

偶图的算法及应用

南京师范大学附属中学 孙方成

【摘要】

本文首先介绍了匹配这种无向图中特殊的关系，以及偶图这种特殊图的定义。然后将两者结合起来，介绍了偶图的最大基数匹配和最佳匹配的有效算法。同时通过给出有关偶图的最大匹配数和最小覆盖数间的数量关系，说明了和一般图相比，偶图所具有的独特优势。

【关键词】

偶图 匹配 增广路 覆盖集 算法复杂度

一、前言

偶图是一种特殊的图。偶图的结点总是被分成两个互补的部分，这两部分常常用来分别表示两类不同的事物。而两类事物间的最基本的关系，就是匹配的关系。如果能根据具体的情况，将偶图和匹配结合起来，则可以在很大程度上打开思路，优化算法。总之，偶图这种特殊的图，在程序设计中有着广泛的应用。它的高效性有助于对某些复杂问题的较特殊情况，给出完美的解。

二、匹配的概念

定义 1 设图 $G=(V(G),E(G))$ ，而 M 是 $E(G)$ 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 中的某条边与顶点 v 关联，则称 M 饱和顶点 v ，并称 v 是 M 饱和的。

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 $G=(V(G),E(G))$ ， $V(G)$ 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 $M \subseteq E(G)$ 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。

在大多数情况下,如果直接从可增广路的角度求一个图的最大(最佳)匹配,其算法效率较低。所以,对于一般图的匹配算法同时还要涉及到花苞¹的定义即处理。但由于本文主要考虑偶图的匹配问题,所以对这些内容不进行展开。

三、偶图的定义和判定

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。

根据这个定义, V_1 或 V_2 中任何两个顶点都不邻接。偶图也可表示为 $G = (V_1, V_2; E)$ 。对于顶点集 $V \subseteq V_1$, $P(V)$ 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。

对于较简单的图, 可以用肉眼来判断其是否为偶图。然而, 根据定义或用直观的改画方法判定一个较复杂的图是否是偶图是很不方便的, 因此有必要寻求一个行之有效的判定定理。

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。

证: 必要性, 即若 G 是偶图, 则 G 的任一回路的次数为偶数。

设 G 中任意一个长度为 m 的回路 $C = v_{i_0}, v_{i_1}, v_{i_2}, \dots, v_{i_{m-1}}, v_{i_0}$ 。因 G 是偶图, 因此可将 V 分为两个互补结点子集 V_1 和 V_2 , 因 v_{i_0} 和 v_{i_1} 是邻接的, 设

$$v_{i_0} \in V_1$$

$$v_{i_1} \in V_2$$

必有同理可得

$$v_{i_2}, v_{i_4}, \dots, v_{i_{m-2}} \in V_1$$

$$v_{i_3}, v_{i_5}, \dots, v_{i_{m-1}} \in V_2$$

从结点的下标可以看出, 下标为偶数的结点属于 V_1 , 而下标为奇数的结点属于 V_2 , 今 $v_{i_{m-1}}$ 属于 V_2 , 故 $m-1$ 为奇数, 即 m 为偶数。

充分性, 即 G 的任一回路的次数为偶数时, G 必是偶图。分两种情况进行讨论

¹我们把匹配边数达到最大时的奇次回路称为“花苞”。

(1) G 是连通图

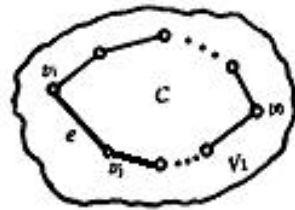
将图的结点集合 V 按下定义分为两个子集。

$$V_1 = \{v_i \mid v_i \text{ 与某一指定结点 } v_0 \text{ 的距离为偶数}\}$$

$$V_2 = V - V_1$$

下面证明 V_1 和 V_2 就是 V 的两个互补结点子集。

任取图 G 的一条边 $e = (v_i, v_j)$ ，如果 e 的两个端点 v_i 和 v_j 都在 V_1 中，如图所示的那样得到一个回路 $C = v_i, \dots, v_0, \dots, v_j, v_i$ ，因 $v_i, v_j \in V_1$ ，按定义 v_i 和 v_j 到 v_0 的距离都是偶数，再加上边 e ，故回路 C 的长度为奇数，与题设矛盾，说明 v_i 和 v_j 不可能都处于 V_1 中。



