# 偶图的算法及应用

南京师范大学附属中学 孙方成

# 目录



# 匹配的概念 (1)

定义1　设图$G=\left(V\left(G\right),E\left(G\right)\right)$，而$M$是$E\left(G\right)$的一个子集，如果$M$中的任两条边均不邻接，则称$M$是$G$的一个匹配。$M$中的一条边的两个端点叫做在$M$下是配对的。

若匹配$M$中的某条边与定点$v$关联，则称$M$饱和顶点$v$，并称$v$是$M$饱和的。

# 匹配的概念 (2)

设$M$是图$G$的一个匹配，若$G$中存在一条基本路径$R$，路径的边是由属于$M$的匹配边和不属于$M$的非匹配边交替出现组成，则称$R$为交替路。若$R$的两个端点都是$M$的非饱和点，则称这条交替路为可增广路。

设图$G=\left(V\left(G\right),E\left(G\right)\right)$，$V\left(G\right)$被分成两个非空的互补顶点子集$X$和$Y$，若图$G$的一个匹配$M$ ô $E\left(G\right)$能饱和$X$中的每个顶点，换言之，$X$中的全部顶点和$Y$中的一个子集的顶点之间确定一个一一对应关系，则称$M$是图$G$的一个完备匹配。

# 偶图的定义

定义2　设图$G=(V,E)$，若能把$V$分成两个集合$V_1$和$V_2$，使得$E$中的每条边的两个端点，一个在$V_1$中，另一个在$V_2$中，这样的图称为偶图，也叫二分图，或是二部图。偶图也可表示为$G=(V_1,V_2;E)$。

对于顶点集$V$ □ $V_1$，用$P(V)$表示$V_2$中所有和$V$相连的顶点的集合。

定义3　如果偶图$G$的互补结点子集$V_1$中的每一结点都与$V_2$中的所有结点邻接，则称$G$为完全偶图。

# 偶图的判定

定理1　当且仅当无向图*G*的每一个回路的次数均为偶数时，*G*才是一个偶图。如果无回路，相当于任一回路的次数为0,0视为偶数。

# 偶图的最大匹配

Edmonds 于 1965 年提出了解决偶图的最大匹配的匈牙利算法：

(1)　从$G$中取一个初始匹配$M$。

(2)　若$X$中的顶皆为$M$中边的端点，止，$M$即为完备匹配；否则，取$X$中不与$M$中边关联的顶$u$，记 $S=\{u\},T=$• 。

(3)　若$P(S)=T$，止，无完备匹配，否则，取 $y$ ô $P(S)-T$。

(4)　若$y$是$M$中边的端点，设$yz$ ô $M$，令 $S$ ô $S$ U $\{z\},T$ ¿ $T$ U $\{y\}$，转(3)；否则，取可增广路径$R(u,y)$，令$M$ ô $M$ ¿ $E(R)$，转(2)。

# 偶图的最小覆盖问题

一般图的最小覆盖问题是一个已被证明的NPC 问题。换一句话说，一般图的最小覆盖问题，是没有有效算法的图论模型。所以，将一个实际问题抽象成最小覆盖问题，是没有任何意义和价值的。

但是，如果问题可以抽象成偶图的最小覆盖问题，结局就不一样了。由于偶图的特殊性，偶图的最小覆盖问题存在多项式算法。

# 最大匹配与最小覆盖的关系

在证明这个定理的过程中，要用到 **Hall 婚姻定理**：

设$G$是一个偶图，顶集划分成$V_1$和$V_2$，$G$中存在对于$V_1$的完备匹配的充要条件是，对于一切$S$ ô $V_1$，都有$|S|$ ô $|P(S)|$。

1931 年 K^nig 给出最大匹配与最小覆盖的关系定理如下：

在偶图$G$中，若$M^*$是最大匹配，$K^*$是最小覆盖集，则$|M^*| = |K^*|$。

# 偶图的最佳匹配问题

定义4　$G=(V_1,V_2;E)$是加权完全偶图，$V_1=\{x_1,x_2,\ldots,x_n\}$，$V_2=\{y_1,y_2,\ldots,y_n\}$，权$w(x_iy_j)\geq 0$。如果有一完备匹配$M$，对所有完备匹配$M'$，都有$W(M)\geq W(M')$，则称$M$为偶图$G$的最佳匹配。

