图论模型

数学系 孙艳蕊
yanruisun@126.com

主要内容

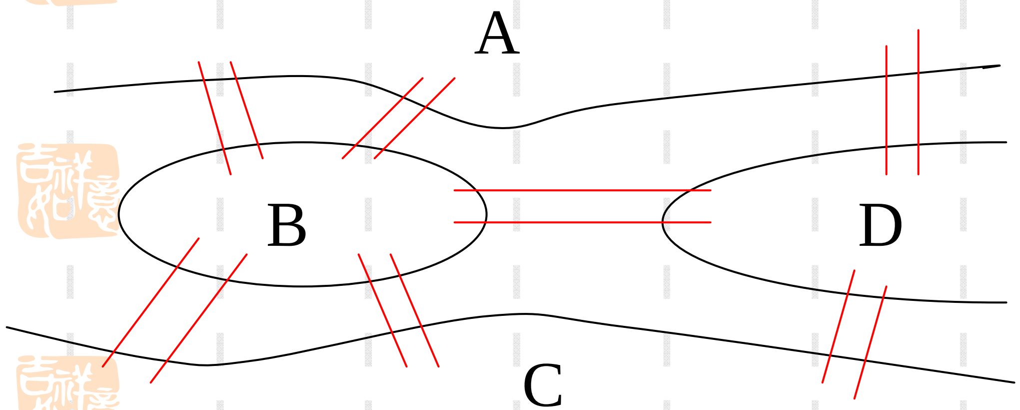
- 图论模型
- 图论的基本概念
- 最短路问题
- 最小生成树问题
- 旅行售货员问题
- 最大流问题
- 匹配问题

一、图论模型

- 例1. 哥尼斯堡七桥问题(18世纪)

(1) 问题

能否从一块陆地出发,走遍每座桥一次且仅一次然后回到出发地?

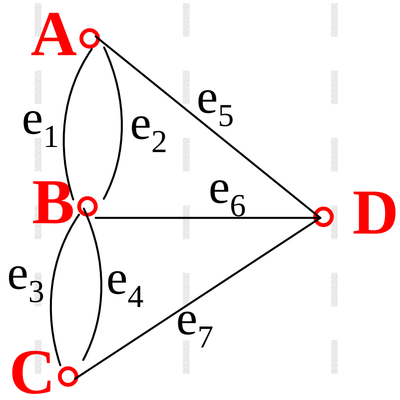


(2) 问题分析与模型假设

问题的本质是能否从一地无重复地一次走遍七桥，与所走过的桥的大小、形状、长短、曲直等均无关，因此不妨将其视为一条弧线；

四块陆地可重复经历,至于陆地的大小、形状、质地等也与问题的无关，因而可视四块陆地为四个点 **A**、**B**、**C**、**D**。

对四个陆地 **A**、**B**、**C**、**D**，若其间有桥，则用一条弧线连接起来，有两座桥，则连两条不重合的弧线，便得到一个图，并称代表陆地的四个点为顶点，代表桥的弧线为边。这样，能否从一地出发走遍七座桥一次且仅一次再回到出发点就变成了：能否从这个图上任一顶点出发，经过每条边一次且仅一次而回到出发顶点。



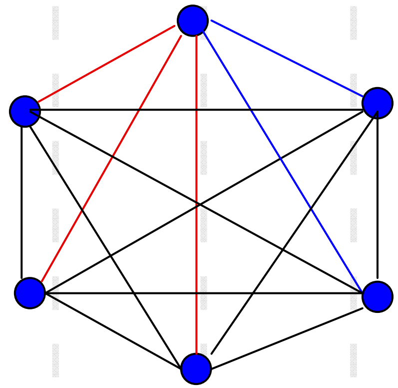
例2 现有6个人，任意两人之间或者相互认识，或者相互不认识，证明这6个人中，或者有3个人彼此认识，或者有3个人彼此不认识。

分析：枚举$2^{15} = 2^5\ 2^4\ 2^3\ 2^2\ 2^1 = 2^{15}$

计算量很大，借助图模型解决.

以人为顶点，关系为边，

构造一无向图.问题转化为：

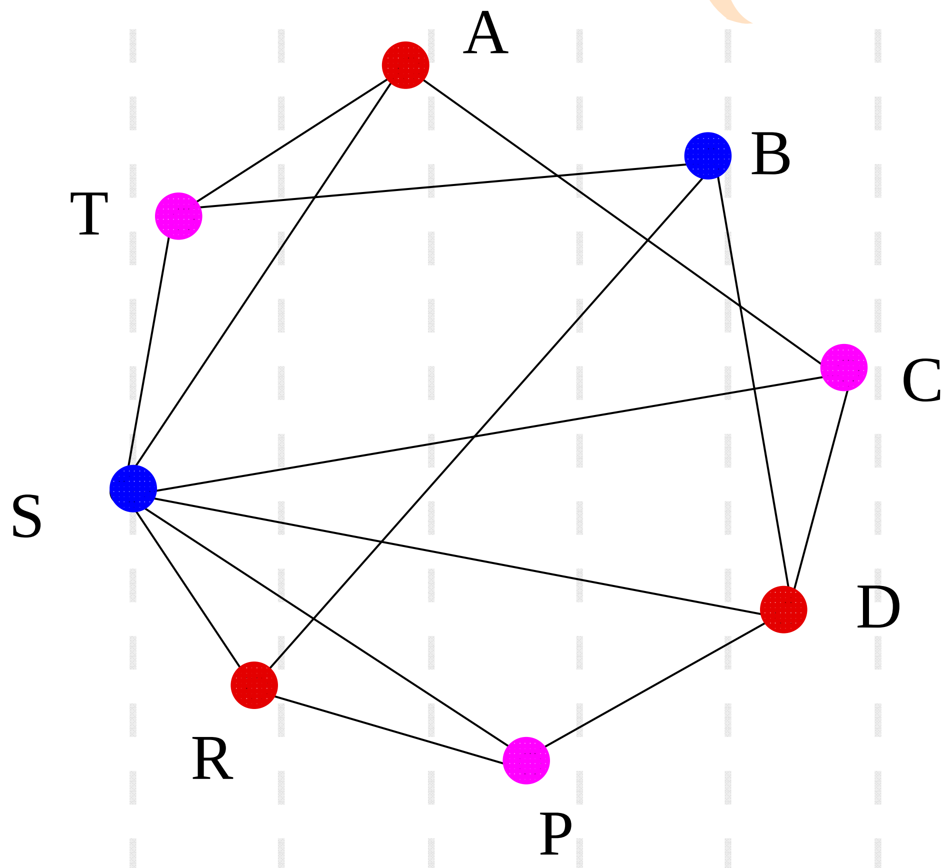


证明这个图中或存在一红三角形(认识)，或存在一蓝三角形(不认识)。

例3 药品存储问题

- 有8种化学药品A、B、C、D、P、R、S和T要放进贮藏室保管,出于安全原因,下列各组药品不能贮在同一室内:A—R，A—C，A—T，R—P，P—S，S—T，T—B，B—D，D—C，R—S，R—B，P—D，S—C，S—D，试为这8种药品设计一个使用房间数最少的贮藏方案。

A—R，A—C，A—T，
R—P，P—S，S—T，
T—B，B—D，D—C，
R—S，R—B，P—D，
S—C，S—D.



至少需用3个房间。

例4 过河问题：摆渡人Ferryman,狼wolf,羊sheep,卷心菜cabbage过河问题．如何摆渡使得它们不能互相伤害.

例5 考试安排问题：学校期末考试安排*n*门课的考试时间时，不能把同一位学生选修的两门课安排在同一时间考试，问学校考试最少要进行多长时间？

例6 信道分配问题：发射台所用频率从小到大编号为1,2, ...称为信道。用同一信道的两个台站相距得少于一个常数*d*，问各台至少需同时使用几个不同的信道？…………

二、图论的基本概念

1. 图的概念

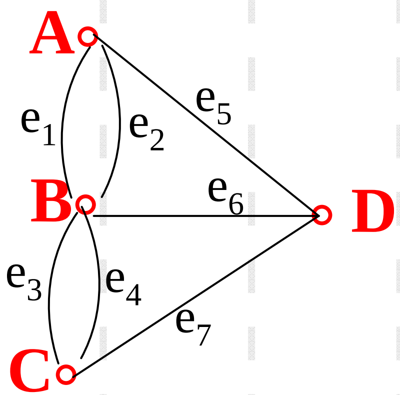
一个图G是指一个二元组 $(V(G), E(G))$， 其中 $V(G)=\{v_1,v_2,\cdots,v_\nu\}$是非空有限集，称为顶点集，$V(G)$中的元素称为图$G$的顶点；$E(G)$是$V(G)$中的无序或有序的元素对 (v_i,v_j) 组成的集合，称为边集, $E(G)$中的元素称为边.

$|V(G)|$：图的顶点数； $|E(G)|$：图的边的数。

用 $G=(V(G),E(G))$ 表示图，简记 $G=(V,E)$.

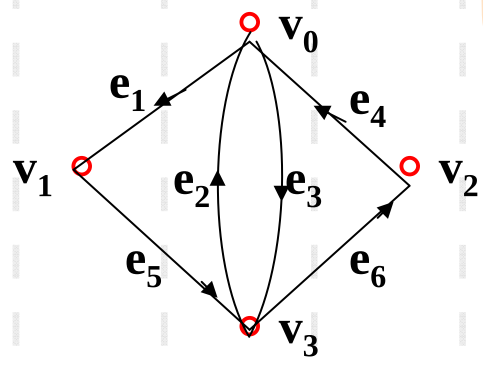
v_iv_j 表示边 (v_i,v_j).

无向图



V={A,B,C,D},

E={e_1,e_2,e_3,e_4,e_5,e_6,e_7}



有向图

V={v_1,v_2,v_3,v_4},

E={e_1,e_2,e_3,e_4,e_5,e_6}

图：无向图，有向图和混合图。

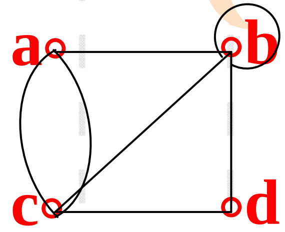
孤立结点:不与任何边关联的结点.

零图:仅由一些孤立结点构成的图.

平凡图：只有一个顶点的图。

常用术语

1) 边和它的两端点称为互相关联.



2)与同一条边关联的两个端点称为相邻的顶点，与同一个顶点关联的两条边称为相邻的边.

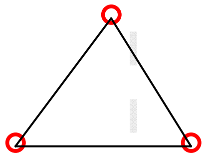
3) 端点重合的边称为环，
端点不相同的边称为连杆.

4) 若一对顶点之间有两条以上的边联结，则这些边称为重边.

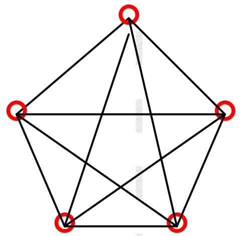
5) 既没有环也没有重边的图，称为简单图.

常用术语

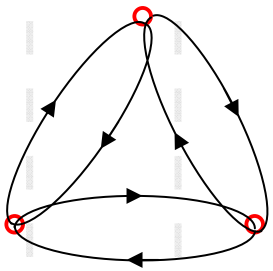
6) 任意两顶点都相邻的简单图,称为完全图. 记为K_v.



K_3



K_5

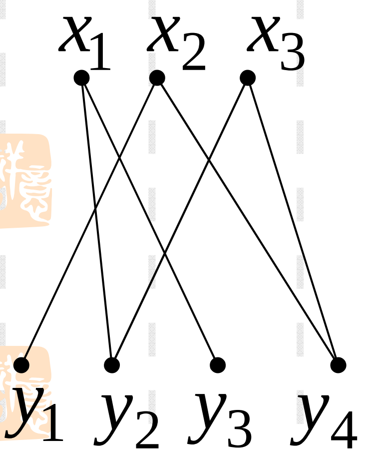


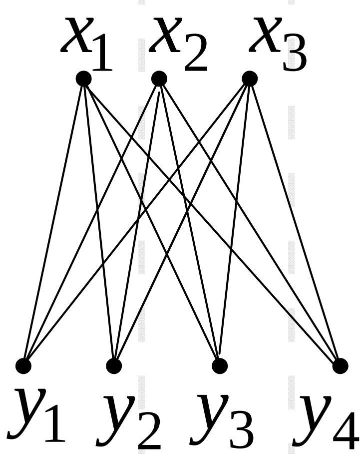
7) 若 $V(G)=X\cup Y$ $X\cap Y=\phi$

且两个点集内部 任意两顶点不相邻，

则称 G 为二部图或偶图；若$\boldsymbol{X}$中每一顶点皆与$\boldsymbol{Y}$中一切顶点相邻,称为完全二部图或完全偶图，

记为 $K_{m,n}(m=|X|,n=|Y|)$.





$K_{3,4}$

2．赋权图与子图

定义 若图 $G=(V(G),E(G))$ 的每一条边e 都赋以一个实数$w(e)$，称$w(e)$为边e的权，G 连同边上的权称为赋权图.

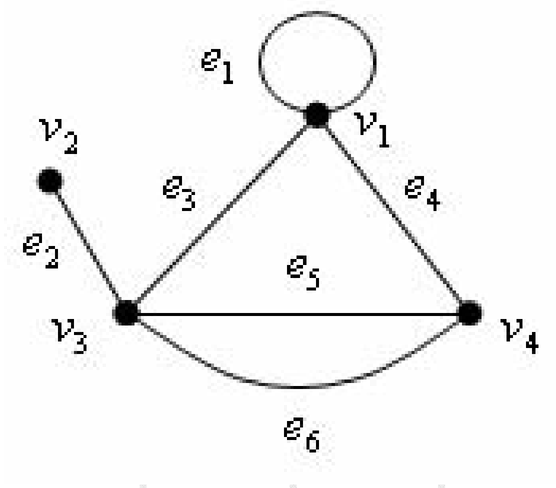
定义 设 $G=(V,E)$ 和 $G'=(V',E')$ 是两个图.

1) 若 $V'\subseteq V, E'\subseteq E$,称 G' 是 G 的一个子图.

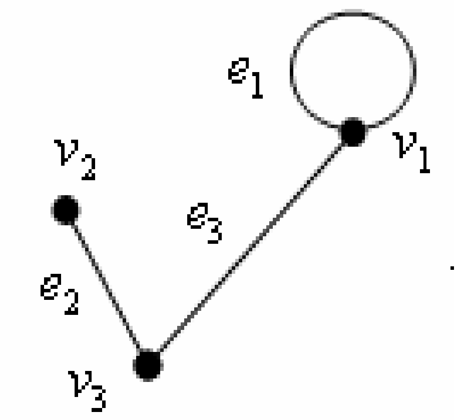
2) 若 $V'=V, E'\subseteq E$，则称 G' 是 G 的生成子图.

3) 若 $V'\subseteq V$，且 $V'\neq\phi$，以 V' 为顶点集，以两端点均在 V' 中的边的全体为边集的图 G 的子图，称为 G 的由 V' 导出的子图，记为 $G[V']$.