如果任意边 e 的两个端点 v_i 和 v_j 都在 V_2 中，则由定义 v_i 和 v_j 到 v_0 的距离都是奇数，再加上边 e ，故回路 C 的长度仍为奇数，也与题设矛盾，说明 v_i 和 v_j 也不可能都处于 V_2 中。因此只有唯一一种可能，即 e 的两个端点，一个在 V_1 中而另一个在 V_2 中，由于 e 是任意一条边，根据偶图的定义， G 是偶图。

(2) G 是非连通图

此时可分片讨论，对 G 的每个连通分量应用上面的证明，然后合并起来，即可得证。

四、偶图的最大匹配

根据偶图的定义，它在继承了一般图的性质的同时，更有一些特殊的性质。因此，偶图的最大匹配算法相对于一般图的最大匹配算法较为简单，也有更广泛的适用性。可以说，偶图的最大匹配算法是所有偶图算法的基础。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 $S = \{u\}, T = \emptyset$ 。
- (3) 若 $P(S) = T$ ，止，无完备匹配，否则，取 $y \in P(s) - T$ 。
- (4) 若 y 是 M 中边的端点，设 $yz \in M$ ，令 $S \leftarrow S \cup \{z\}, T \leftarrow T \cup \{y\}$ ，转(3)；

否则, 取可增广通路 $R(u, y)$, 令 $M \leftarrow M \oplus E(R)$, 转(2)。

分析上述算法, 可以得到求偶图最大匹配的算法的时间复杂度为:
 $O(|E(G)|)$ 。

在现实生活中, 有很多的问题都可以转化成偶图的最大匹配问题, 例如工作安排问题等。这里, 举一个相对较新的例子。《小狗散步》选自 NOI2001 上海市选拔赛试题:

Grant 喜欢带着他的小狗 Pandog 散步。Grant 以一定的速度沿着固定路线走, 该路线可能自交。Pandog 喜欢游览沿途的景点, 不过会在给定的 N 个点和主人相遇。小狗和主人同时从 (X_1, Y_1) 点出发, 并同时在 (X_n, Y_n) 点汇合。小狗的速度最快是 Grant 的两倍。当主人从一个点以直线走向另一个点时, Pandog 跑向一个他感兴趣的景点。Pandog 每次与主人相遇之前最多只去一个景点。

你现在的任务是: 为 Pandog 寻找一条路线 (有可能与主人的路线部分相同), 使它能够游览最多的景点, 并能够准时与主人在给定地点相遇或者汇合。

如果直接考虑小狗的行进路线, 每到达一个新的汇合点, 都要考虑小狗的行动方案。总的计算量为 P_m^n , 这种指数级复杂度的算法无疑是低效的, 不切合实际的。同时, 由于题目模型本身和排列模型的相关性。过高的后效性, 从根本上否定了动态规划算法的可能性。即使是使用网络流模型, 如果单从狗的角度直接构造, 也非常难以实现。但如果能抓住题目中某些关键性叙述, 将其抽象成偶图的模型, 则能有较好的效果。

由于“当主人从一个点以直线走向另一个点时, Pandog 跑向一个他感兴趣的景点”, 并且主人的移动路线是确定的, 所以寻找一条路线, 可以看作确定在每一段路线上, 小狗的选择: 跑向哪一个景点, 或是和主人的路线重合。显然, 如果把主人的移动路线的每一段都抽象成点, 其性质和小狗所向往的景点的性质是不同的。所以图中的点被自然的分成了两类, 题目要求的也就是这两类点之间的关系。

由于在一个线段上, 小狗只能跑向一个景点, 所以景点的可达与否只和景点到这个线段的两端点的距离有关。由于一条线段只能和一个景点发生关联, 而一个景点也只在第一次访问的时候才有意义, 所以任何两项关联都不会有任何交点。这显然满足匹配的定义。所以, 题目的所求就是在这样一个由两类点所组成的图中, 两个点集间的最大匹配。

偶图的构造过程如下:

设共有 n 个定点和 m 个景点, $V_1 = \{\overline{P_i P_{i+1}} \mid 1 \leq i < n\}$; $V_2 = \{Q_i \mid 1 \leq i \leq m\}$;

$V(G) = V_1 \cup V_2$ 。对于有向线段 $\overline{P_i P_{i+1}}$ 和景点 Q_k , 当且仅当

$D(P_i, Q_k) + D(Q_k, P_{i+1}) \leq 2D(P_i, P_{i+1})$ 时, 边 $(\overline{P_i P_{i+1}}, Q_k)$ 属于 $E(G)$ 。其中 $D(A, B)$ 定

