北京郵電大學
Beijing University of Posts and Telecommunications

图论及其应用

北京邮电大学理学院 寇彩霞

Ch6 网络流问题

Ch6 网络流

- 网络应用广泛：

  交通网、天然气输送网、通信网、物联网…

  集成电路网、神经网络、量子关联网络……

  安全网络、社交网络……

- 研究网络中的个体、之间的关联……

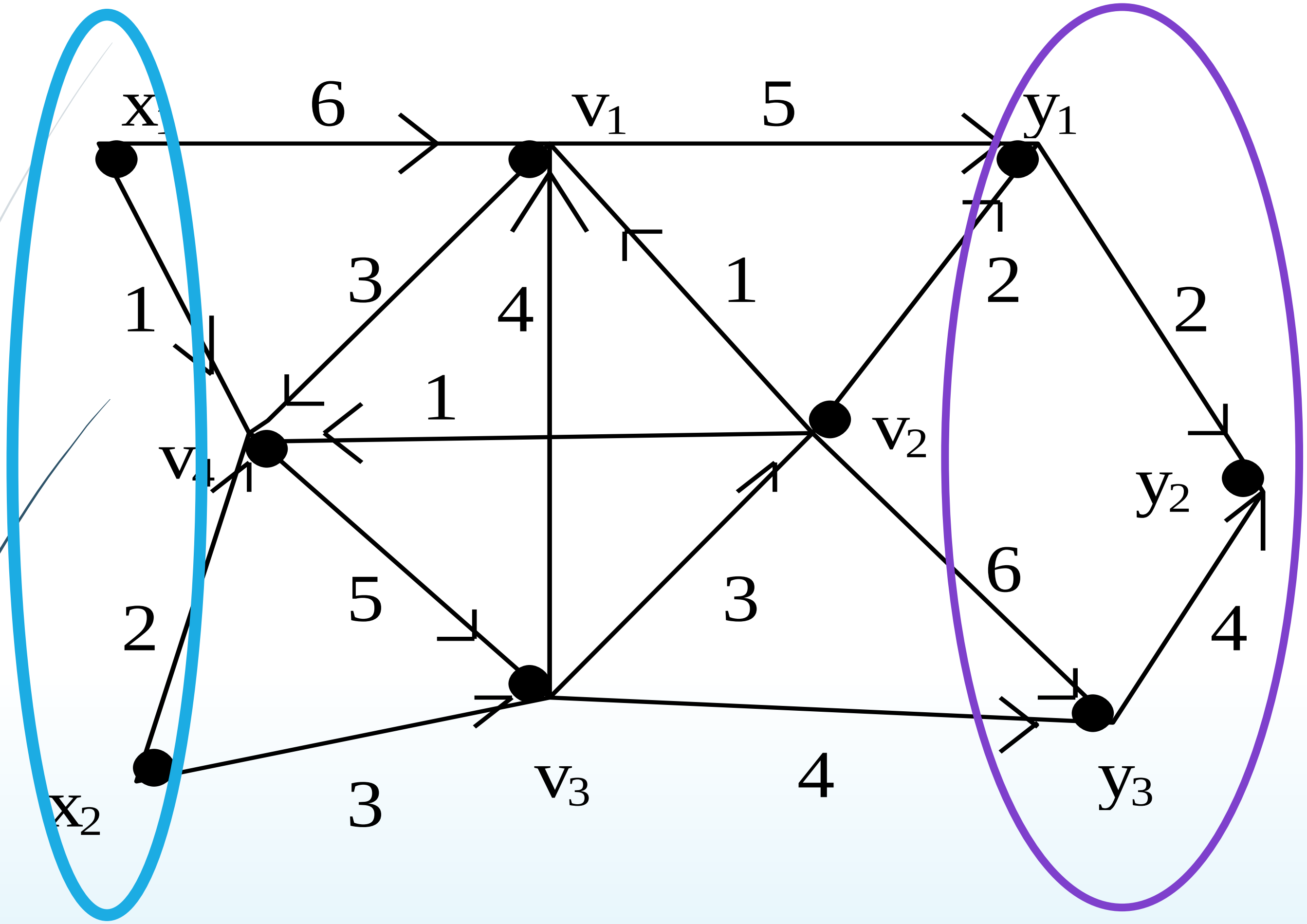
Ch6 主要内容

- 网络与流
- 网络最大流
- 最小费用流问题

网络

- 一个网络（Network）N是由一有向图（称为基础有向图（underlying digraph）；及二特定的不相交顶点子集X和Y；以及弧集A上定义的非负整数值函数c(·)（容量函数）。
- X的每个顶点称为发点（source）；
- Y的每个顶点称为收点（sink）；
- I = V \ (X ∪ Y)中每个顶点称为中间顶点（intermediate vertex）；
- 每弧a上c(a)的值称为a的容量（capacity）。