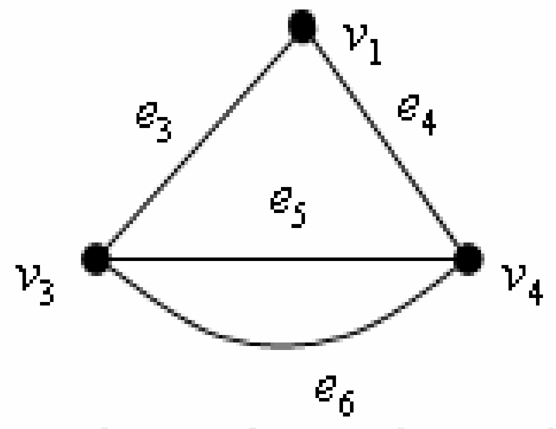
4) 若 $E'\subseteq E$，且 $E'\neq\phi$，以 E' 为边集，以 E' 的端点集为顶点集的图 G 的子图，称为 G 的由 E' 导出的边导出的子图，记为 $G[E']$.



$G[\{v_1,v_2,v_3\}]$



$G[\{e_3,e_4,e_5,e_6\}]$



3) 若 $V'\subseteq V$，且 $V'\neq\phi$，以 V'为顶点集，以两端点均在V'中的边的全体为边集的图G的子图，称为G的由V'导出的子图，记为 $G[V']$．

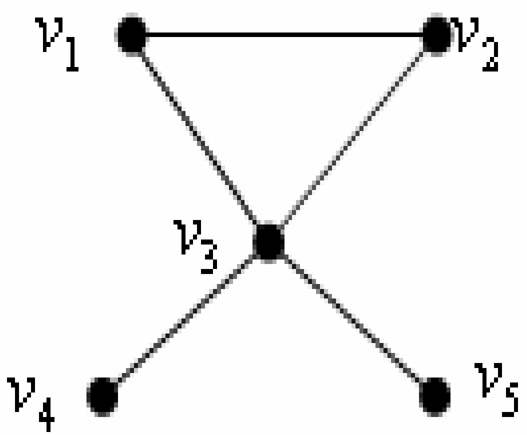
4) 若$E'\subseteq E$，且$E'\neq\phi$，以E'为边集，以E'的端点集为顶点集的图G的子图，称为G的由E'导出的边导出的子图，记为$G[E']$．

3 图的矩阵表示

邻接矩阵: **(以下均假设图为简单图).**

1) 对无向图G，其邻接矩阵 $A=(a_{ij})_{v\times v}$ ，其中：

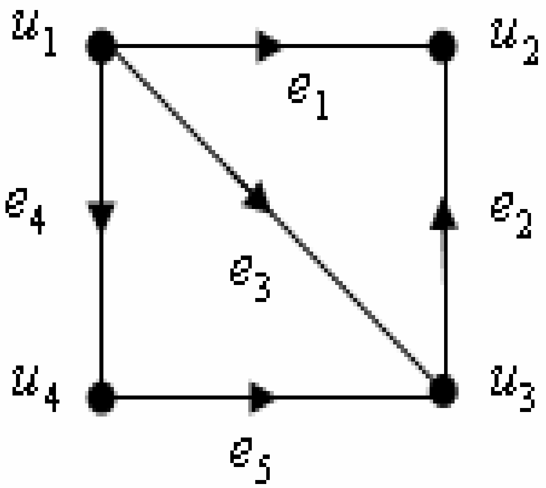
$$a_{ij}=\begin{cases}1, & 若v_i与v_j相邻,\\ 0, & 若v_i与v_j不相邻.\end{cases}$$



$$A=\begin{array}{c} \\ \\ \end{array}\begin{array}{cc} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} v_1\\ v_2\\ v_3\\ v_4\\ v_5 \end{array} \end{array}$$

2) 对有向图 $G=(V,E)$,其邻接矩阵 $A=(a_{ij})_{v\times v}$,其中:

$$a_{ij}=\begin{cases}1, & 若(v_i,v_j)\in E,\\ 0, & 若(v_i,v_j)\notin E.\end{cases}$$



$$A=\begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \end{array}$$

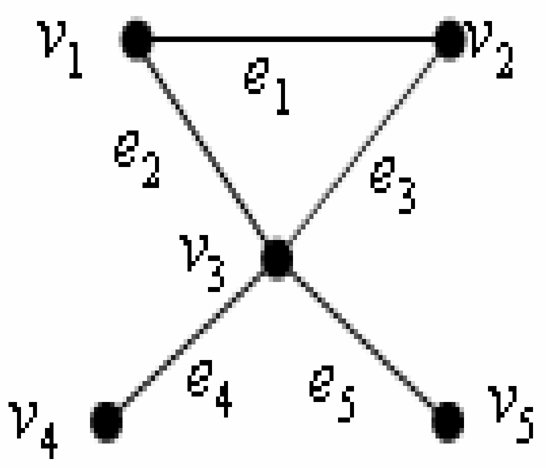
$$A=\begin{bmatrix}0&1&1&1\\0&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}\begin{array}{c}u_1\\u_2\\u_3\\u_4\end{array}$$

关联矩阵

1) 对无向图 $G=(V,E)$，其关联矩阵 $M=(m_{ij})_{\nu\times\varepsilon}$，

其中：

$$m_{ij}=\begin{cases}1, & \text{若}v_i\text{与}e_j\text{相关联},\\ 0, & \text{若}v_i\text{与}e_j\text{不关联}.\end{cases}$$

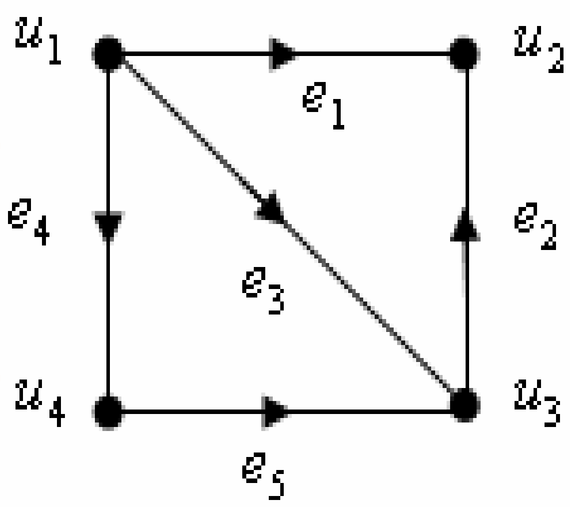


$$M=\begin{array}{c}\begin{array}{ccccc}e_1 & e_2 & e_3 & e_4 & e_5\end{array}\\ \begin{bmatrix}1 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 0 & 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{array}{c}v_1\\ v_2\\ v_3\\ v_4\\ v_5\end{array}\end{array}$$

2) 对有向图 $G=(V,E)$，其关联矩阵 $M=(m_{ij})_{\nu\times\varepsilon}$，

其中：

$$m_{ij}=\begin{cases}1, & \text{若} v_i \text{是} e_j \text{的尾}, \\ -1, & \text{若} v_i \text{是} e_j \text{的头}, \\ 0, & \text{若} v_i \text{不是} e_j \text{的头与尾}.\end{cases}$$



$$M=\begin{array}{c} \\ \\ \end{array}\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & \\ \end{array}$$

$$M=\left[\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{array}\right]\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \end{array}$$

(columns labelled e_1, e_2, e_3, e_4, e_5)

4. 顶点的度

定义 在无向图G中，与顶点v关联的边的数目(环算两次),称为顶点v的度，记为$d(v)$.

在有向图中，从顶点v引出的边的数目称为顶点v的出度，记为$d^+(v)$，从顶点v引入的边的数目称为v的入度，记为$d^-(v)$. 称$d(v)= d^+(v)+d^-(v)$为顶点v的度.

定理 $\sum_{v\in V} d(v) = 2\varepsilon.$

推论 任何图中奇点的个数为偶数.

5. 路和连通

定义 无向图G的一条途径（或通道或链）是指一个有限非空序列 $W = v_0e_1v_1e_2\ldots e_kv_k$ ，它的项交替地为顶点和边，使得对 $1 \le i \le k$，e_i的端点是v_{i-1}和 v_i，称W是从v_0到v_k 的一条途径，或一条 (v_0, v_k) 途径. 整数k称为W的长. 顶点 v_0 和 v_k 分别称为的起点和终点，而 $v_1, v_2, \ldots, v_{k-1}$称为$W$的内部顶点.

若途径W的边互不相同但顶点可重复，则称W为迹或简单链.

若途径W的顶点和边均互不相同，则称W为路或路径. 一条起点为 v_0 ,终点为v_k 的路称为 (v_0, v_k) 路记为$P(v_0, v_k)$.

起点与终点重合的途径称为闭途径.

起点与终点重合的的路称为圈(或回路)，长为*k*的圈称为*k*阶圈，记为C_k.

若在图*G*中存在(*u*,*v*)路，则称顶点*u*和*v*在图*G*中连通.

图*G*中任意两点皆连通的图称为连通图.

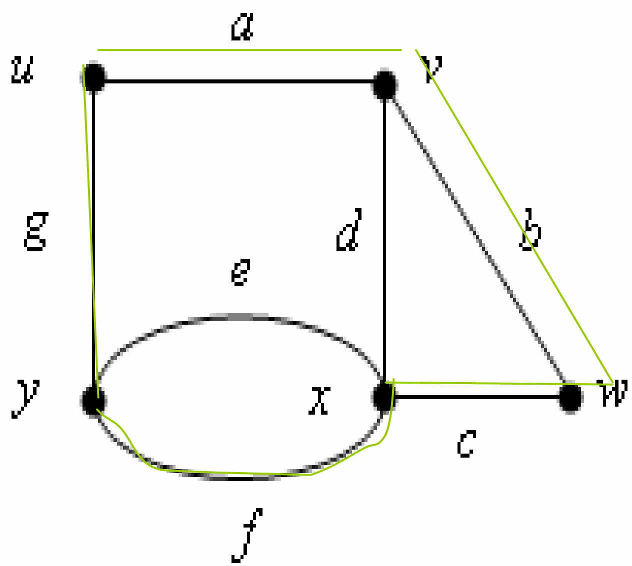
例 在右图中：

途径或链：*ugyexeyfxcw*

迹或简单链:*vbwcxdvaugy*

路或路径：*uavdxcw*

圈或回路：*uavbwcxfygu*



三、最短路问题及算法

最短路问题是图论应用的基本问题，很多实际问题，如线路的布设、运输安排、运输网络最小费用流等问题,都可通过建立最短路问题模型来求解.

- 最短路问题的两种方法：***Dijkstra***和***Floyd***算法 .

1) 求赋权图中从给定点到其余顶点的最短路.

2) 求赋权图中任意两点间的最短路.

若***P*(*u*,*v*)**是赋权图***G***中从***u***到***v***的路,称 $w(P)=\sum_{e\in E(P)} w(e)$ 称为路 ***P*(*u*,*v*)**的权.

在赋权图***G***中，从顶点***u***到顶点***v***的具有最小权的路***P***(*u*,*v*)**，称为***u***到***v***的最短路.

把赋权图中一条路的权称为它的长，把**(*u*,*v*)**路的最小权称为***u***和***v***之间的距离，并记作 ***d*(*u*,*v*).**

在赋权图中找出指定两点之间的最短路的问题称之为最短路问题。

赋权图中求给定点到各顶点的最短路的算法

(Dijkstra算法，1959年）

基本思想：若使 $(u_0,u_1,u_2,\ldots,u_{n-1},u_n)$最短, 就要使 $(u_0,u_1,u_2,\ldots,u_{n-1})$最短, 即保证从$u_0$到以后各点的路都是最短的.

令图$G=\langle V,E,W\rangle$, 集合$S_i \subseteq V$,$S_i'=V-S_i$，令$|V|=n$

$S_i=\{u|$从u_0到u的最短路已求出$\}$

$S_i'=\{u'|$从u_0到u'的最短路未求出$\}$

Dijkstra算法:(求从u_0到各点u的最短路长)

第一步. 置初值: $d(u_0,u_0)=0$ $d(u_0,v)=\infty$ (其中$v\neq u_0$) $i=0$ $S_0=\{u_0\}$ $S_0'=V-S_0$,

第二步.若 $i=n-1$ 则停. 否则转第三步

第三步. 对每个$u' \in S_i'$

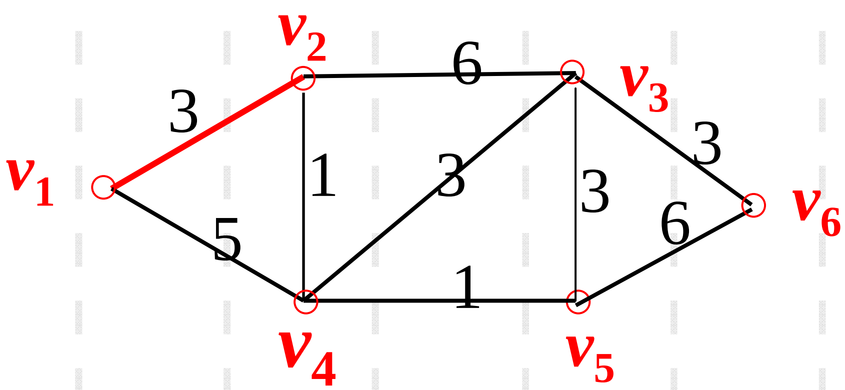
计算 $d(u_0,u')=\min_{u_i \in S_i}\{d(u_0,u'), d(u_0,u_i)+c(u_i,u')\}$

计算 $\min_{u' \in S_i'}\{d(u_0,u')\}$

并用u_{i+1}记下达到该最小值的那个结点u',

置$S_{i+1}=S_i\cup\{u_{i+1}\}$, $i=i+1$ $S_i'=V-S_i$, 转第二步.