义为 $\sqrt{(X_A - X_b)^2 + (Y_A - Y_b)^2}$ 。参照偶图的定义，这是一个典型的偶图。然后就是根据求偶图的最大匹配的匈牙利算法，求出其最大匹配数及匹配方案，最后按题目要求生成输出。

五、偶图的最小覆盖问题

偶图的最大匹配算法在解决实际问题中，有着广泛的应用。最典型的就是利用偶图的最大匹配算法解决偶图的最小覆盖问题。如 **BOI2001** 的《Knights》一题，问题描述如下：

一张大小为 $n \times n$ 的国际象棋棋盘，上面有一些格子被拿走了。你的任务是确定在这个棋盘上放置尽可能多的马并使他们不互相攻击。棋盘的大小 n 不超过 200。

	X		X	
X				X
		S		
X				X
	X		X	

一只马放在 S 处，攻击所有标有 X 的点

题目要求放置尽可能多的马，并要求不能相互攻击。类似这种描述的题目非常普遍，最常见的就是 8 皇后问题。8 皇后问题是回溯算法的典型例题，由于在每一行中对几乎每一个点都要进行一次穷举，所以事件复杂度为 $O(n!)$ ，更是经典的 NP 问题。如果按照 8 皇后问题的思路来解决这道题，面临的挑战是十分严峻的。

首先，8 皇后问题的扩展方式远比本题简单。由于皇后的攻击范围是完整的一行一列，或是一条斜线，所以在扩展时可以大量的回避无效状态，而本题的攻击范围相对复杂，所以在搜索过程中直接回避无效状态的难度较大，算法效率自然不理想。其次，本题的问题规模达到 200，已经远远超过 8 皇后问题的可解范围。同时由于马控制范围较小，必然带来可行方案的增多，而回溯或是其它的搜索算法都是基于求方案而得到问题的解，其基本思想还是穷举，解的增多必然带来效率的大幅降低。而本题的求解目标却是针对最多放置马的数目，搜索类算法显然针对性太差。

然后考虑 NOI2001 中《炮兵阵地》一题。在《炮兵阵地》一题中，采用了状态分离的办法，最终实现了动态规划。但动态规划过程中，如果相对于宽度来说，算法复杂度仍然是指数级的。这道题和《炮兵阵地》相比，攻击范围的不规则性同样是本题的劣势，在状态转移过程中的难度必然大大增加。更主要的是，动态规划解决《炮兵阵地》问题的关键在于长宽的极端不对称性。而在本题中，棋盘是严格的正方形，状态分离的思想很难得到实现（时空上都无法得到保证）。所

以说动态规划的方法也不适用本题。也就是说，从常规思路类比考虑，无法得到关于本题的可行解法。

下面，从构造图的角度分析本题的模型。将棋盘的每一个格子作为一个顶点，两点间直接可达（互可攻击）的关系作为边。则题目所要求的就是在这样一张图中的最大独立子集。由于对于任意一张图，其最大独立子集和最小覆盖集互为补集，所以本题也是一个求最小覆盖集的图论问题。众所周知，一般图的最小覆盖图是一个 NPC 问题。那么本题能否逃脱厄运呢？这只有从本题的特殊性来分析。

常识告诉我们，在一张国际象棋的棋盘上，白格和黑格是相交替的，一匹马只能从一个白格跳到一个黑格，或是从一个黑格跳到一个白格。由于这种特殊性，马的攻击范围的不规则，在这时却给我们的解题带来了方便。因为一个位置上的马只能攻击和其所在位置颜色不同的格子，这是一个必要条件。换句话说，同一颜色的任意两个格子间不存在边。所以，棋盘可以按照颜色分成两个部分，在同一部分的任意两点都是不能互相攻击的。也就是说，题述所描述的棋盘，定义在攻击关系上是一个偶图。

同时，由于偶图的独特性质，求它的最小覆盖数有着多项式的算法。偶图的最大匹配数和最小覆盖数之间存在着确定的数量关系。1931 年 König 给出最大匹配与最小覆盖的关系定理如下：