由于引入了权，所以最佳匹配不能直接套用最大匹配算法进行求解。同时，由于对最佳匹配的定义是建立在完全加权偶图的基础上的，对于不完全图，可以通过引入权为 0 （或是其他不影响最终结果的值），使得偶图称为完全偶图，从而使用最佳匹配算法来解决。

# KM 算法前的准备

在介绍求最佳匹配的 KM 算法前，首先介绍一些相关的概念：

定义5　映射$l:V(G)\quad R$满足$\forall x[\ V_1$，$\forall y\quad V_2$，成立

$l(x)+l(y)\ \square\ w(xy)$，则称$l(v)$是偶图$G$的可行顶标；令

$$E_l=\{xy\,|\,xy\ \square\ E(G),l(x)+l(y)=w(xy)\}$$

称以$E_l$为边集的生成子图为“相等子图”，记做$G_l$。

可以证明， $G_l$ 的完备匹配即为 G 的最佳匹配。

以此为基础， 1955 年 Kuhn ， 1957 年 Munkres 给出修改顶标的方法，使新的相等子图的最大匹配逐渐扩大，最后出现相等子图的完备匹配。 这就是 KM 算法。

# KM 算法

(1) 选定初始可行顶标$l$，在$G_l$上选取一个初始匹配$M$。

(2) 若$V_1$中的顶皆为$M$中边的端点，止，$M$即为最佳匹配；否则，取$G_l$中不与$M$中边关联的顶$u$，记$S=\{u\},T=\emptyset$。

(3) 若$P(S)\neq T$,转(4)；若$P(S)=T$，取

$$a_l=\min_{x\in S,y\notin T}\{l(x)+l(y)-w(xy)\}\quad l'(v)=\begin{cases}l(v)-a_l, v\in S,\\ l(v)+a_l, v\in T,\\ l(v)，其它。\end{cases}$$

$l\leftarrow l', G_l\leftarrow G_{l'}$

(4) 选$P(S)-T$中的一点$y$，若$y$是$M$中边的端点，且$yz\in M$，则$S\leftarrow S\cup\{z\},T\leftarrow T\cup\{y\}$，转(3)；否则，取$G_l$中$M$可增广路径$R(u,y)$，令$M\leftarrow M\oplus E(R)$，转(2)。

# 一个例题

某公司有工作人员 $x_1,x_2,\ldots,x_n$ ，他们去做工作 $y_1,y_2,\ldots,y_n$ ，每个人都能做其中的几项工作，并且对每一项工作都有一个固定的效率。问能否找到一种合适的工作分配方案，使得总的效率最高。要求一个人只能参与一项工作，同时一项工作也必须由一个人独立完成。不要求所有的人都有工作。

# 一个实例

| | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ |
|---|---|---|---|---|---|
| $X_1$ | 3 | 5 | 5 | 4 | 1 |
| $X_2$ | 2 | 2 | 0 | 2 | 2 |
| $X_3$ | 2 | 4 | 4 | 1 | 0 |
| $X_4$ | 0 | 1 | 1 | 0 | 0 |
| $X_5$ | 1 | 2 | 1 | 3 | 3 |