例.求右图中从v_1到v_6的最短路。



$d(u_0,v_2)=3$, 路u_0v_2

$d(u_0,v_4)=4$, 路$u_0v_2v_4$

$d(u_0,v_5)=5$, 路$u_0v_2v_4v_5$

$d(u_0,v_3)=7$, 路 $u_0v_2v_4\ v_3$

$d(u_0,v_6)=10$, 路$u_0v_2v_4v_3v_6$

实例 一个多阶段存储问题

- 某商场欲编制某商品未来n个月的进货计划。假设每月月初进货，当月不需附加存储费，若一个月后销售则需加存储费。设每月需求量、进货单价和存储单价如下表：

月	1	2	…	n
需求量	b_1	b_2	…	b_n
单价	c_1	c_2	…	c_n

存储月份	1~2	2~3	…	n-1~n
存储单价	d_1	d_2	…	d_{n-1}

确定一个进货与存储的合理水平使总费用最少。

- 分析： 第k月的需要量可在第1月，第2月，…第k月的任何一个月进货，分别计算费用如下：

第1月进货总费用：$a_1=b_k(c_1+d_1+ \ldots+d_{k-1})$

第2月进货总费用：$a_2=b_k(c_2+d_2+ \ldots+d_{k-1})$

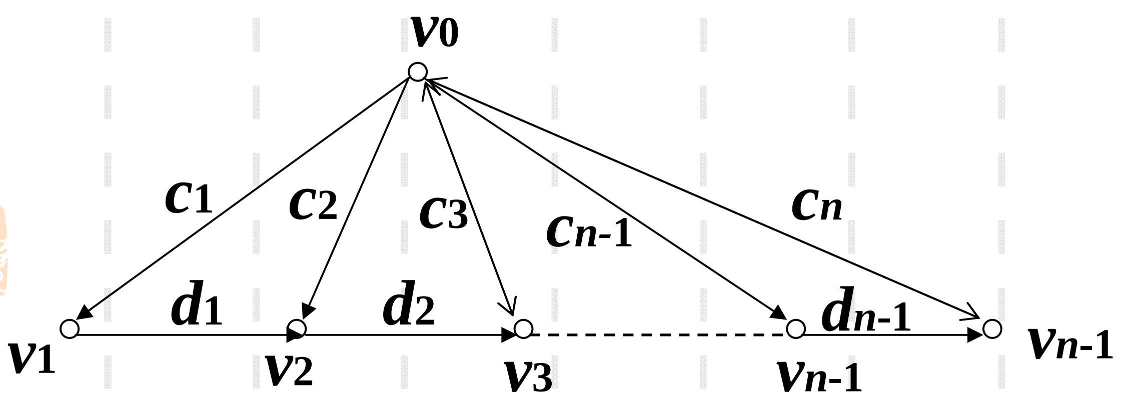
… …

第k月进货总费用：$a_k=b_k d_k$

比较$a_1,a_2, \ldots,a_k$的大小即可确定哪一个月进货总费用更省。

在确定借贷与存储的合理水平时，仅考虑一个单位货物，而不考虑进货量。

模拟图：以顶点v_i表示第i月，i=1,2,…,n,连结点v_i,v_{i+1},得边v_iv_{i+1}，给边v_iv_{i+1}赋权d_i，i=1,2, … ，n-1；附加顶点v_0，连结顶点v_0,v_i，得边 v_0v_i, 给边 v_0v_i 赋权 c_i，i=1,2, …,n，得到n+1个顶点的赋权有向图，如下：



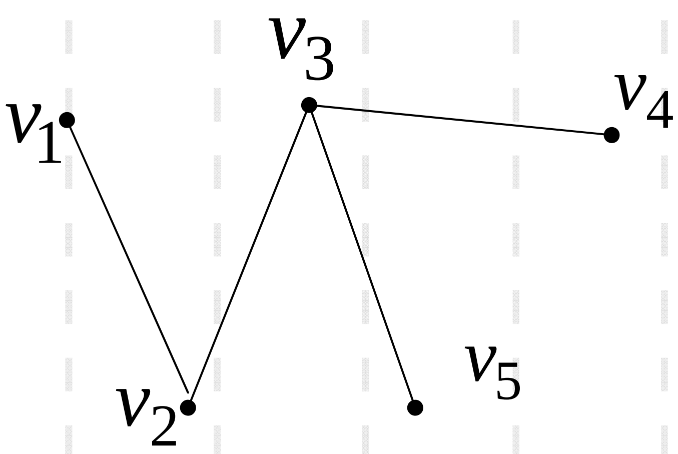
问题：将边的权理解为距离，费用最小就是求距离最短。因此只要求出顶点v_0到其余各点的最短路，即可确定每个月的需要量该由哪个月进货最好，从而制定出总费用最少的n个月的进货与存储计划。

四、最小生成树及算法

1. 树的定义

连通且不含圈的无向图称为树．常用T表示.

树中的边称为树枝. 树中度为1的顶点称为树叶.



2. 树的等价定义

设G是具有n个顶点的图，则下述命题等价

1) G是树（ $\Leftrightarrow$ G无圈且连通）；

2) G无圈，且有n-1条边;

3) G连通，且有n-1条边;

4) G无圈，但添加任一条新边恰好产生一个圈;

5) G连通，且删去一条边就不连通了（即G为最小连通图）；

6) G中任意两顶点间有唯一一条路.

3. 生成树

定义 若*T*是包含图*G*的全部顶点的子图,它又是树,则称*T*是*G*的生成树. 图*G*中不在生成树的边叫做弦.

图*G*=(*V*,*E*)有生成树的充要条件是图*G*是连通的.

找生成树的方法：避圈法和破圈法.

4. 赋权图的最小生成树

一棵生成树中的所有边的权之和称为该生成树的权. 具有最小权的生成树,称为最小生成树.

最小生成树很有实际应用价值.

例如 顶点是城市,边的权表示两个城市间的距离从一个城市出发走遍各个城市,如何选择最优的旅行路线.

城市间的通信网络问题,如何布线,使得总的线路长度最短.

求最小生成树一般有两种方法:

*Kruskal*算法（或避圈法）和破圈法.

A. Kruskal算法（或避圈法）

步骤如下：

(1) 选择边$\boldsymbol{e_1}$，使得$\boldsymbol{w(e_1)}$尽可能小；

(2) 若已选定边$e_1, e_2, \ldots, e_i$，则从$E\setminus\{e_1, e_2, \ldots, e_i\}$中选取$e_{i+1}$，使得:

i) $G[\{e_1, e_2, \ldots, e_{i+1}\}]$为无圈图，

ii) $w(e_{i+1})$是满足**i)**的尽可能小的权，

(3) 当第**(2)**步不能继续执行时，则停止.

B. 破圈法

算法2 步骤如下：

(1) 从图G中任选一棵树T_1.

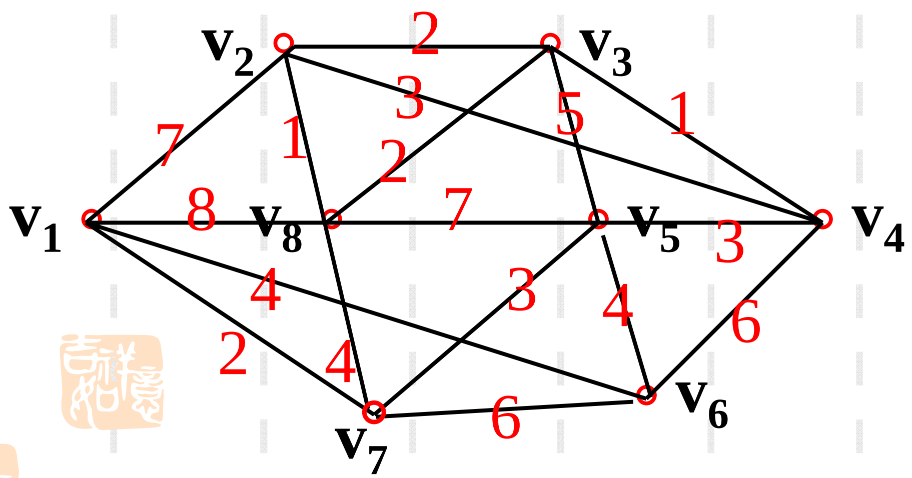
(2) 加上一条弦e_1，T_1+e_1中

生成一个圈. 去掉此圈中最大权边，得到新树T_2，以T_2代T_1,重复(2)再检查剩余的弦，直到全部弦检查完毕为止.

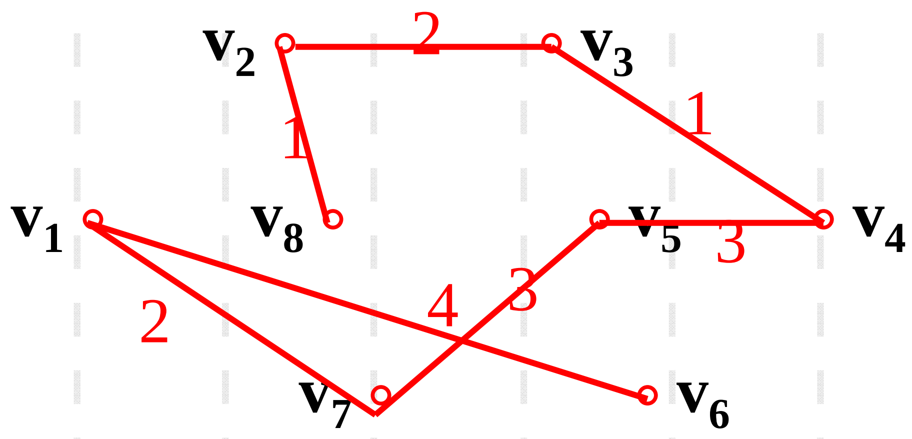
例 n个城市,各城市之间的距离如下表(距离为∞，表示两个城市之间没有直接到达的线路)。从一个城市出发走遍各个城市,如何选择最优的旅行路线.

$$\begin{pmatrix} 0 & 7 & \infty & \infty & \infty & 4 & 2 & 8 \\ 7 & 0 & 2 & 3 & \infty & \infty & \infty & 1 \\ \infty & 2 & 0 & 1 & 5 & \infty & \infty & 2 \\ \infty & 3 & 1 & 0 & 3 & 6 & \infty & \infty \\ \infty & \infty & 5 & 3 & 0 & 4 & 3 & 7 \\ 4 & \infty & \infty & 6 & 5 & 0 & 6 & \infty \\ 2 & \infty & \infty & \infty & \infty & 4 & 0 & 4 \\ 8 & 1 & 2 & \infty & 7 & \infty & 4 & 0 \end{pmatrix}$$

城市作为顶点，两个城市之间有直达的线路，则连边，且给边赋权距离，得一个赋权图。



问题就是求一棵最小生成树。



五、旅行售货员问题

一个旅行售货员想去访问若干城镇，然后回到出发地.给定各城镇之间的距离后，应怎样计划旅行路线，使他能对每个城镇恰好经过一次而总距离最小？

它可归结为这样的图论问题：在一个赋权完全图中,找出一个**最小权的H圈**.

但这个问题是***NP-hard***问题，即不存在多项式时间算法.也就是说,对于大型网络(赋权图),目前还没有一个求解旅行售货员问题的有效算法，因此只能找一种求出相当好（不一定最优）的解.

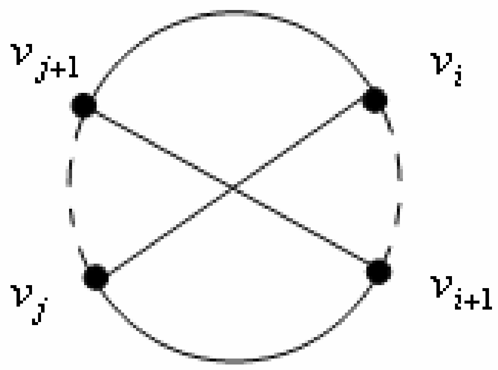
定义设$G=(V,E)$是连通无向图，包含图G的每个顶点的路称为G的哈密尔顿路**(Hamilton路或H路).**

包含图G的每个顶点的圈，称为G的哈密尔顿圈(或**Hamilton圈或H圈**).

含**Hamilton**圈的图称为哈密尔顿图。

一个可行的方法：

先求一个***H***圈,再适当修改,得到具有较小权的另一***H***圈.



设找到一个初始的 ***H*** 圈 $C = v_1 v_2 \ldots v_n v_1$，对所有适合 $1 < i+1 < j < v$ 的 i 和 j，总可得到一个新的 ***H*** 圈（C 中去掉边 $v_i v_{i+1}$ 和 $v_j v_{j+1}$，再加上边 $v_i v_j$ 和 $v_{i+1} v_{j+1}$）：$C_{ij} = v_1 v_2 \ldots v_i v_j v_{j+1} \ldots v_{i+1} v_{j+1} v_{j+2} \ldots v_n v_1$，如图所示。

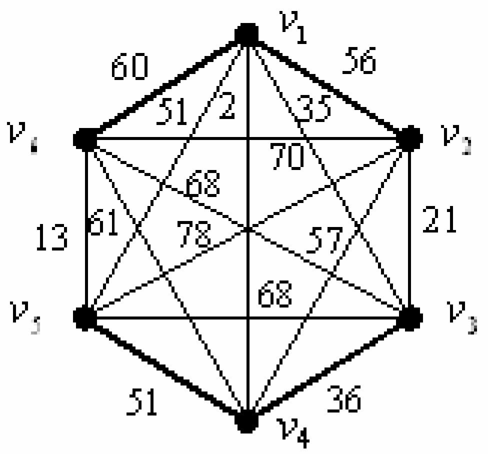
定义 若对于某一对$\boldsymbol{i}$和$\boldsymbol{j}$，有

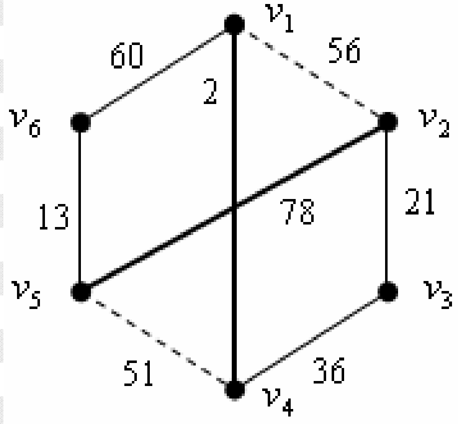
$$w(v_i v_j) + w(v_{i+1} v_{j+1}) < w(v_i v_{i+1}) + w(v_j v_{j+1})$$

则圈$\boldsymbol{C_{ij}}$将是圈$\boldsymbol{C}$的一个改进.

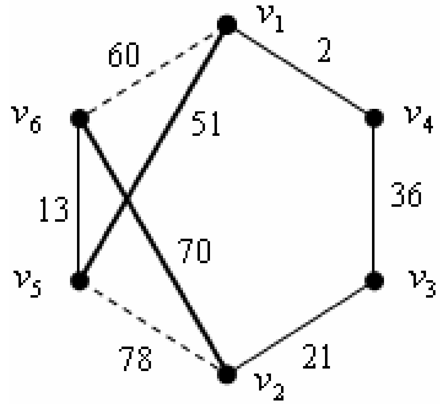
在进行一系列改进之后, 最后得一个圈,不能再用此方法改进了，这个最后的圈可能不是最优的,但是它是比较好的，为了得到更高的精度，这个过程可以重复几次，每次以不同的圈开始. 这种方法叫做二边逐次修正法.

例 对下图的K_6, 用二边逐次修正法求较优H圈.

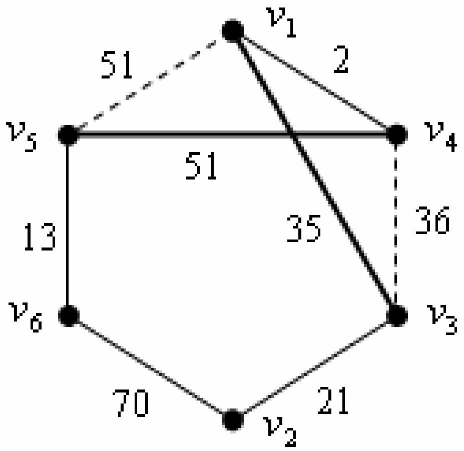




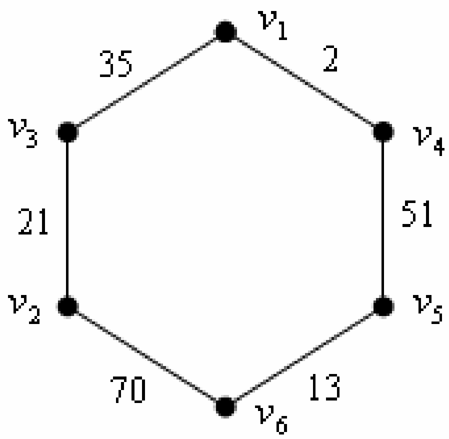
(a)



(b)



(c)



(d)

较优H圈:

$C_3 = v_1v_4v_5v_6v_2v_3v_1$

其权为$W(C_3)$=192

六、网络与网络流

1. 网络流的基本概念

先来看一个实例。

现在想将一些物资从*S*运抵*T*，必须经过一些中转站。连接中转站的是公路，每条公路都有最大运载量。

每条弧代表一条公路，弧上的数表示该公路的最大运载量。最多能将多少货物从*S*运抵*T*?

定义6.1 若有向图满足下列条件：

(1)有且仅有一个入度为零的顶点s，称为源点；

(2) 有且仅有一个出度为零的顶点t，称为汇点；

(3) 每一条弧(v_i, v_j)都有一个非负数c_{ij}，称为该边的容量。如果v_i,v_j之间没有边，$c_{ij}=0$。

则称该有向图为网络，记为$N = (V, E, C)$.