定理 2 设 r 是偶图 G 的最大匹配数， s 是其邻接矩阵的最小覆盖数，则有 $r = s$ 。

要证明这个定理，要用到 **Hall 婚姻定理**：设 G 是一个偶图，两个顶点集分别为 V 和 W 。则存在 G 的一个完备匹配，当且仅当 $|S| \leq |P(S)|$ 对所有 $S \subseteq V$ 均成立。对于这个定理的证明将在附录中给出。下面证明最大匹配最小覆盖定理：

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 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 $s \geq r$ 。再证明 $r \geq s$ 。不失一般性，设最小覆盖盖住 A 的 c 行、 d 列，即 $s = c + d$ 。设这 c 行对应的结点子集是 X_c 其余为 $X - X_c$ ； d 列对应的结点集是 Y_d ，其余为 $Y - Y_d$ 。把矩阵 A 的行、列进行调整，如图所示，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 $A_{21} = 0$ 。

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

现证明 X_c 到 $Y - Y_d$ 存在完备匹配。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，

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' , 总有 $|P(V')| \geq |V'|$ 。根据 Hall 婚姻定理, X_c 到 $Y - Y_d$ 存在完备匹配 M_1 , $|M_1| = c$ 。同理 Y_d 到 $X - X_c$ 也存在完备匹配 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 $r \geq s$ 。证明结束。

由于这个定理的存在, 有关偶图的最小覆盖集、最大独立子集等算法, 都可以通过求偶图的最大匹配数来间接得到。

六、偶图的最佳匹配问题

前文给出了偶图的最大匹配的算法, 但在实际的工作中, 效率往往是考虑问题时必不可少的一环。比如说工作效率不同的工作安排问题, 或是在《小狗散步》一题中加入各个景点的权信息……仅仅用最大匹配就无法求解。下面举一个加权的工作安排问题, 问题描述如下:

某公司有工作人员 x_1, x_2, \dots, x_n , 他们去做工作 y_1, y_2, \dots, 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。

对于如右方表格所示的数据 (如果员工完全不能完成一项工作, 则效率用 0 来表示), 如果采用最大基数匹配算法, 得到的解为: 甲—A, 乙—B, 丙—C, 丁—D, 最后总的价值为 4。但是, 如果采用甲项目不匹配, 乙—A, 丙—C, 丁—D, 虽然匹配数较少, 但是最后的总价值却达到了 12。事实证明, 最大匹配的算法对解决这类效益至上的问题没有实际的效果。

	甲	乙	丙	丁
A	1	10	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

那么, 是否有解决这类问题的其它有效途径呢? 由于一个人只能做一项工作, 一项工作也只能由一个人做, 所以在方案的决策上所带来的后效性是十分明显的。如果要通过改变状态的描述方式来摆脱后效性的麻烦, 也必将带来维数灾难的动态规划致命问题。所以说, 动态规划不能用来解决此类问题。

可以用来解决此类问题的可行办法是分支定界算法。也就是通过估价函数的合理选择, 在深度搜索的基础上, 不断的更新最优结果, 以达到减少无效搜索, 提高效率的目的。然而, 分支定界算法虽然在中小规模的应用中有着不错的效率, 但是其本身指数级的复杂度, 导致其效率随着规模的增加而急剧降低。

实际上, 这类问题就是图论中的最佳匹配问题。问题描述如下:

$G = (V_1, V_2; E)$ 是加权完全偶图, $V_1 = \{x_1, x_2, \dots, x_n\}$, $V_2 = \{y_1, y_2, \dots, y_n\}$, 权 $w(x_i y_j) \geq 0$, 求权最大的完备匹配, 这种匹配称为最佳匹配。由于定义中要求的是加权完全偶图 (从算法角度分析, 这种定义的主要目的是保证完备匹配的必然存在)。所以, 当原图中某些边不存在时, 往往通过将边的权赋为 0 或 1 或是其

它一些不影响最终结果的值,使边达到完备。在做好这些准备工作后,具体解决这类问题之前,首先要了解一些相关概念。

定义 4 映射 $l:V(G) \rightarrow R$ 满足 $\forall x \in V_1, \forall y \in V_2$, 成立 $l(x)+l(y) \geq w(xy)$,

则称 $l(v)$ 是偶图 G 的可行顶标; 令

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

称以 E_l 为边集的 G 的生成子图为“相等子图”, 记成 G_l 。

可行顶标是存在的, 例如

$$l(v) = \begin{cases} l(x) = \max_{y \in V_2} w(xy), x \in V_1 \\ l(y) = 0, y \in V_2 \end{cases}$$