若工人 x 完全不能参与工作 y，则 $w(x,y)=0$

# 流程 (1)

首先，选取可行顶标 l(v) 如下：

$$l(y)=0, i=1,2,3,4,5;$$

$$l(x_1)=\max\{3,5,5,4,1\}=5,$$

$$l(x_2)=\max\{2,2,0,2,2\}=2,$$

$$l(x_3)=\max\{2,4,4,1,0\}=4,$$

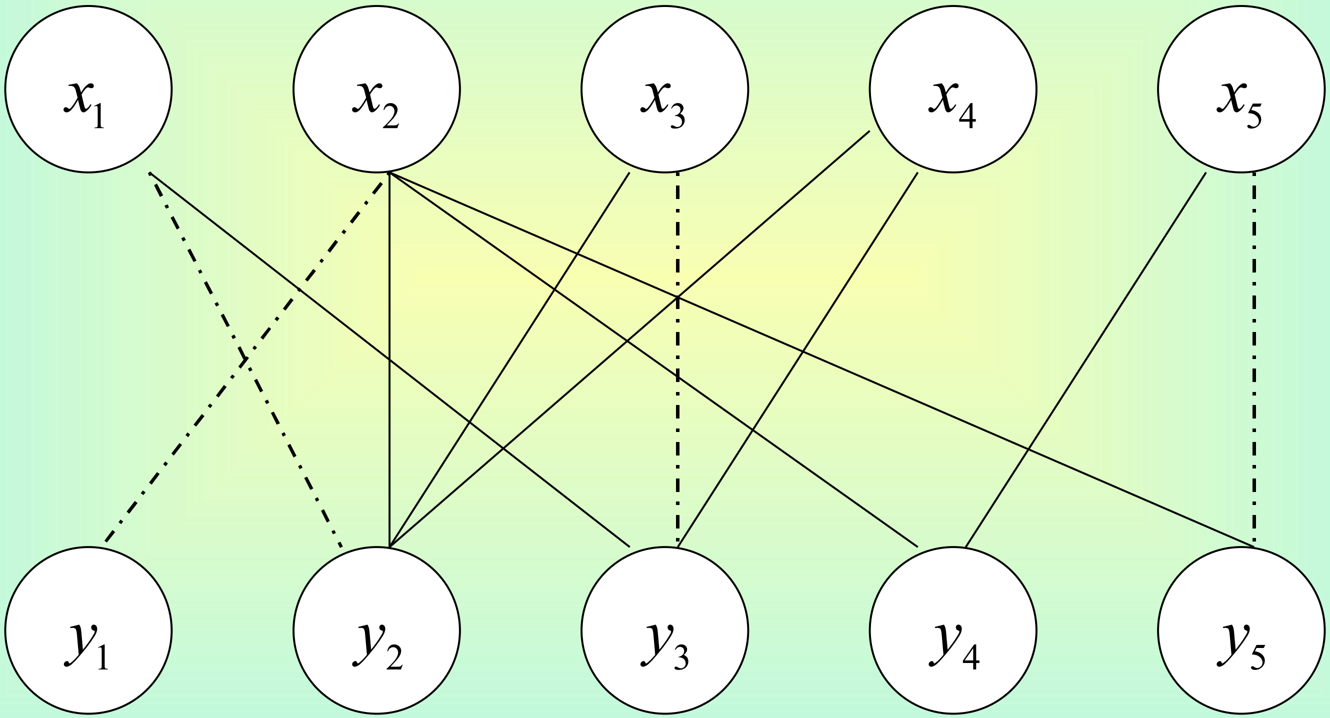
$$l(x_4)=\max\{0,1,1,0,0\}=1,$$

$$l(x_1)=\max\{1,2,1,3,3\}=3$$

构造 $G_1$，并求其最大匹配：（其流程过长，此处略
）

# 流程 (2)

其最终得到的最大匹配如图所示：




图中粗点划线构成最大匹配。

# 流程 (3)

$G_1$ 中无完备匹配，故修改顶标。

由于$u = x_4, S = \{x_4, x_3, x_1\}, T = \{y_3, y_2\}$，所以

$$a_l = \min_{x \in S, y \notin T} \{ l(x) + l(y) - w(xy) \} = 1$$