图6.1所给出的一个赋权有向图N就是一个网络，指定v_1是源点，v_4为汇点，弧旁的数字为c_{ij}。

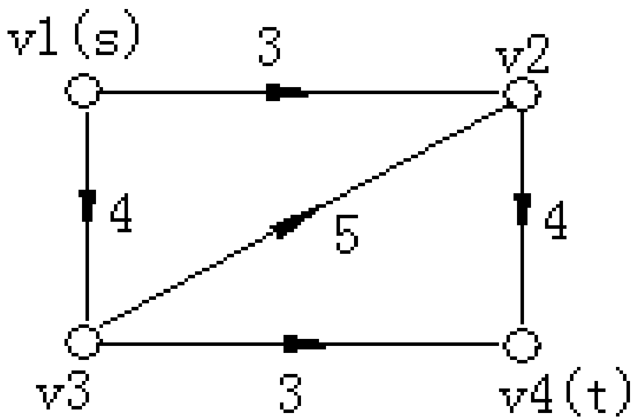


图6.1

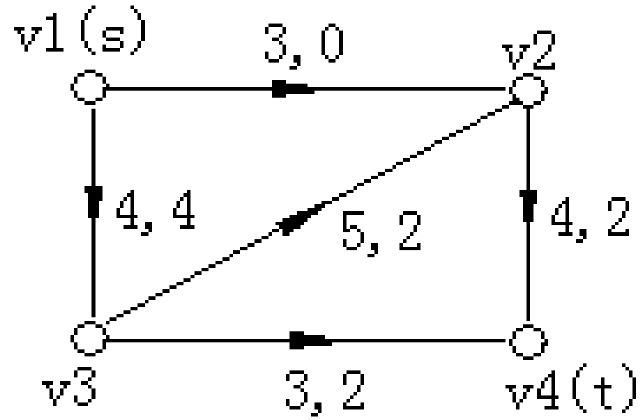


图6.2

网络流：是定义在弧集合E上一个函数$f=\{f(v_i,v_j)\}$，并称$f(v_i,v_j)$为弧(v_i,v_j)上的流量(简记为f_{ij})。如图6.2所示的网络N，弧上两个数，第一个数表示容量c_{ij}，第二个数表示流量f_{ij}。

2.可行流与最大流

(1) 定义

在实际问题中，对于流有两个显然的要求：一是每个弧上的流量不能超过该弧的最大通过能力(即弧的容量)；二是中间点的流量为0，源点的净流出量和汇点的净流入量必相等。因此有定义如下。

定义6.2 网络$N = (V, E, C)$中如果每条边都给定一个非负实数f_{ij}满足下列条件

(1)容量约束：$0 \leqslant f_{ij} \leqslant c_{ij}$，$(v_i,v_j) \in E$，

(2)守恒条件

对于中间点：流入量=流出量，即

$$\sum_j f_{ij} = \sum_j f_{ji}, \quad i \neq s,t$$

对于源点与汇 点：源点的净流出量=汇点的净流入量，即

$$\sum_j f_{sj} = \sum_j f_{jt} = w$$

那么这一组f_{ij}称为网络N上的可行流，w称为流量.

网络N中流值最大的流f^*称为N的最大流.

(2) 可增广(流)路径

可增广路径，是指这条路径上的流可以修改，通过修改，使得整个网络的流值增大。

定义6.3 设f是一个可行流，P是从源点s到汇点t的一条路，若P满足下列条件：

(1)在P上的所有前向弧$(v_i \to v_j)$都是非饱和弧，即$0 \leqslant f_{ij} < c_{ij}$;

(2)在P上的所有后向弧$(v_i \leftarrow v_j)$都是非零弧，即$0 < f_{ij} \leqslant c_{ij}$。则称$P$为(关于可行流$f$的)一条可增广路径。

(3) 割及其容量

定义6.4 如果S是V的一个子集， $\bar{S} = V - S$

$s \in S, t \in \bar{S}$，则称边集$(S, \bar{S})$ 为网络N的一个割。显然，若把某一割的弧从网络中去掉，则从s到t就不存在路。所以直观上讲，割是从s到t的必经之道。

定义5　给一割 $(S,\bar{S})$ ，把其中所有弧的容量之和称为这个割的容量，记为 $c(S,\bar{S})$ ，即

$$c(S,\bar{S})=\sum_{e\in E}c(e)$$

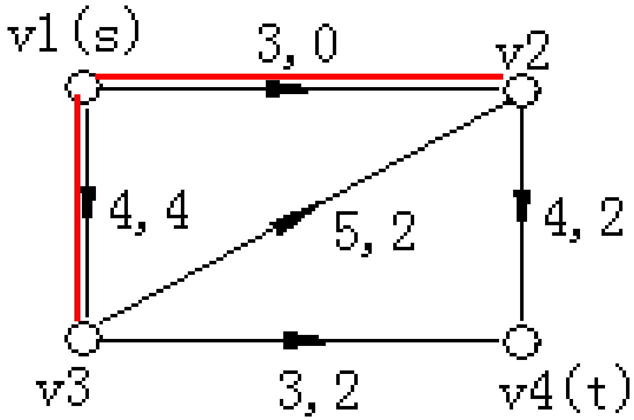
网络N中容量最小的割 $(S^*,\bar{S}^*)$ 称为N的最小割。

不难证明，任何一个可行流的流量w都不会超过任一割的容量，即

$$w\le c(S,\bar{S})$$

例如，图6.2中，若 $S=\{s\},(S,\bar{S})=\{(s,v_3),(s,v_2)\}$

$c(S,\bar{S})=4+3=7.$



定理1 网络的最大流量不超过最小的割的容量，即

$$\max w \le \min c(S,\bar{S})$$

证明 设f是给定网络的任意可行流，由可行流的性质

$$\sum_j f_{sj} = w$$

$$\sum_j (f_{ij} - f_{ji}) = 0. \quad i \ne s,t, j \in V$$

任给一个割 $(S,\bar{S})$

$$\sum_{i\in S, j\in V} (f_{ij} - f_{ji}) = w.$$

即

$$\sum_{i\in S, j\in S} (f_{ij} - f_{ji}) + \sum_{i\in S, j\in \bar{S}} (f_{ij} - f_{ji}) = w.$$

因 $$\sum_{i\in S,j\in S}(f_{ij}-f_{ji})=0,$$

所以 $$\sum_{i\in S,j\in \bar{S}}(f_{ij}-f_{ji})=w.$$

由于 $$0\le f_{ij}\le c_{ij},\quad f_{ij}-f_{ji}\le f_{ij}$$

所以

$$w=\sum_{i\in S,j\in \bar{S}}(f_{ij}-f_{ji})\le \sum_{i\in S,j\in \bar{S}}f_{ij}\le \sum_{i\in S,j\in \bar{S}}c_{ij}=c(S,\bar{S})$$

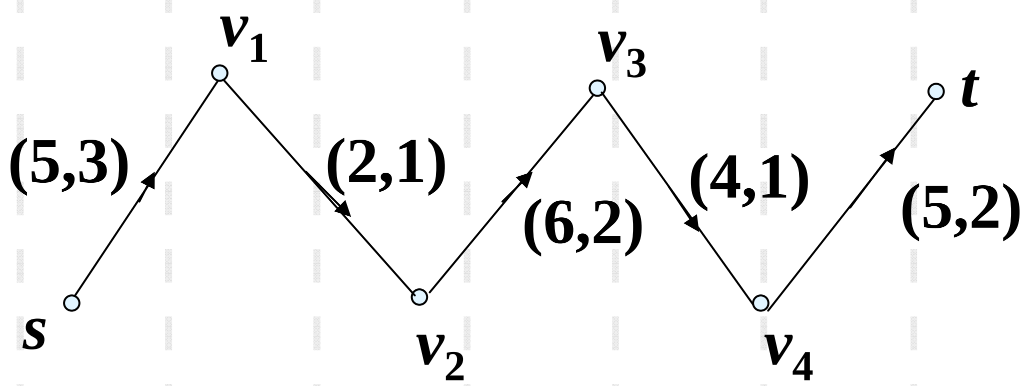
由于可行流和割的任意性，定理成立。

如果网络的可行流不是最大流，就一定存在从s到t的可增流路径。

令s,v_1,v_2,…,v_k,t是一条s到t的路径P_{st}，其中每条边的方向都是v_j到v_{j+1},称为向前边。如果这条路径上每条边e_{ij}都有$f_{ij}<c_{ij}$，那么令

$$\delta = \min_{e_{ij} \in P_{st}} (c_{ij} - f_{ij})$$

令P_{st}每条边的流都增加δ，所得流分布仍然是网络的可行流分布，但流增加了δ.



δ=1

图6.3 网络中的一部分

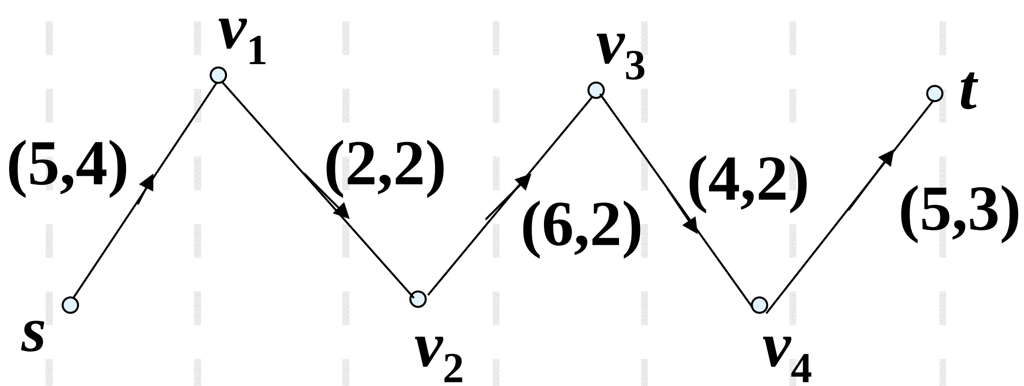


图 6.4

还可以包含向后的可增流路径P_{st}，要求向前边e_{ij}都有$f_{ij}<c_{ij}$，后向边e_{ij}满足$f_{ij}>0$,对前向边e_{ij}

$$\delta_1 = \min_{e_{ij}\in P_{st}}(c_{ij} - f_{ij})$$

后向边e_{ji},

$$\delta_2 = \min_{e_{ji}\in P_{st}}(f_{ji})$$

$$\delta = \min\{\delta_1, \delta_2\}$$

图6.5，δ=1,

图6.5，δ=1,

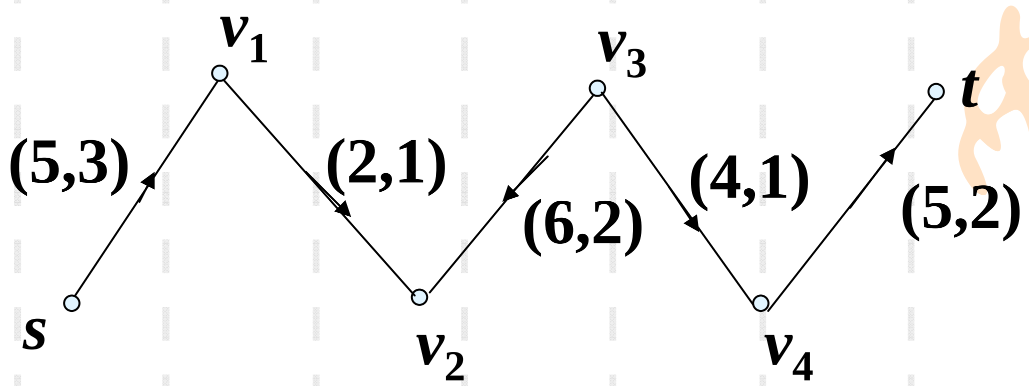
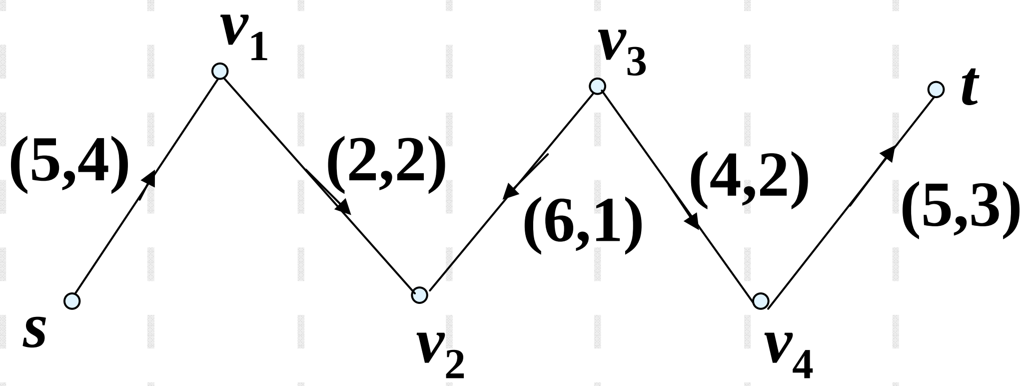