定理 3 G_l 的完备匹配即为 G 的最佳匹配。

证: 设 M^* 是 G_l 的一个完备匹配, 因为 G_l 是 G 的生成子图, 故 M^* 也是 G 的完备匹配。 M^* 中的边的端点集合含 G 的每个顶点恰一次, 所以

$$W(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in V(G)} l(v)$$

另一方面, 若 M 是 G 中任一完备匹配, 则

$$W(M) = \sum_{e \in M} w(e) \leq \sum_{v \in V(G)} l(v)$$

所以 $W(M^*) \geq W(M)$, 即 M^* 是最佳匹配。

根据定理 3, 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。

Kuhn-Munkres 算法 (KM 算法)

(1) 选定初始可行顶标 l , 在 G_l 上选取一个匹配 M 。

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

(3) 若 $P_{G_l}(S) \supset T$, 转 (4); 若 $P_{G_l}(S) = T$, 取

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

$$l'(v) = \begin{cases} l(v) - a_l, v \in S, \\ l(v) + a_l, v \in T, \\ l(v), \text{ 其它。} \end{cases}$$

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

(4) 选 $P_{G_l}(S)-T$ 中一顶 y ，若 y 与 M 中边关联，且 $yz \in M$ ，则 $S \leftarrow S \cup \{z\}, T \leftarrow T \cup \{y\}$ ，转(3)；否则，取 G_l 中一个 M 的可增广通路 $R(u, y)$ ，令 $M \leftarrow M \oplus E(R)$ ，转(2)。

下面，通过一个实例来进一步了解算法的具体实现和各方面的效率。设五名员工参加五个项目的效率如下表所示：

	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

首先，选取可行顶标 $l(v)$ 如下：

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

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

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

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

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

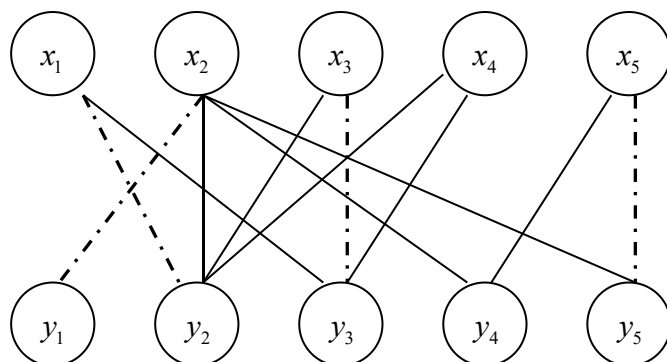
$$l(x_5) = \max \{1, 2, 1, 3, 3\} = 3$$

构造 G_l ，并求其最大匹配，流程如下：

步骤编号	S	T	M
1、选取初始匹配	—	—	\emptyset
2、取 $u = x_1$	$\{x_1\}$	\emptyset	\emptyset
4、取 $y = y_2$	—	—	$\{e(x_1y_2)\}$
2、取 $u = x_2$	$\{x_2\}$	\emptyset	$\{e(x_1y_2)\}$
4、取 $y = y_1$	—	—	$\{e(x_1y_2), e(x_2y_1)\}$
2、取 $u = x_3$	$\{x_3\}$	\emptyset	$\{e(x_1y_2), e(x_2y_1)\}$
4、取 $y = y_3$	—	—	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3)\}$
2、取 $u = x_5$	$\{x_5\}$	\emptyset	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3)\}$

4、取 $y = y_5$	—	—	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3), e(x_5y_5)\}$
2、取 $u = x_4$	$\{x_4\}$	\emptyset	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3), e(x_5y_5)\}$
4、取 $y = y_3$	$\{x_4, x_3\}$	$\{y_3\}$	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3), e(x_5y_5)\}$
4、取 $y = y_2$	$\{x_4, x_3, x_1\}$	$\{y_3, y_2\}$	$\{e(x_1y_2), e(x_2y_1), e(x_3y_3), e(x_5y_5)\}$
2、 $P_{G_l}(S) = T$	无完全匹配		