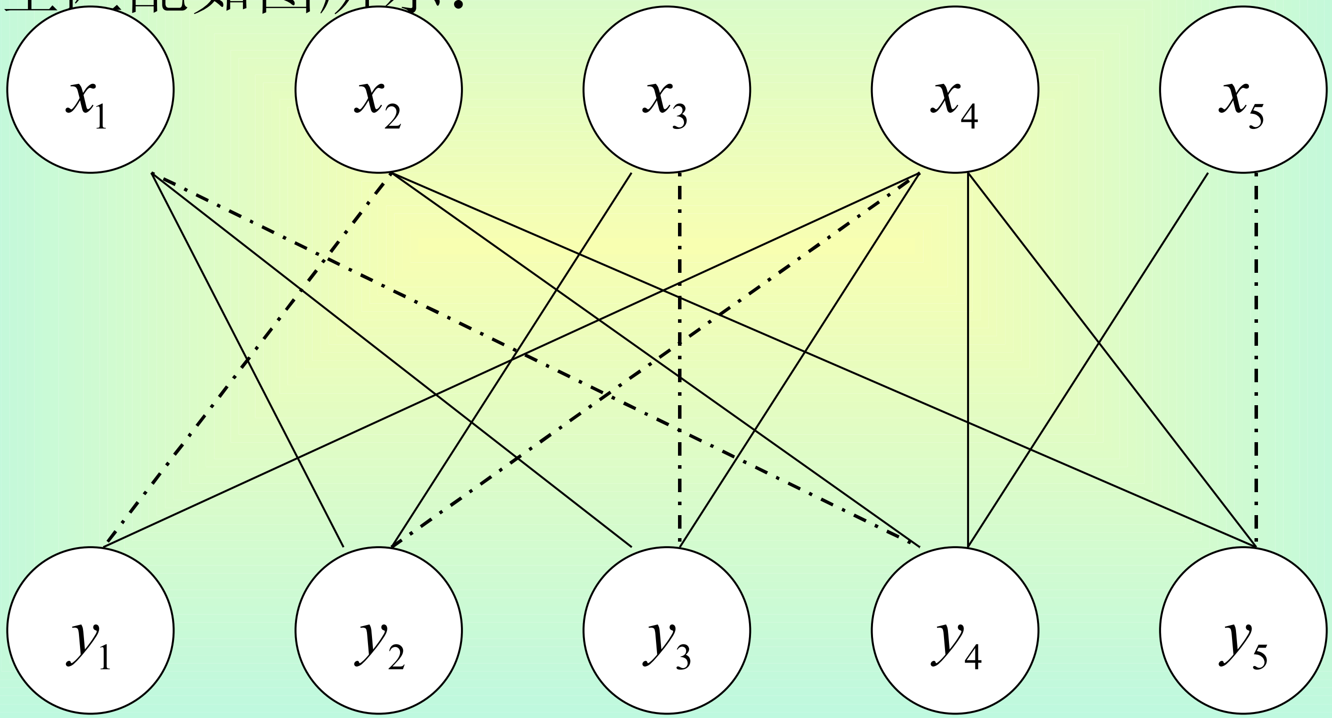
修改后的顶标为：

$$x_1 = 4, x_2 = 2, x_3 = 3, x_4 = 0, x_5 = 3;$$

$$y_1 = 0, y_2 = 1, y_3 = 1, y_4 = 0, y_5 = 0$$

# 流程 (4)

根据新的顶标构造 $G_1$ ，并求其上的一个完全匹配如图所示：




图中粗点划线给出了一个最佳匹配，其最大权为 4 ＋ 2 ＋ 4 ＋ 1 ＋ 3 ＝ 14 。题目完成。

# 小结

偶图是一种特殊的图，所以它不但具备了信息量丰富这个图模型共有的优点，同时它也具备了大量一般图所不具备的内涵和算法优势。

偶图的结点分成两个部分，这就是它和自然界、数学界的对应关系，或者说匹配关系有着深刻的联系。因此，匹配的算法是所有偶图算法的核心。

如果能将实际问题，通过合理的抽象，变成两种事物之间的矛盾，则这种问题就可以抽象成偶图的模型。所以，偶图的模型有着广泛的应用。同时，偶图的算法有着高效实用的特点，所以也使通过偶图模型解决问题成为可能。

综上所述，我认为，偶图是一种高效的，有着广泛使用价值的模型。合理、有效的使用偶图模型，将大大提高编程及解决现实生活中实际问题的能力。

谢谢！