图6. 5，δ=1,



第1条可增路s,c,b,t, δ=2

第2条可增路s,a,b,c,d,t,

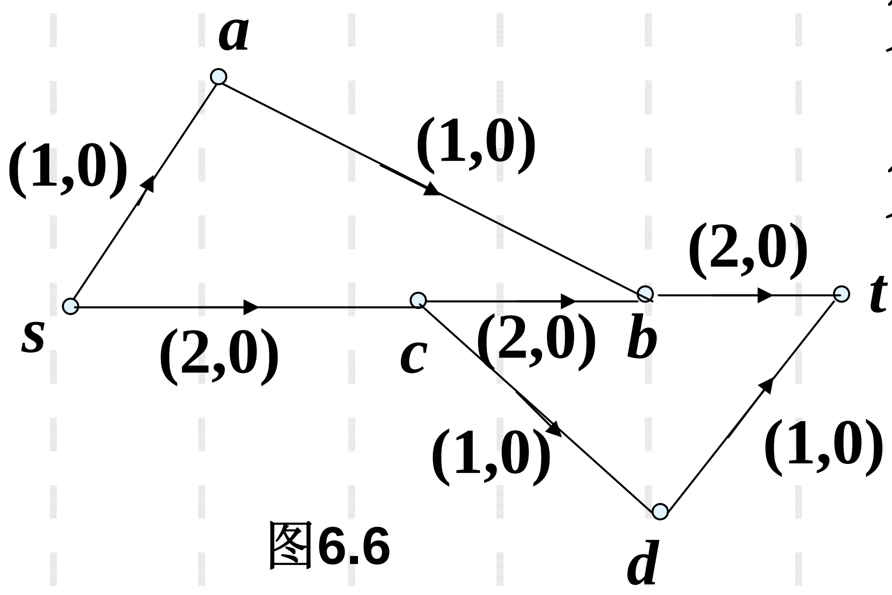


图6.6

最大流w=3

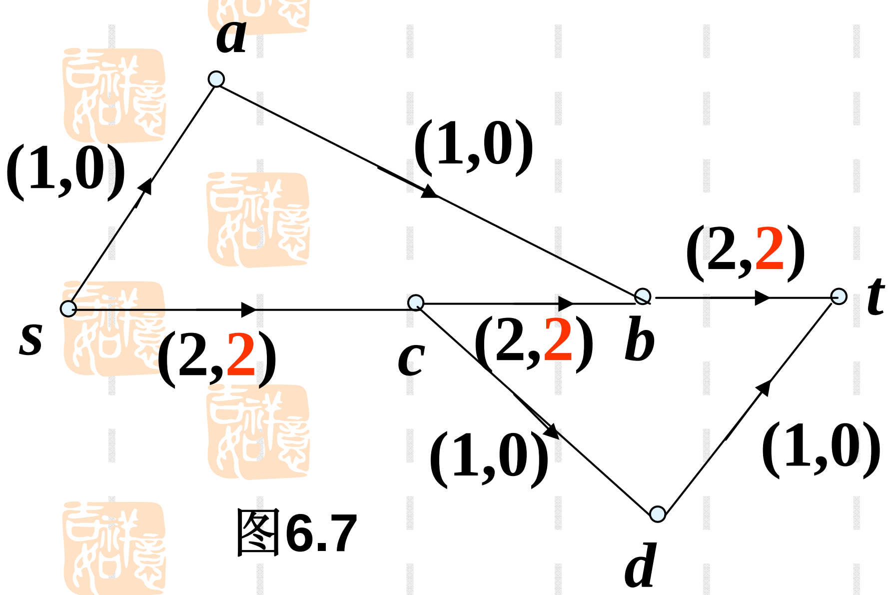


图6.7

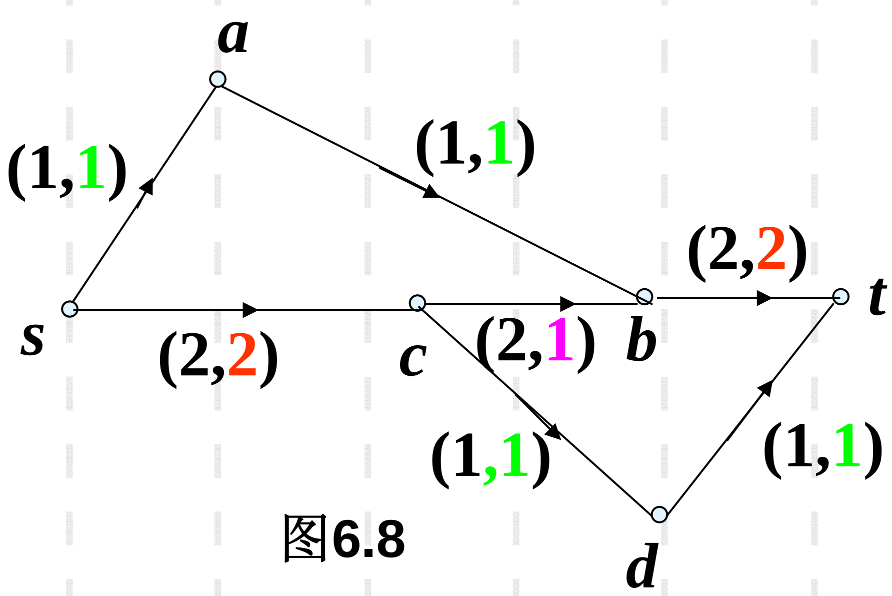


图6.8

定理2 最大流最小割定理：在一个网络N中，最大流量等于最小割的容量。

证明　设网络的一个可行流f为最大流，确定一个割如下：

$s \in S$

若 $x \in S,(x,y)$ 是向前边且 $f_{xy} < c_{xy}$ ，则 $y \in S.$

若 $x \in S,(y,x)$ 是后前边且 $f_{yx} > 0$ ，则 $y \in S.$

则 $t \notin S$ ，否则存在s到t的一条可增路，矛盾。

因此，$\bar{S} \neq \varnothing$ ，则任意 $x \in S, y \in \bar{S}$ 的边(x,y)有

若 (x,y)是向前边，$f_{xy} = c_{xy}$；(y,x) 是后前边，

$$f_{yx} = 0$$

由定理1， $\max w \le \min c(S, \bar{S})$

又 $w = \sum_{x \in S, y \in \bar{S}} (f_{xy} - f_{yx}) = \sum_{x \in S, y \in \bar{S}} c_{ji} = c(S, \bar{S}),$

所以 $\max w = \min c(S, \bar{S})$

最大网络流

最大流问题实际上是求一可行流$\{f_{ij}\}$，使得w达到最大。若给了一个可行流f，只要判断N中有无关于f的增广路径，如果有增广路径，改进f，得到一个流量增大的新的可行流；如果没有增广路径，则得到最大流。

设$(S^*, \bar{S}^*)$是最小割，下面用顶点标号法来定义S^*，在标号过程中，有标号的顶点表示是S^*中的点，没有标号的点表示不是S^*中的点。如 果t有标号，则说明找到了一条增广路；如果标号过程进行不下去，而t没有标号，则说明不存在增广路，于是得到了最大流，同时也得到了一个最小割集。

求最大流的标号法(**Ford,Fulkerson**)

从一个可行流(一般取零流)开始，不断进行以下的标号过程与增广过程，直到找不到关于f的可增广路径为止。

1. 标号过程**A**

标记过程中每个结点给予**3**个标号，第一个标号表示该点的先驱点，第二个标号为“+”或“-”，表示先驱点与该点连接的边在可增广路中是前向边还是反向边，第三个标号表示这条边上能增加或减少的流值。

stepA1 发点s标记为(s，+, ∞)，此时称为已标记，未检查，其余点均称为未标记，未检查。

stepA2 任选一已标记未检查的结点x，若结点y与x邻接且为标记，则当

(1) 若$(x,y)\in E$且$c_{xy}>f_{xy}$时，则y标记为$(x ,+,\delta_y)$，其中 $\delta_y= \min\{\delta_x, c_{xy}-f_{xy}\}$

之后，称y已标记未检查。

(2) 若$(y, x)\in E$且$f_{yx}>0$时，则y标记为$(x ,-,\delta_y)$，其中

$$\delta_y= \min\{\delta_x, c_{xy}-f_{xy}\}$$

之后，称y已标记未检查。

(3) 与结点**x**邻接的所有结点都标记完之后，将**x**的标记的符号**“+”**或**“-”**加以标记，表示**x**已标记且已检查。

StepA3 重复**stepA2**，直到收点***t***被标记，或者收点不能获得标记为止。如果是前者，转向增广过程，如果是后者，算法结束，所得流即是最大流。

2. 增广过程B

stepB1 令 z=t

stepB2 若z的标记为(q,+,δ_z)，则

$$f_{qz} \leftarrow f_{qz} + \delta_z$$

若z的标记为(q,-,δ_z)，则

$$f_{qz} \leftarrow f_{qz} - \delta_z$$

stepB3 若 q=s，则把全部标记去掉， 转向标记过程A, 否则令z=q，转到B2.

例 求图6.9所示的最大流。边上的数字表示容量。

设f是任意可行流，我们从零流开始，设每条边的流均为0，即

$$f_{uv}=0,\ \ (u,v)\in E$$

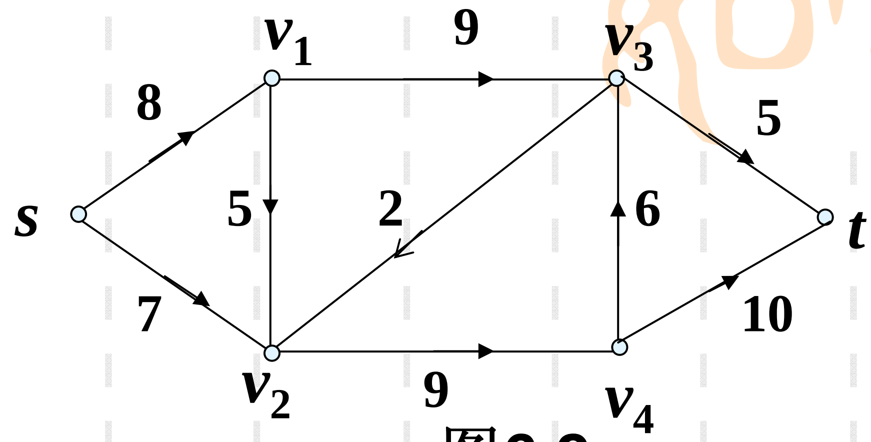


图6.9

1. 找一条可增广路并增加其流值

A (标记过程)

(1)发点s标记为(s，+, ∞)

(2) 考察与s邻接的点v_1和v_2。对点v_1，

$$(s,v_1)\in E\quad \text{and}\quad c_{sv1}>f_{sv1},$$

则

$$\delta_{v1} = \min\{\delta_s, c_{sv1} - f_{sv1}\} = 8$$

于是，**v_1**标记为（**s,+,8**）

同样的方法**v_2**得到标记（**s,+,7**）。

与**s**邻接的点都已标记，**s**标记中的“+”写成⊕，表示**s**已标记，已检查，如图**6.10**。

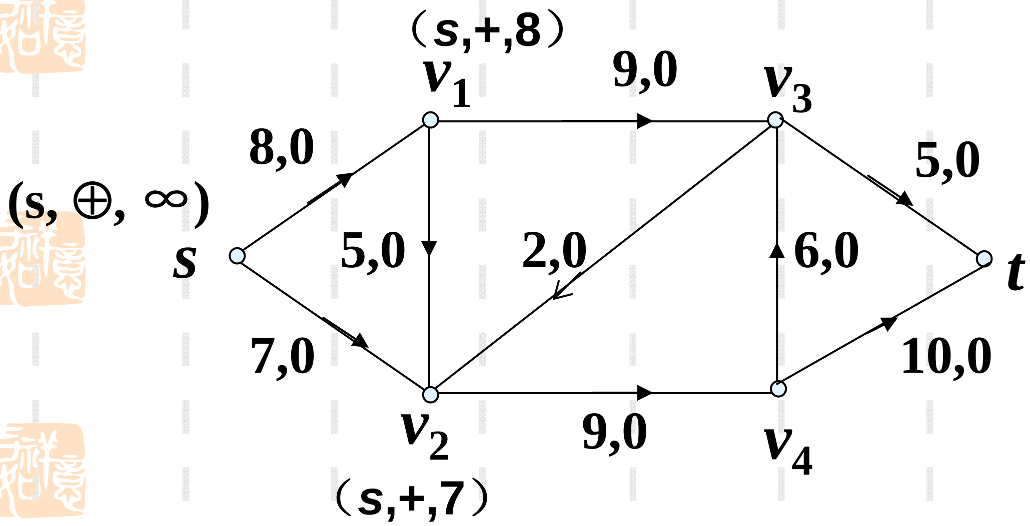


图**6.10**

(3) 重复**stepA2,**选一已标记、未检查的点，如选**v_1**点，与**v_1**邻接且未标记的点只有**v_3**。因

$$(v_1, v_3) \in E \quad \text{and} \quad c_{v1v3} > f_{v1v3},$$

则 $$\delta_{v3} = \min\{\delta_{v1}, c_{v1v3} - f_{v1v3}\} = 8$$

v_3标记为**(v_1,+,8)**。

点**v1**已标记、一检查，将其标记中的“+”写成⊕，如图**6.11**。

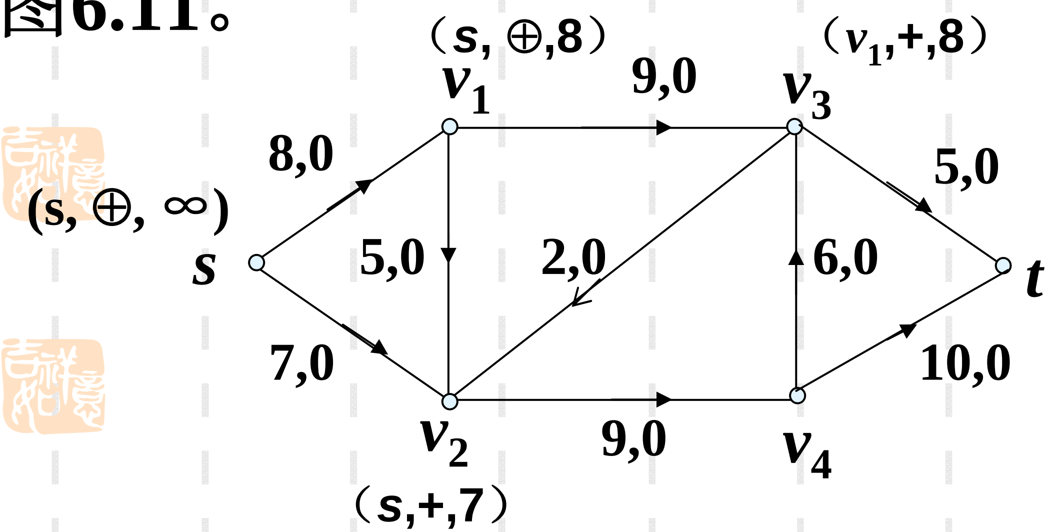


图6.11

(4) 重复stepA2,选一已标记、未检查的点，如选 v_3点，与v_3邻接且未标记的点有v_4和t。

对于点v_4，因（v_4,v_3）$\in E$ 且f_{v4v3}=0，因此，不能用点v_3去标记点v_4.对于点t，因

$$(v_3,t)\in E \quad \text{and} \;\; c_{v3t} > f_{v3t},$$

$$\delta_t = \min\{\delta_{v3}, c_{v3t} - f_{v3t}\} = 5$$

t标记为(v_3,+,5)。
如图6.12。
由于t已标记，
转到增广过程B.

