)=T$，修改标号 l 如下：

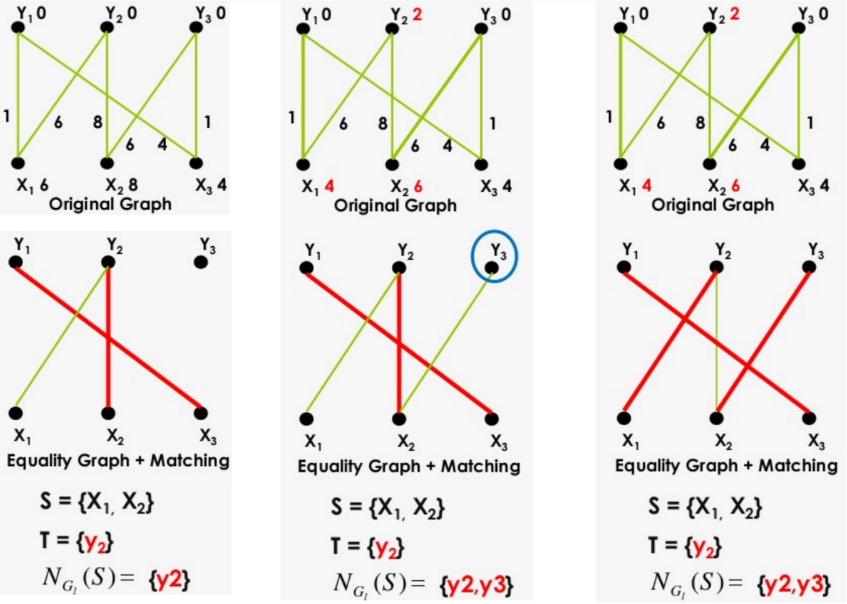
$$\alpha_l=\min\{l(x)+l(y)-w(xy)|x\in S,y\in Y-T\}$$

$$l(v)\leftarrow\begin{cases}l(v)-\alpha & v\in S\\ l(v)+\alpha & v\in T\end{cases}$$

- 第 3 步：在 $N_{G_l(S)}-T$ 中选择一个顶点 y，若 y 是 M 饱和的，记 $yz\in M$，更新 $S\leftarrow S\cup\{z\}$，$T\leftarrow T\cup\{y\}$，转第 2 步；否则，寻找 G_l 中的 M 可扩 (u,y) 路，用其更新 M，转第 1 步



最优匹配 Kuhn-Munkres 算法



最优匹配 Kuhn-Munkres 算法

- 标号的调整不影响已有匹配
- 每次改变标号都会扩大 T，从而扩大匹配
- 完全图存在完美匹配
- 算法复杂度 $O(n^3)$

任务分配匈牙利算法

问题：N 个人分配 N 项任务，每人分配一项，将一项任务分给一个人需支付报酬，如何分配任务，支付的报酬总数最少

- Step 1: Subtract row minima
- Step 2: Subtract column minima
- Step 3: Cover all zeros with a minimum number of lines
  - If n lines are required, stop: an optimal assignment exists among 0s
- Step 4: Create additional zeros
  - Find the smallest element (call it k) not covered by a line in Step 3
  - Subtract k from uncovered elements, add k to elements covered twice.
  - Go to Step 3

任务分配匈牙利算法

	J1	J2	J3	J4
W1	82	83	69	92
W2	77	37	49	92
W3	11	69	5	86
W4	8	9	98	23

Step 1: Subtract row minima

	J1	J2	J3	J4	
W1	13	14	0	23	(-69)
W2	40	0	12	55	(-37)
W3	6	64	0	81	(-5)
W4	0	1	90	15	(-8)

Step 2: Subtract column minima

	J1	J2	J3	J4
W1	13	14	0	8
W2	40	0	12	40
W3	6	64	0	66
W4	0	1	90	0
				(-15)

Step 3: Cover all zeros with a minimum number of lines

	J1	J2	J3	J4	
W1	13	14	**0**	8	
W2	**40**	**0**	**12**	**40**	x
W3	6	64	**0**	66	
W4	**0**	**1**	**90**	**0**	x
			x		

Step 4: Create additional zeros

	J1	J2	J3	J4
W1	7	8	0	2
W2	40	0	18	40
W3	0	58	0	60
W4	0	1	96	0

Step 3: Cover all zeros with a minimum number of li

	J1	J2	J3	J4	
W1	**7**	**8**	**0**	**2**	x
W2	**40**	**0**	**18**	**40**	x
W3	**0**	**58**	**0**	**60**	x
W4	**0**	**1**	**96**	**0**	x

The optimal assignment

	J1	J2	J3	J4
W1	7	8	**0**	2
W2	40	**0**	18	40
W3	**0**	58	0	60
W4	0	1	96	**0**

	J1	J2	J3	J4
W1	82	83	**69**	92
W2	77	**37**	49	92
W3	**11**	69	5	86
W4	8	9	98	**23**

匈牙利算法应用

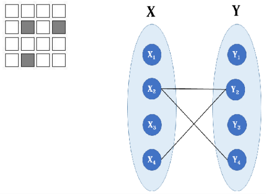
Konig 定理：二分图中最大匹配数等于最小点覆盖数

习题

一种数学锁是一个按钮矩阵，某些格子凸起，每次操作可以把某行或某列的格子按下去，如果能用最少次数按下所有格子，锁就打开，求策略。

解答

- 将矩阵转化为二分图，X, Y 中每个节点对应每一行、列
- 按下一行或一列，就是选择与某个点相连的所有边
- 问题转化为求最小点覆盖数



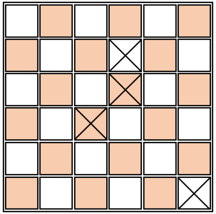
匈牙利算法应用

习题

有一张 $n \times n$ 的国际象棋盘，其中被删除了一些方格，最多能够放下多少 1×2 的多米诺骨牌

解答

- 将棋盘黑白相间染色
- 构造黑白二分图，每个未删除格子与它上下左右未删除格子相连
- 二分图的最大匹配数，就是能放下最多的多米诺骨牌数



课后练习与思考题

- 列举 Peterson 图所有的完美匹配
- 如何用匈牙利算法求最大匹配