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。 G_l 中无完备匹配，故修改顶标。

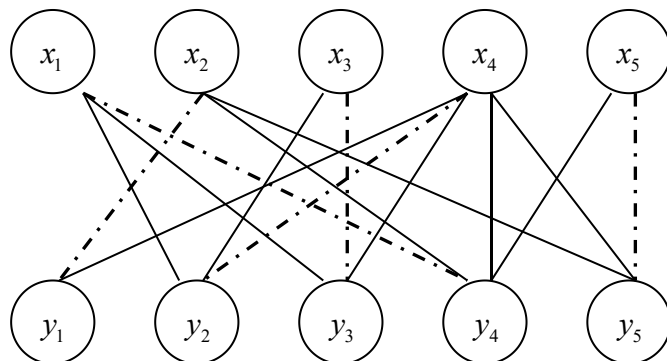
由于 $u = x_4, S = \{x_4, x_3, x_1\}, T = \{y_3, y_2\}$ ，所以

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

修改后的顶标为：

$$\begin{aligned} x_1 &= 4, x_2 = 2, x_3 = 3, x_4 = 0, x_5 = 3; \\ y_1 &= 0, y_2 = 1, y_3 = 1, y_4 = 0, y_5 = 0 \end{aligned}$$

根据新的顶标构造 G_l ，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。

通过解题的过程，并对比 KM 算法和匈牙利算法的描述可以发现，除了 KM

算法中第三部分修改顶标的内容外，绝大部分是相同的。可以说，KM 算法就是在匈牙利算法的基础上加入了修改顶标的部分。由于 M 是逐步增大的，所以对于每一个 G_i 的增广过程中产生的信息，都很好的得到了继承和合理的利用。因此，KM 算法虽然加入了对权的考虑，有着更广泛的适用性，但从时间复杂度的数量级上是和匈牙利算法本质相同的，也是 $O(|E(G)|)$ 。

七、小结

偶图是一种特殊的图。对于一个偶图 G ，其点集 $V(G)$ 可被分成两个互补的子集 V_1, V_2 ，对于属于同一子集的两点 x, y ，都有边 $e(x, y) \notin E(G)$ 。由于这种特殊性，偶图作为描述现实世界中两类不同事物间的相互关系的有效模型而得到广泛的应用。

图结构因为其包含信息的多元化，往往能反映出问题的本质属性，抓住问题的关键。尤其是在解决动态规划的后效性、维数灾等问题时，一般能发挥出较好的效果。又由于建立在偶图上的算法有很多区别于一般图算法的优势，所以将一个一般图转换成偶图的形式，在程序设计中更是有很大作用。而转换的关键，一般是从题目本身的条件出发，挖掘题目中深层次的信息，从而将关系复杂的图的结点进行分类。例如在棋盘上，染色往往是一种很好的分类方式。

在建立起偶图的模型后，求解也是十分重要的。本文给出了偶图的一些基本算法和相关定理。通过合理的使用这些信息，基本上可以解决程序设计中遇到的大部分偶图问题。如果有更多的限制和要求，可以通过最大流等更复杂的模型来解决。但是，从算法效率和编程复杂度上说，基于偶图的算法一般比基于最大流的算法简单高效。

对于匹配算法，除偶图的匹配之外，对一般图的匹配也有定义，但求解算法相对复杂。同时，我个人认为，由于匹配是两物品之间一一对应的关系。所以建立两类事物间的匹配关系，也就是偶图的匹配，比一般图的匹配有着更高的实用价值。

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。

【附录】

Hall 婚姻定理的证明如下：

显然，对 X 每一个子集 A ， $|A| \leq |P(A)|$ 是存在 X 到 Y 内的完备匹配的必要条件。因为，若有 X 的一个子集 A_0 ，使得 $|A_0| > |P(A_0)|$ ，则不可能在 A_0 的顶点和 Y 的子集中的顶点之间确定一个一一对应关系。下面，用两种方法证明条件的充分性。

用归纳法证明条件的充分性。

对于 X 中只有一个顶点的任何偶图，条件显然是充分的，因为这个顶点必定同 Y 中至少一个顶点相邻接。现假定，对于 X 中有不多于 $m-1$ 个顶点的偶图这个条件是充分的。假设存在一个偶图 G ， X 中有 m 个顶点，对 X 的每一个子集都有 $|A| \leq |P(A)|$ ，考察下面两种情况，从而证明存在 X 到 Y 内的一个完备匹配：