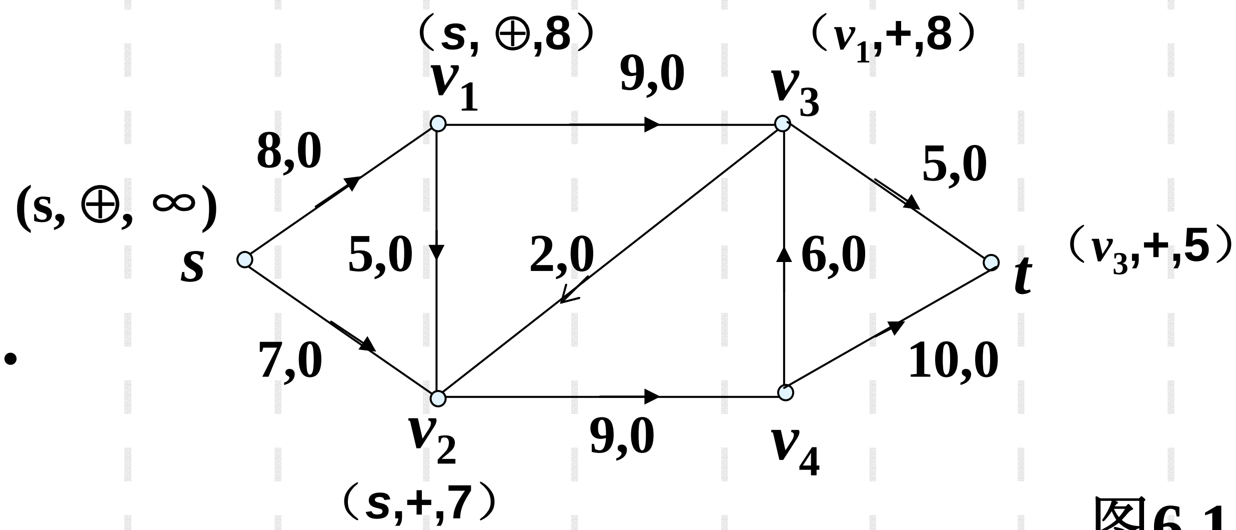


图6.12

B. 增广过程

$$f_{v3t} \leftarrow 0+5=5,$$

$$f_{v1v3} \leftarrow 0+5=5,$$

$$f_{sv1} \leftarrow 0+5=5$$

至此，完成增广过程。如图6.13。

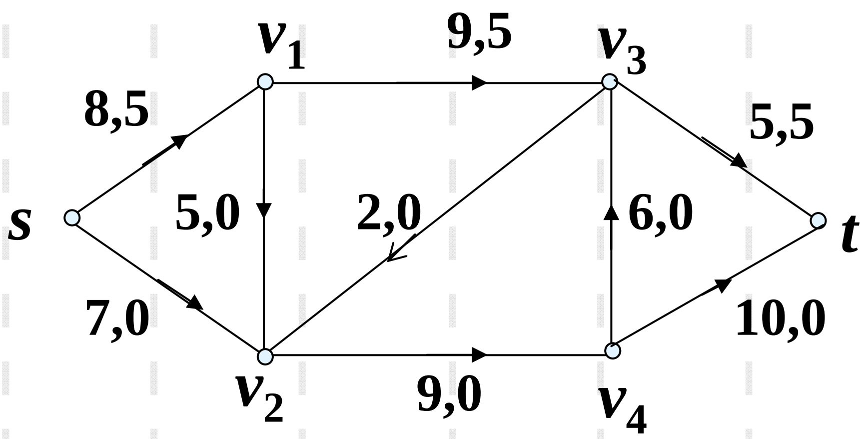


图6.13

2.找一条可增广路并增加其流值

A (标记过程)

对图13重新标记得到图6.14。

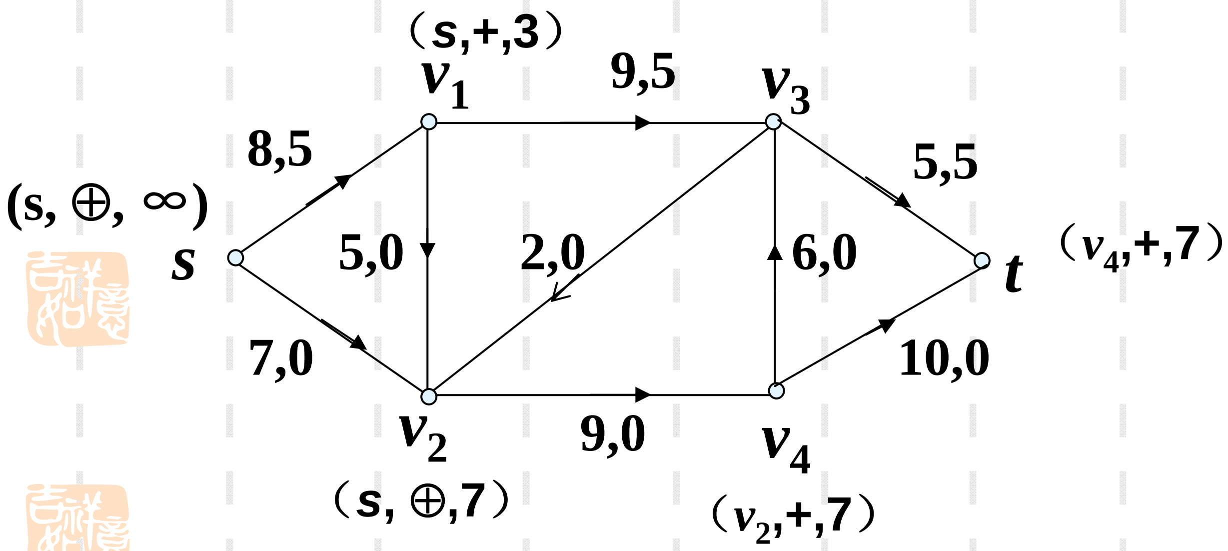


图6.14

B. 增广过程。从标记过程得到一条可增广路：

$$s \rightarrow v_2 \rightarrow v_4 \rightarrow t$$

增值δ=7,于是得到图6.15，至此又完成一次增广过程。

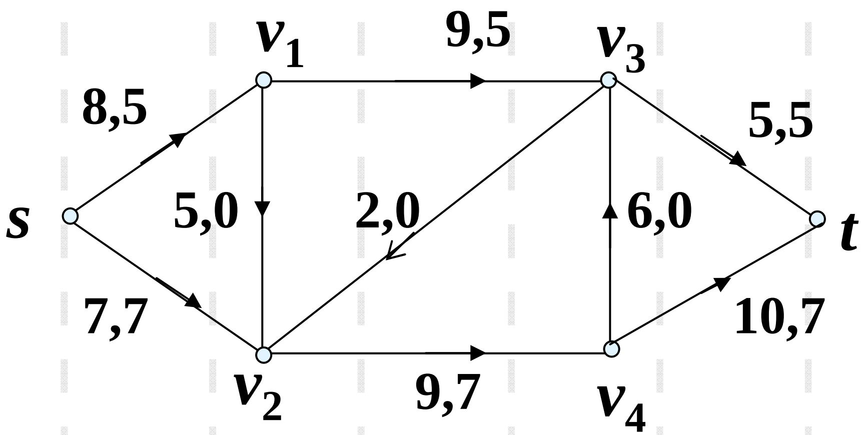


图6.15

3.找一条可增广路并增加其流值

对图6.15重新标记得到图6.16，得到一条可增广路：

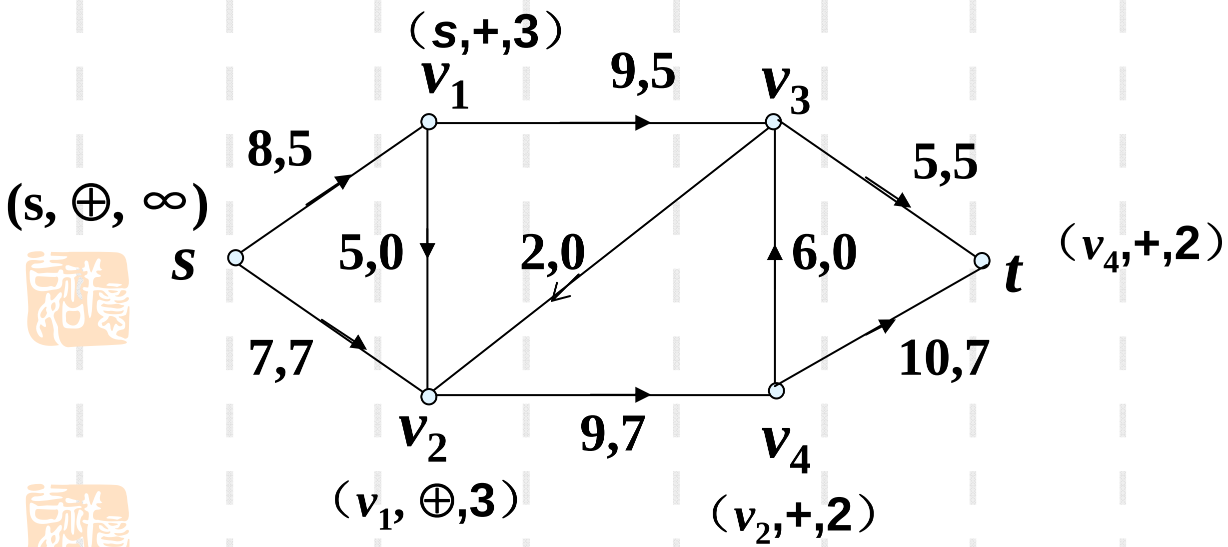


图6.16

$$s \to v_1 \to v_2 \to v_4 \to t$$

增值δ=2,于是得到图6.17。

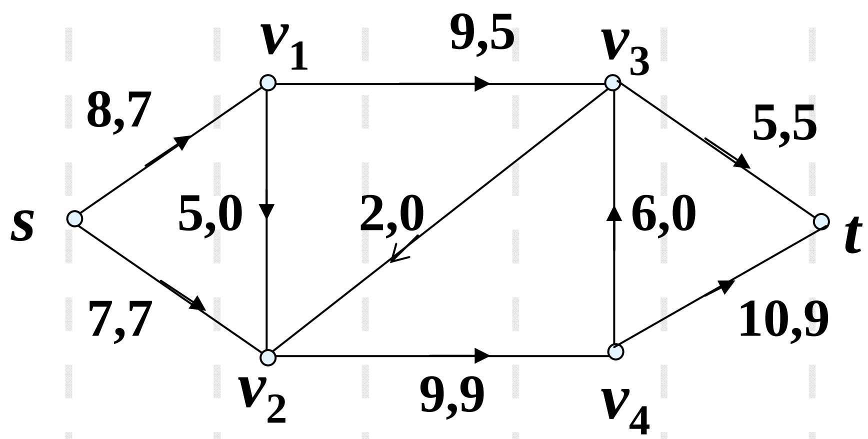


图6.17

4. 对图6.17重新标记得到图18，v4和t均不能再获标记，算法结束。最大流为14。

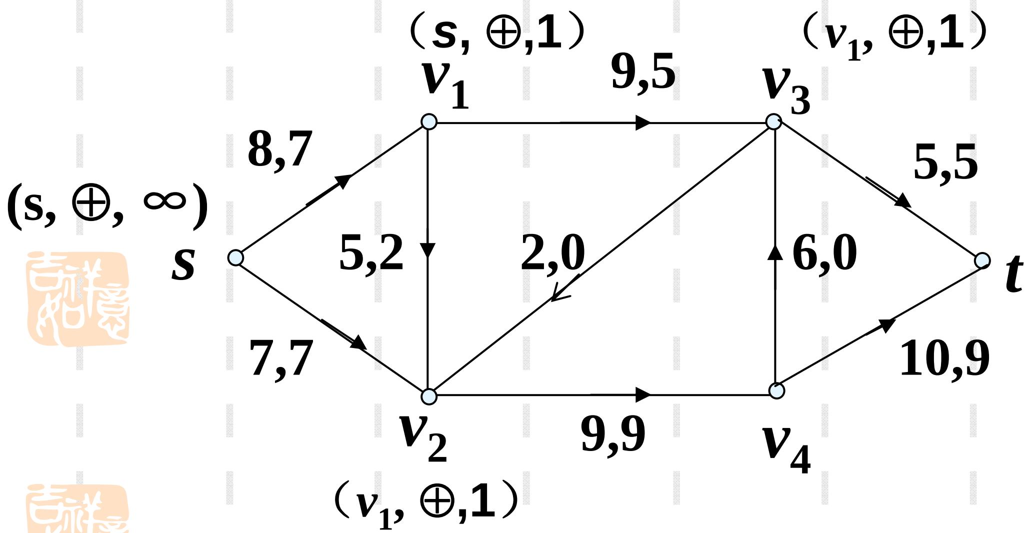


图6.18

将获得标记的结点归为S,不能标记的结点归为 $\bar{S}$

即 $S=\{s,v_1,v_2,v_3\}$,$\bar{S}=\{v_4,t\}$

得到最小割为：

$$(S,\bar{S})=\{(v_2,v_4),(v_3,t)\}$$

其容量为 $C(S,\bar{S})=9+5=14$

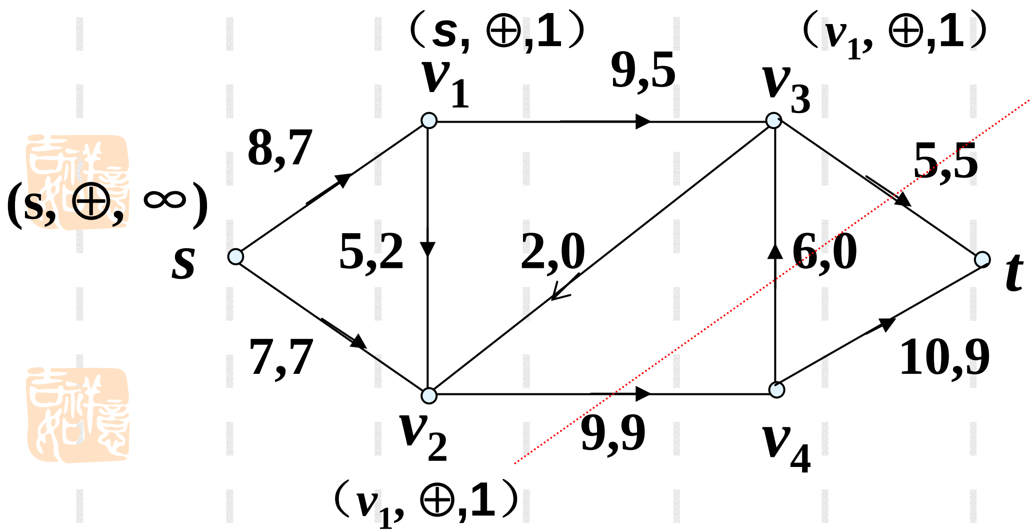


图6.18

七、匹配问题

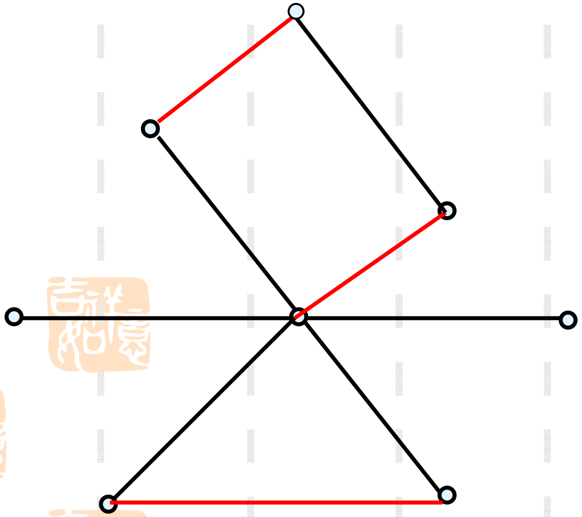
- 问题引出

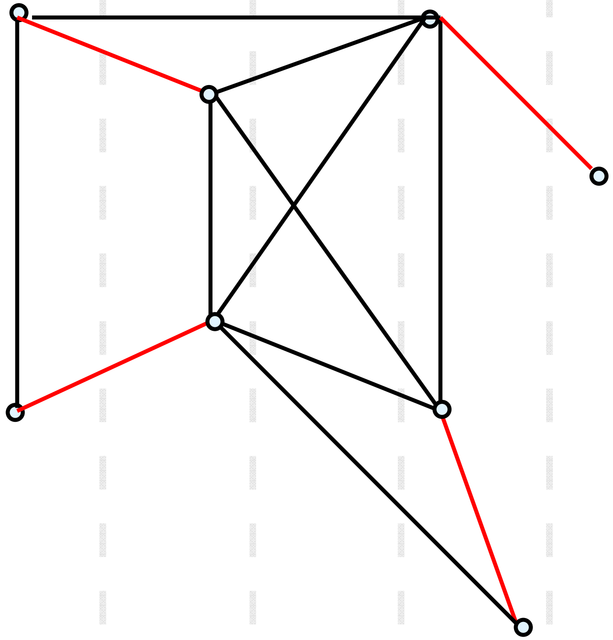
  *m*家公司到某校研究生院招聘经理，每家公司只招收一名，只要是该校毕业生他们都满意。但毕业生每个人心目中有自己可以接受公司的一个清单，不是任何公司他们都会应聘。设有*n*名毕业生，问是否每位毕业生都可能得到他可以接受的岗位？如果不可能，最多可能有多少位毕业生满意？