例



流

- 定义在弧集A(D)上的整数值函数f(·)若满足以下条件，就称为网络N上的流（flow）：

① 容量约束（capacity constraint）

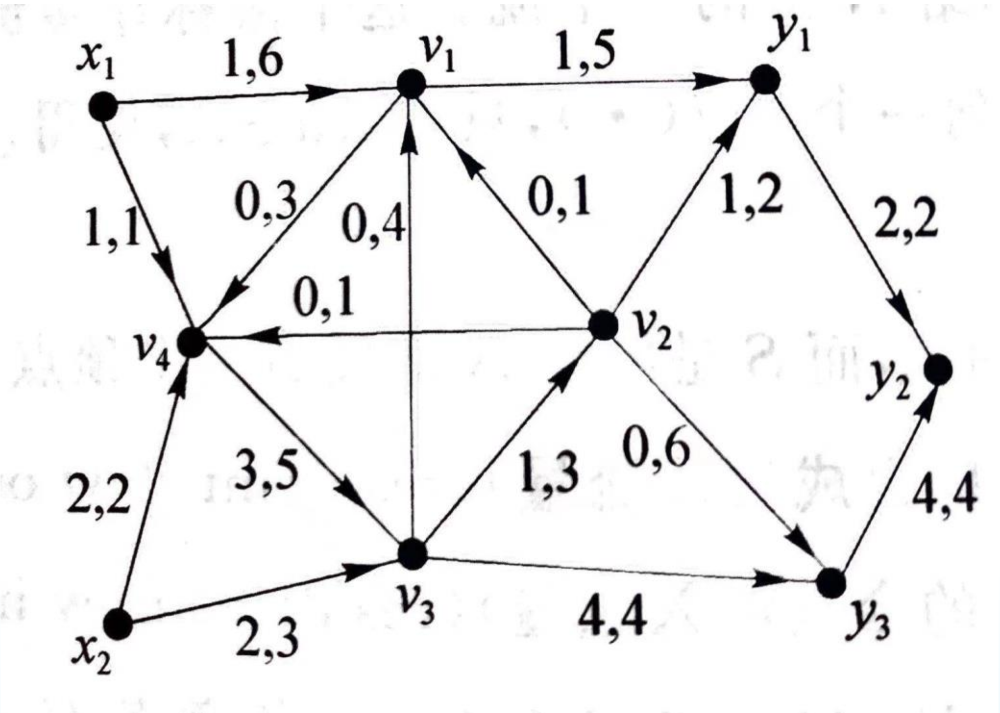
$$0 \le f(a) \le c(a), \quad \forall a \in A(D)$$