(1) 对 X 的每一非空真子集 A ，都有 $|A| < |P(A)|$ 。可取 X 中任一顶点 x_0 且把它与 Y 中一个顶点 y_0 相匹配。从 G 中移去顶点 x_0, y_0 以及它们相关联的全部边，得到图 G' 。在偶图 G' 中，因 $X - \{x_0\}$ 的每一个子集中的顶点都同 $Y - \{y_0\}$ 中的至少 $|A|$ 个顶点相邻接，故由归纳假设，存在 $X - \{x_0\}$ 到 $Y - \{y_0\}$ 内的一个完备匹配。因此，图 G 中存在 X 到 Y 的一个完备匹配。

(2) 存在 X 的一个非空真子集 A_0 ，且 $|A_0| = |P(A_0)|$ 。设 G' 是 G 的这样一个子图，它包含顶点集 A_0 以及同 A_0 中的顶点相邻接的顶点集 $P(A_0)$ ，和连接这两个顶点集中的顶点之全部边。由于 A_0 的每一子集 A 中的顶点同 $P(A_0)$ 中至少 $|A|$ 个顶点相邻接，故由归纳假设，在图 G' 中存在 A_0 到 $P(A_0)$ 内的一个完备匹配。设 G'' 中包含顶点集 $X - A_0$ 到 $Y - P(A_0)$ 以及连接这两个子集中的顶点的边的子图，可以断言， G'' 中有 $X - A_0$ 到 $Y - P(A_0)$ 内的完备匹配。如若不然，则由归纳假设，存在 $X - A_0$ 的一个子集 A_1 ，在图 G'' 中， A_1 中的顶点同 $Y - P(A_0)$ 中少于 $|A_1|$ 个顶点相邻接，这意味着，在图 G 中，子集 $A_0 \cup A_1$ 中的顶点同少于 $|A_0 \cup A_1|$ 个顶点相邻接，这同假设“对 X 的每一个子集 A 都有 $|A| \leq |P(A)|$ ”相矛盾。因此，在图 G'' 中有 $X - A_0$ 到 $Y - P(A_0)$ 内的一个完备匹配。由此推知，在图 G 中有 X 到 Y 内的一个完备匹配。证毕。

用反证法证明条件的充分性。

设 G 不存在完备匹配，则 V_1 中 $|V_1| = n_1$ 个结点不能与 V_2 中 n_1 个不同结点配对，现在假设找到 G 的一个最大基数匹配 M^* ，则中至少有一个结点不是 M^* 饱和点（否则 M^* 就是完备匹配了），设这个结点为 v_0 ，我们从 v_0 开始找出所有 M^* 的交替路，并把这些交替路上的结点集合记作 S ， S 中属于 V_1 的结点集合记作 A ，属

于 V_2 的结点集合记作 B ，即 $S = A \cup B$ ； $A = S \cap V_1$ ； $B = S \cap V_2$ 如图所示。

根据交替路的性质可知，与 A 中结点邻接的结点一定是 M^* 饱和点，这些结点一定都在从 v_0 开始的 M^* 交替路上，因此有

$$P(A) = B \quad (1)$$

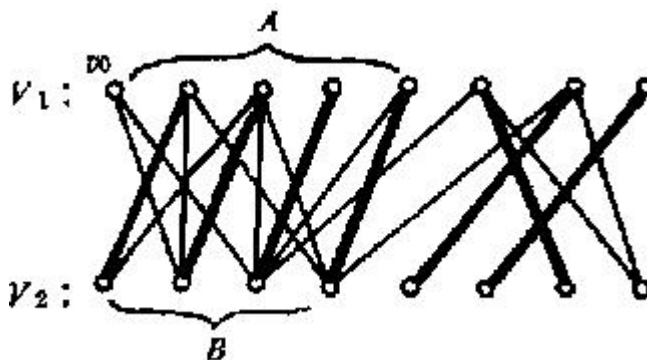
此外， A 中除 v_0 外，它的每个结点都与 B 中的结点一一配对，因此有

$$|B| = |A| - 1 \quad (2)$$

由(1)和(2)可得

$$|P(A)| = |A| - 1 < |A|$$

与题设条件矛盾。得证。



【参考文献】

1. 《组合数学及其算法》 中国科学技术大学出版社 杨振生 编著
2. 《图论及其算法》 航空工业出版社 肖位枢 主编
3. 《信息学奥林匹克》季刊 全国青少年信息学（计算机）奥林匹克分区联赛组织委员会主办
4. 《数学建模竞赛教程》 江苏教育出版社 李尚志 主编

除此以外，还参考了 BOI, USACO 以及 ACM 的相关试题及解答，综合了与江苏省其它同学的讨论成果。