1. 匹配

设M是E的一个子集，它的元素都是G的边，并且这些边中的任意两个在G 中均不相邻，则称M为G的匹配(对集match)。

M中一条边的两个端点称为在M下是配对的。若匹配M的某条边与顶点v关联，则称M 饱和顶点v，并且称v是M饱和的，否则，称v是M非饱和的。若G的每个顶点均为M饱和的，则称G为M的完美匹配。若G没有另外的匹配M_1,使得$|M_1|>|M|$,则M成为G的最大匹配；显然，每个完美匹配都是最大匹配。





- 设M是G的匹配，G的M交错路是指边在E\M和M中交替出现的路。
- M可扩路是指其起点和终点都是M非饱和的M交错路。
- 定理1 G的匹配M是最大匹配当且仅当G不包含M可扩路。

证明 设M是G的匹配，并假设G包含M可扩路 $v_0v_1\ldots v_{2m+1}$。定义 $M_1\subseteq E$

2. 偶图的匹配和覆盖

对于图G的任一顶点集合S，定义G中S的邻集为与S的顶点相邻的所有顶点的集，记为$N_G(S)$，简记为N(S)。

问题：假设n个男孩和m个女孩，每个男孩认识几个女孩，问在什么条件下，每个男孩都能娶到一个他认识的女孩？

这个问题用一个偶图G=(X,Y,E)表示，X和Y其中分别表示男孩和女孩的集合，如果男孩认识女孩，则两个点之间连一条边。问题变成：偶图G应满足什么条件才能有一个饱和X中每个顶点的匹配？

显然，为了使每个男孩都能取到他认识的一个女孩，其必要条件是其中的任意**k**$(1\leqslant k\leqq n)$个男孩都至少认识**k**个女孩。这个条件也是充分的。

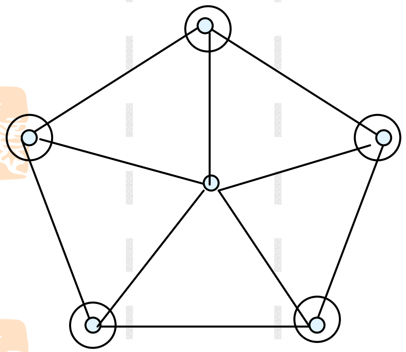
定理2 设G为具有二分类(X,Y)的偶图，则G存在饱和X的每个顶点的匹配当且仅当

$$|N(S)|\geqslant|S| \text{ 对所有} S\subseteq X\text{成立。} \quad (1)$$

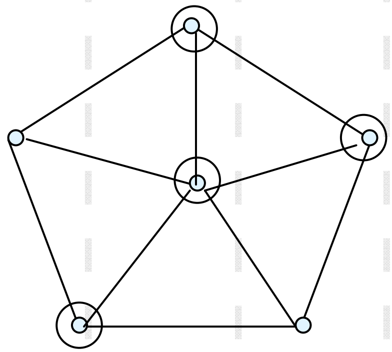
推论 若**G**是**k**正则偶图**(k>0)**，则**G**有完美匹配。

覆盖

图G的一个覆盖是指V的一个子集K，使得G的每条边都至少有一个端点在K中。一个覆盖K称为G的最小覆盖，如果G没有覆盖K_1使得$|K_1|<|K|$，如图。



○一个覆盖



一个最小覆盖

若**K**是**G**的覆盖，**M**是**G**的匹配，则至少包含**M**中每条边的一个端点。于是，对任何匹配**M**和任何覆盖**K**，均有|**M**|<|**K**|。因此，若**M***是最大匹配，$\tilde{K}$是最小覆盖，则有

$$|M^*| \le |\tilde{K}|$$

一般地，上式中的等号不成立，但对于偶图有

$$|M^*| = |\tilde{K}|$$

引理　设**M**是**G**的匹配，**K**是**G**的覆盖，如果|**M**|=|**K**|，则**M**是最大匹配，且**K**是最小覆盖。

定理3 设M*和 $\tilde{K}$ 分别是偶图G的最大匹配和最小覆盖，则

$$|M^*| = |\tilde{K}|$$

练习1 证明一个8*8方格棋盘移去其中两个对角上的1*1方格之后，不可能用1x2的长方形恰好填满。

练习2 矩阵的一行或一列统称为一条线。证明一个(0,1)矩阵中，包含所有1元素的线的最小条数等于两两都不在相同线上的1的元素的最大个数。

3. 完美匹配

图的分支根据它有奇数个或偶数个顶点分别称为奇分支或偶分支。$o(G)$表示G的奇分支个数。

Th(Tutte 1947) 图G有完美匹配当且仅当

$o(G-S)\le|S|$ 对所有$S\subset V$成立。

4. 应用

- **(1)** 人员分派问题

某公司准备分派**n**个工人$X_1,X_2,\ldots,X_n$做**n**件工作$Y_1,Y_2,\ldots,Y_n$,已知这些工人中每个人都能胜任一件或几件工作。试问能否把所有的工人都分派做一件他说胜任的工作?

构造一个具有二分类**(X,Y)**的偶图**G**，这里

$X=\{x_1,x_2,\ldots,x_n\}$， $Y=\{y_1,y_2,\ldots,y_n\}$，且x_i与y_j相连当且仅当工人X_i胜任工作Y_j.于是问题转化为**G**是否存在完美匹配的问题。根据**Hall**定理**2**，**G**或者存在完美匹配，或者存在**X**的子集**S**，使得$|N(S)|<|S|$.

给定任意一个具有二分类(X,Y)的偶图，或者找一个饱和X中每个顶点的匹配，或者找X的一个子集S，使|N(S)|<|S|。

Hungarian Method

以任一匹配M作为开始，可取M= ∅ 。

(1) 若M饱和X的每个顶点，停止（M为完美匹配）。否则，取X中M-不饱和顶点u，令S←{u} ,T← ∅。

(2) N(S)=T,停止（无完美匹配）。否则

(3) （此时N(S)⊃T）取y∈N(S)\T。若y为M饱和的，设yz ∈M，则

$S \leftarrow S \cup \{z\}$,$T \leftarrow T \cup \{y\}$

Goto(2)；否则(y为M-不饱和的)，存在M-可扩路P,令

$$M \leftarrow M \Delta E(P)$$

Goto(1).

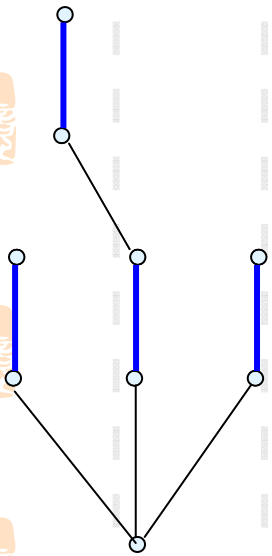
设M是G的匹配，u是X的一个M非饱和顶点。若树H$\subseteq$G满足(1) u$\in$V(H);(2) 对H的每个顶点v，H中唯一的(u,v)路是一条M交错路，则称树H是一条扎根于u的M交错路。

算法的关键就是找M-交错树。过程如下：

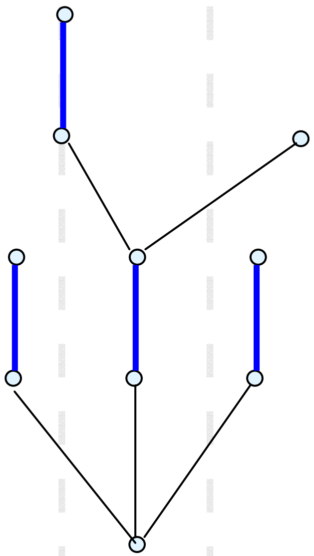
开始时H由一单点u组成。且按以下方式生长：

(1)除u外H的所有顶点是M饱和的，并且在M下匹配，图a，或者

(2)M包含不同于u的M非饱和顶点，图b.



图*a*



图*b*

如果情形(1)出现，则$S=V(H)\cap X$, $T=V(H)\cap Y$, 有$N(S)\supseteq T$；于是，或者$N(S)=T$或者$N(S)\supset T$。

(a)若$N(S)=T$，则$|N(S)|=|S|-1$, 这说明G没有饱和X所有顶点的匹配。

(b) 若$N(S)\supset T$，则存在$Y\backslash T$中的点y相邻于S中的顶点x，由于H的所有顶点，除了u外，都在M下配对，因此或者$x=u$，或者x和H的一个顶点配对，所以$xy\notin M$.

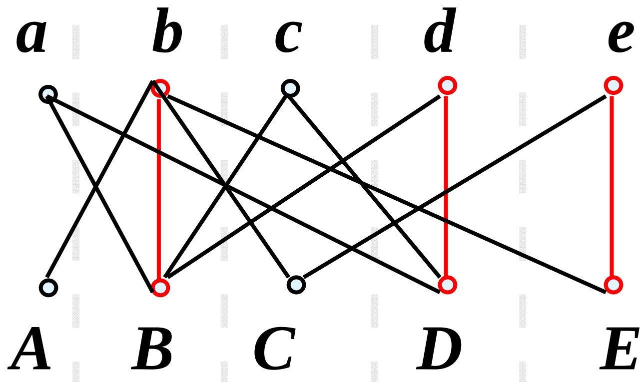


图 *G*

$S=\{a,b\}$

$T=\{B\}$

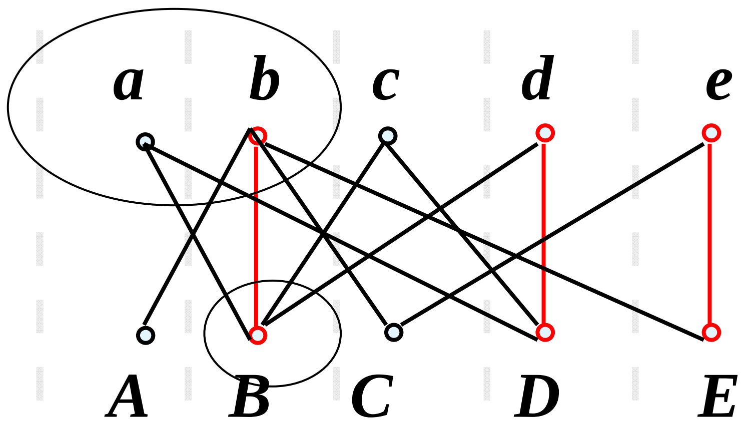
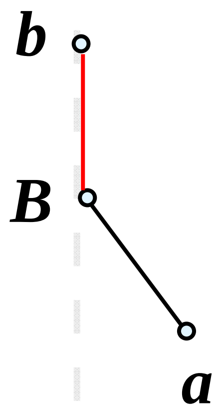


图 *G*



*M*交错树

$S=\{a,b,d\}$

$T=\{B,D\}$

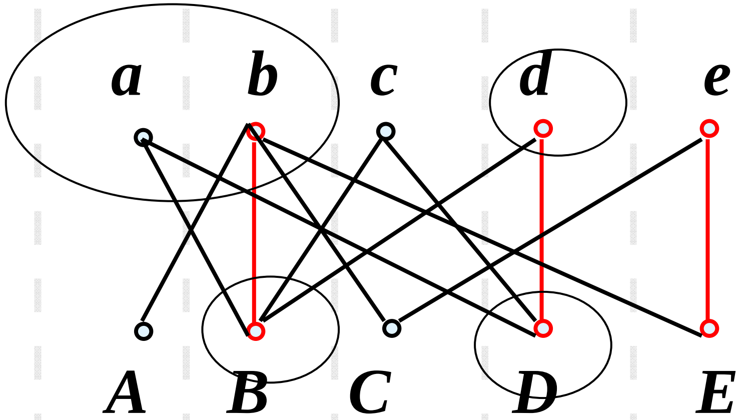
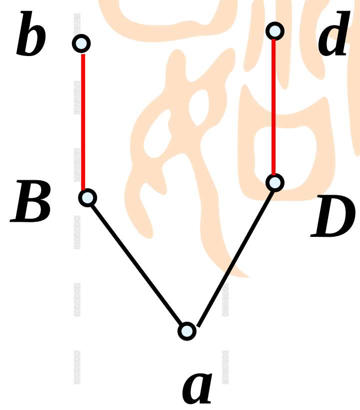


图 G



M交错树

$S=\{a,b,d\}$

$T=\{A,B,D\}$

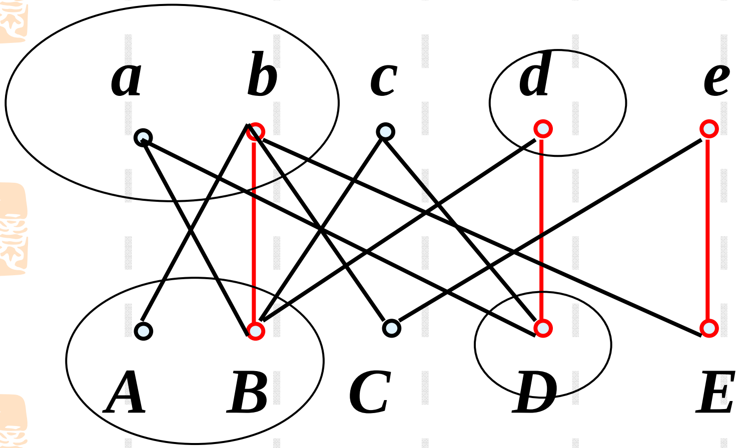
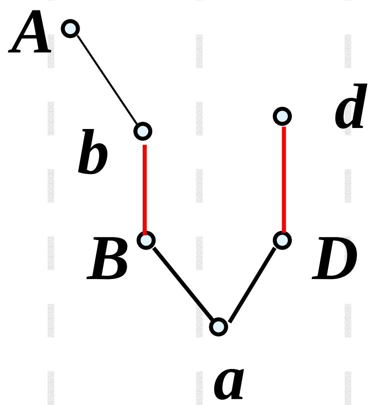


图 G



M交错树

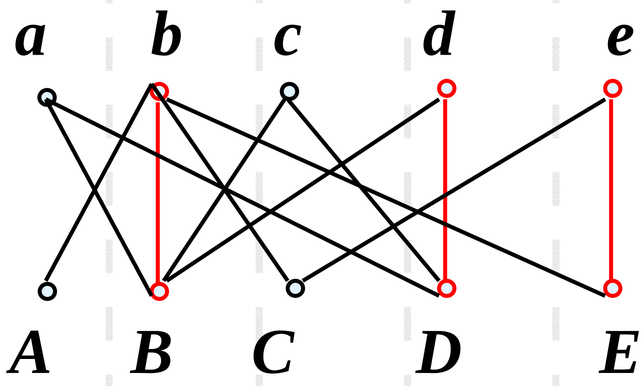
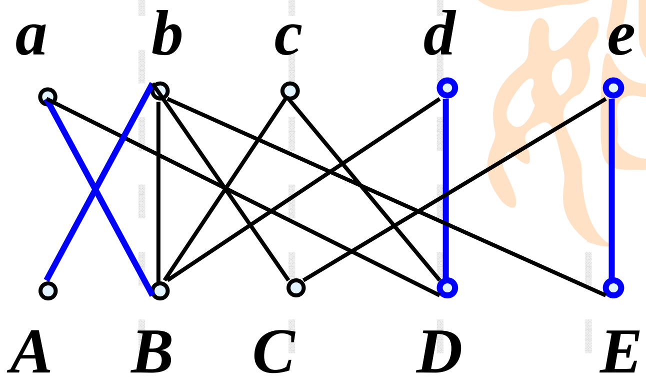
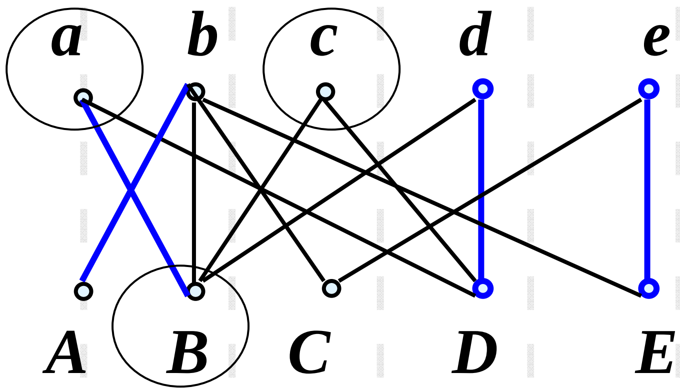


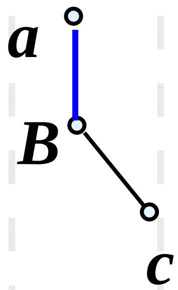
图 *G*



$S=\{a,c\}$

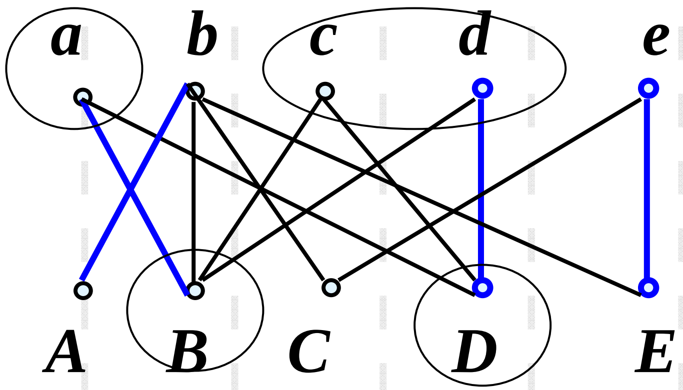
$T=\{B\}$

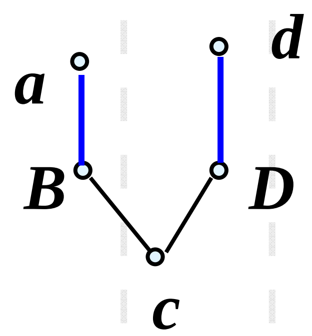




*M*交错树

S={*a*,*c*,*d*}





*M*交错树

T={*B*,*D*}

N(*S*)=*T*，图G中无完美匹配。

(2) 最优分派问题

可以利用**Hungarian Method** 确定工人分派工作的可行方案，且是一个有效的方法。把工人工作效率考虑进去，找一种分派方案使工人们总效率达到最大。寻找这种分派的问题称为最优分派问题。

考察一个具有二分类**(X,Y)**的偶图**G**， 这里 $X=\{x_1,x_2,\ldots,x_n\}$， $Y=\{y_1,y_2,\ldots,y_n\}$，边x_iy_j有权$w_{ij}=w(x_i,y_j)$，表示工人X_i做工作Y_j时的效率。于是问题转化为**G**是否存在完美匹配的问题。最优分派问题等价于在这个赋权图中寻找一个有最大权的完美匹配. 称这种匹配为最优匹配。

Kuhn and Munkras设计了一个求最优匹配的有效算法，将求最优匹配问题转化为可用算法求另一个图完美匹配的问题。

在顶点集合X∪Y中的每个元素(顶点)定义满足下列条件的实值函数*l*：对所有的$x\in X$，$y\in Y$，均有

$$l(x)+l(y)\geq w(x,y)$$

则把这个函数定义为该偶图的一个可行顶点标号，实数$l(v)$称为顶点v的标号。

不管边的权是什么，总可以定义一个可行标号：

$$\begin{cases} l(x)=\max\limits_{y\in Y} w(x,y) & x\in X \\ l(y)=0 & y\in Y \end{cases}$$

若l是可行标号，用E_l表示使$l(x)+l(y)\geq w(x,y)$中等号成立的边的集合，即

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

具有边集E_l的G的生成子图称为对应于可行顶点标号l的相等子图，用G_l表示。

相等子图与最优匹配之间的联系：

定理：设l是G的可行性顶点标号，若G_l包含完美匹配M^*，则M^*是G的最优匹配。

要求最优匹配，只需用Hungarian算法求G_l上的一个完美匹配。

首先给出任一可行顶点标号l，然后确定G_l，在G_l中选取任意匹配M，利用Hungarian算法找Gl的完美匹配。若在G_l中找到一个完美匹配，则该匹配则是G的最优匹配。否则Hungarian算法将终止与非完美的匹配M_1和一个既不包含M_1的可扩路，又不能在G_l中进一步生长的M_1交错树H。随后，把l修改为具有下述性质的另一个可行顶点标号 $\hat{l}$：

M1和**H**都包含在 $G_{\hat{l}}$ 中，并且**H**能够在$G_{\hat{l}}$ 中伸展。

必要时，不断地进行这种可行标号的修改，直到一个完美匹配在某个相等子图中找到为止。

1. 从任一可行顶点标号**l**开始，然后决定**Gl**，并且在**Gl**中选取任一匹配**M**。

2. 若**X**是**M**饱和的，则**M**是完美匹配，且是最优匹配；否则，令**u**是一个**M**非饱和顶点，令

S={u}，**T=**∅。

3. 若$N_{Gl}(S) \supset T$，则转**4**；否则，$N_{Gl}(S)=T$。计算

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

$$\hat{l}(v)=\begin{cases} l(v)-\alpha_l & if \quad v\in S \\ l(v)+\alpha_l & if \quad v\in T \\ l(v) & others \end{cases}$$

$\hat{l}$ 为可行顶点标号。 $\alpha_l>0$, $N_{G\hat{l}}(S)\supset T$

$l\leftarrow\hat{l}$, $G_l\leftarrow G_{\hat{l}}$.

4. 在$N_{Gl}(S)\backslash T$中选择一个顶点y，若y已被M饱和，且$yz\in M$，则$S\leftarrow S\cup\{z\}$， $T\leftarrow T\cup\{y\}$，转3；否则，设P是G_l中的M可扩(u,y)路，设 $\hat{M}=M\Delta E(P)$,

$M\leftarrow\hat{M}$，转1.

例$K_{5,5}$的权矩阵为W, W的元素

$$w_{ij}=w(x_i,y_j),\ \ 1\le i\le 5, 1\le j\le 5.$$

$$W=\begin{pmatrix}3&5&5&4&1\\2&2&0&2&2\\2&4&4&1&0\\0&1&1&0&0\\1&2&1&3&3\end{pmatrix}$$

取可行顶点标号如下：

$$l(y_i)=0,\quad i=1,2,3,4,5,$$

$$l(x_1)=\max_{1\le j\le 5}\{w_{1j}\}=\max\{3,5,5,4,1\}=5,$$

$$l(x_2)=\max_{1\le j\le 5}\{w_{2j}\}=\max\{2,2,0,2,2\}=2,$$

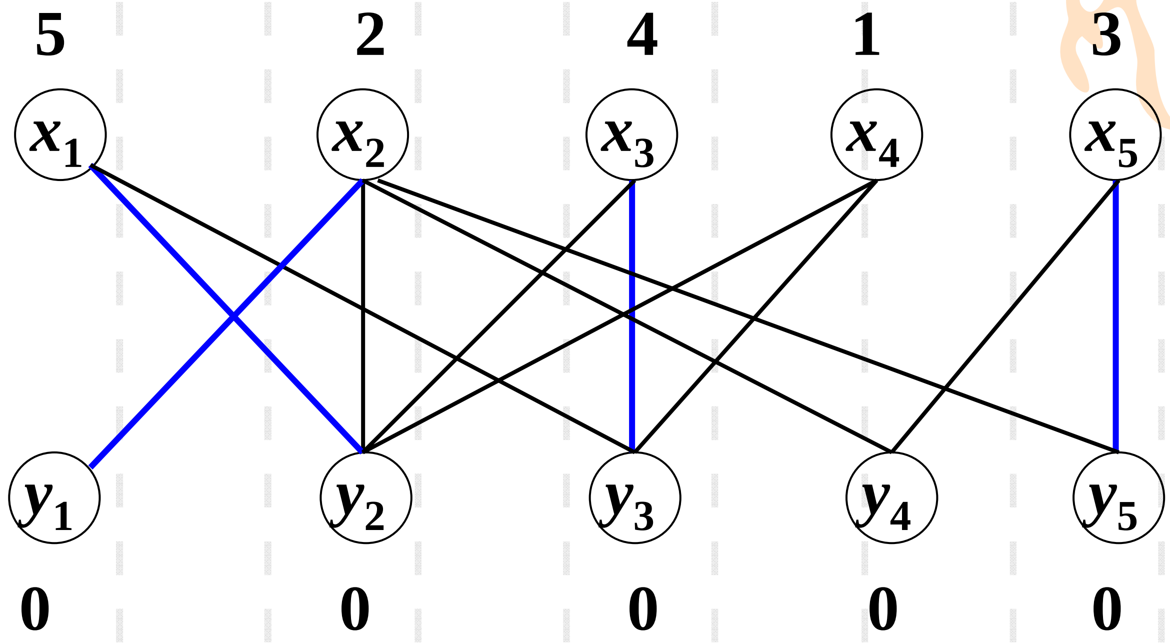
$$l(x_3)=\max_{1\le j\le 5}\{w_{3j}\}=\max\{2,4,4,1,0\}=4,$$

$$l(x_4)=\max_{1\le j\le 5}\{w_{4j}\}=\max\{0,1,1,0,0\}=1,$$

$$l(x_5)=\max_{1\le j\le 5}\{w_{5j}\}=\max\{1,2,1,3,3\}=3.$$

构造图G_l如下：

$M=\{x_1y_2, x_2y_1, x_3y_3, x_5y_5\}$



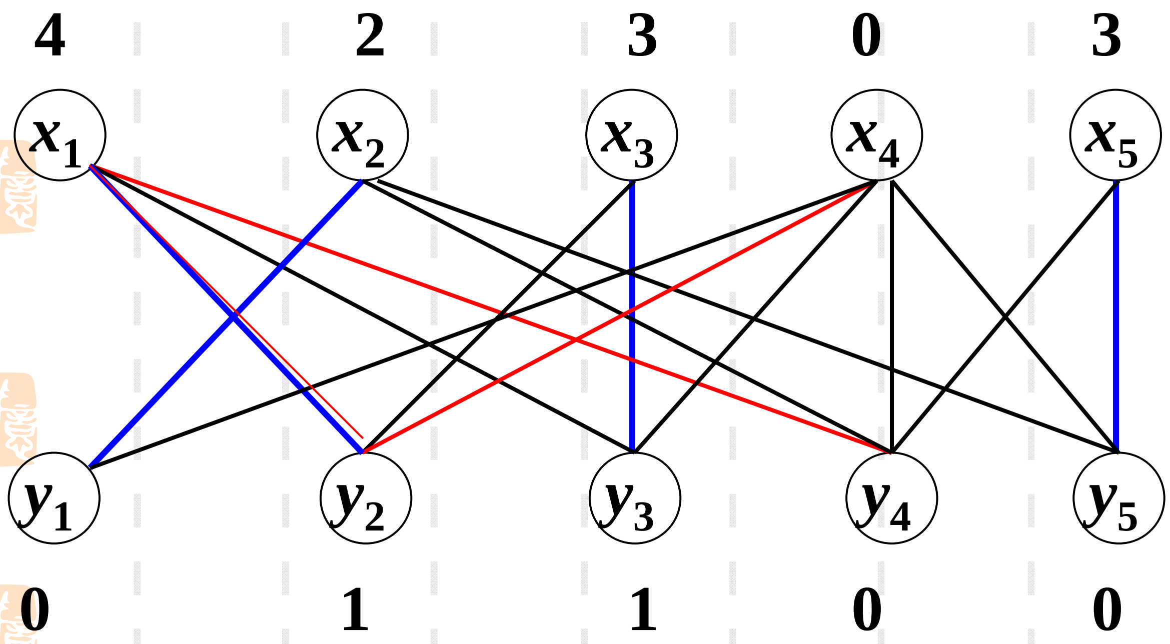
图G_l

G_l无完美匹配，其顶点标号需要修改。

取未被M饱和的点x_4, S={x_4, x_3, x_1}, T={y_3, y_2}.

$N_{Gl}(S)=T$，取

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

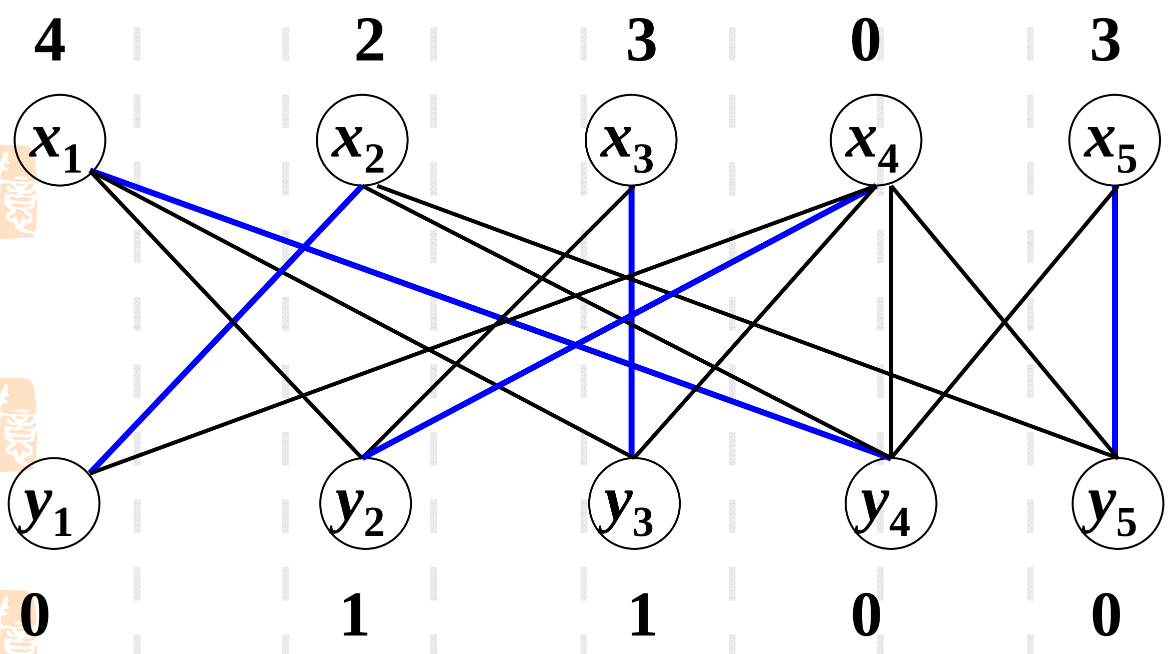


图G_l

取未被M饱和的点x_4, $S=\{x_4, x_3, x_1\}$, $T=\{y_3, y_2\}$.

$N_{Gl}(S)=T$，取

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



图G_l

98年全国大学生数学建模竞赛B题
“最佳灾情巡视路线”

今年(1998年)夏天某县遭受水灾. 为考察灾情、组织自救，县领导决定，带领有关部门负责人到全县各乡（镇）、村巡视. 巡视路线指从县政府所在地出发，走遍各乡（镇）、村，又回到县政府所在地的路线.

1）若分三组（路）巡视，试设计总路程最短且各组尽可能均衡的巡视路线.

2）假定巡视人员在各乡（镇）停留时间T=2小时，在各村停留时间t=1小时，汽车行驶速度V=35公里/小时. 要在24小时内完成巡视，至少应分几组；给出这种分组下最佳的巡视路线.