② 守恒条件（conservation condition）

$$f^+(v) = f^-(v), \quad \forall v \in I\text{中间顶点}.$$

- 流的存在性，零流

例：流

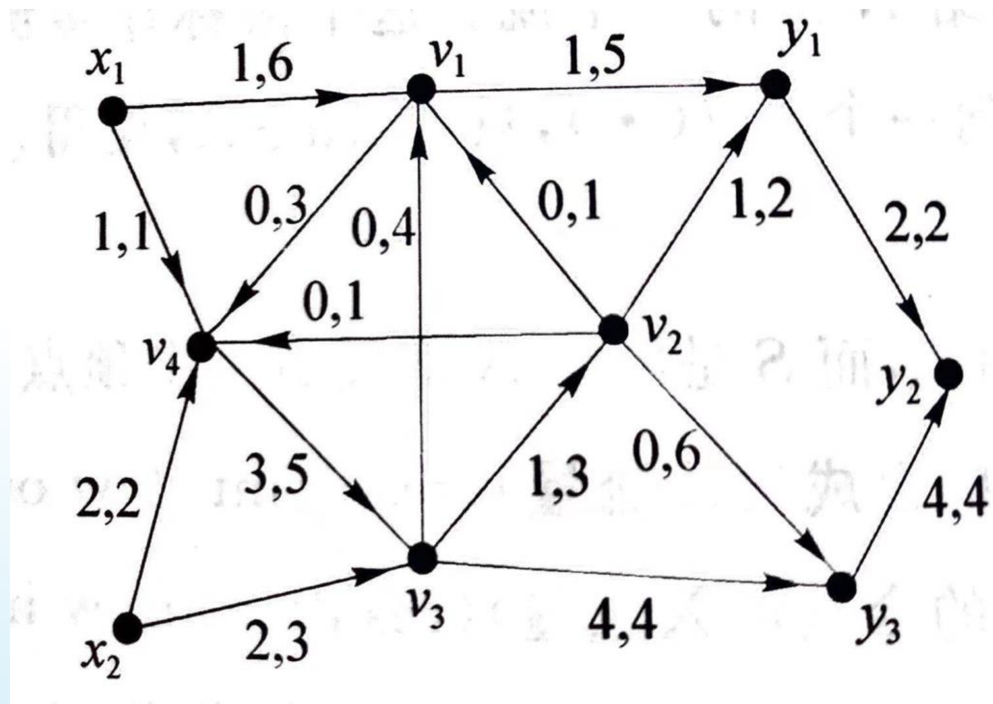


x1
1,6
v1
1,5
y1
1,1
0,3
0,4
0,1
1,2
2,2
v4
0,1
v2
y2
2,2
3,5
1,3
0,6
4,4
v3
4,4
y3
x2
2,3

表示:

$$f^{+}(v)=\sum_{w\in N^{+}(v)} f(v,w)=f(\{v\},V\setminus\{v\});$$

$$f^{-}(v)=\sum_{w\in N^{-}(v)} f(w,v)=f(V\setminus\{v\},\{v\});$$



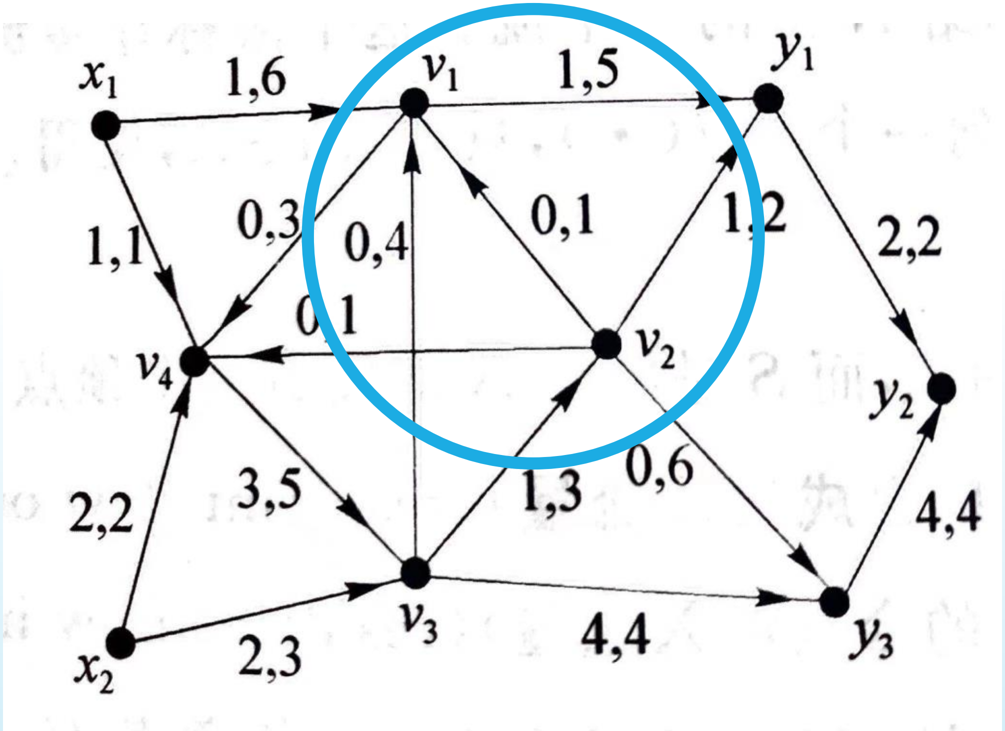
合成流量：设 f(.) 为网络N上的任一流，对任一顶点子集S

$$f^+(S) = f(S,\overline{S}) = \sum_{uv \in (S,\overline{S})} f(u,v);$$

$$f^-(S) = f(\overline{S},S) = \sum_{wz \in (\overline{S},S)} f(w,z);$$

$f^+(S) - f^-(S)$：流出S的合成流量；

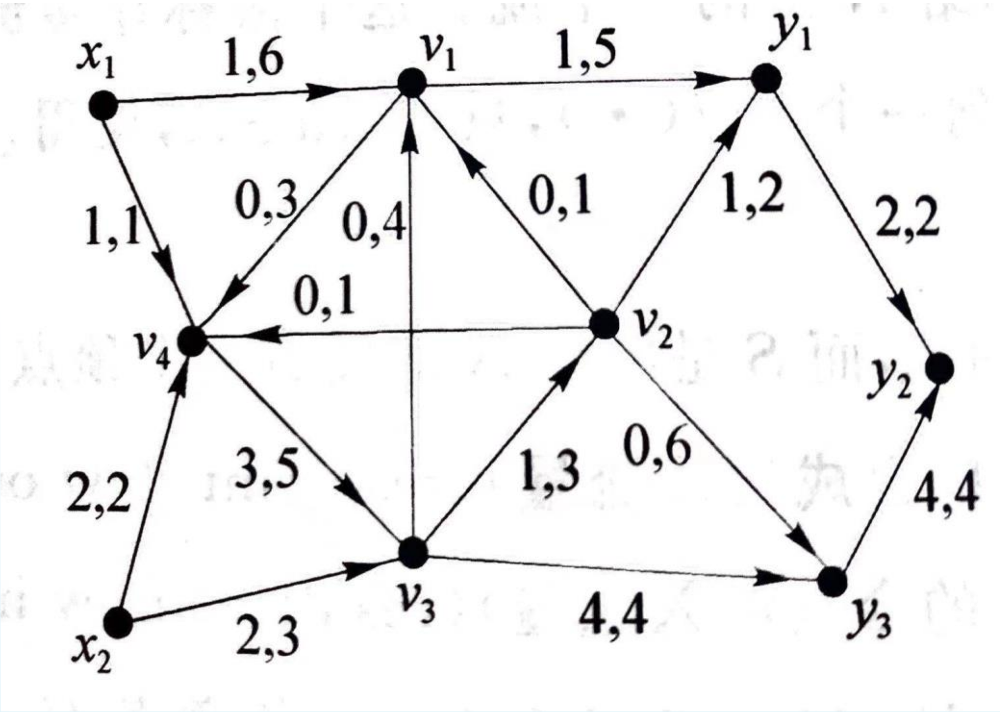
S = {v1, v2}



流值：

- 网络N=（X,Y,I,A,c）及网络N上的任意流f(.),

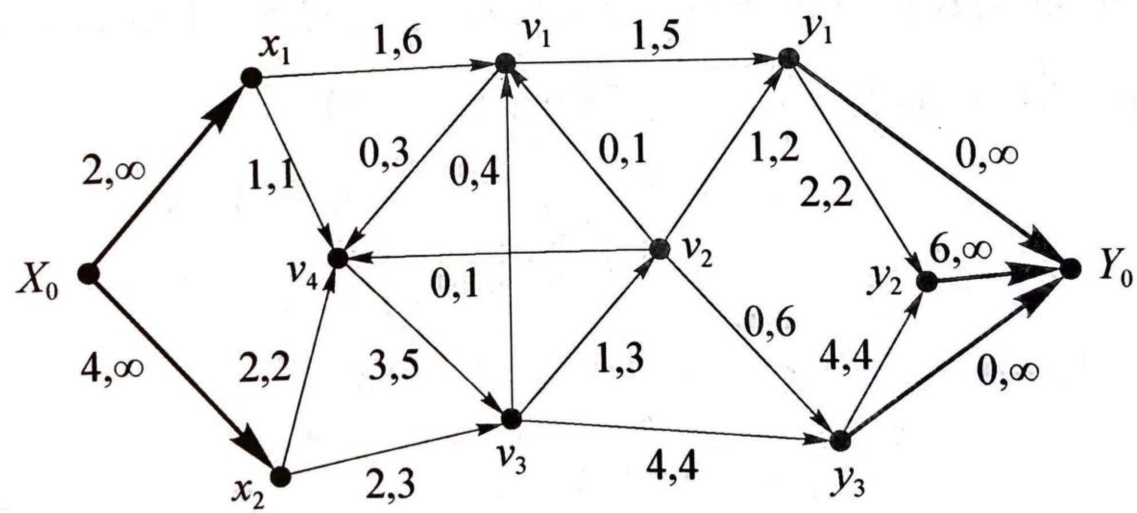
称 $valf = f^{+}(X) - f^{-}(X)$: f的流值



最大流

- 流 f为最大流（maximum flow）⇔ 不存在流f‘，使val *f’* > val*f* 。
- 求最大流时考虑简化模型：
  - 网络N中不存在有向圈C，使得C中每条弧中的流量大于0；
  - 网络N中不存在(xi,xj)-有向路P，使得P中每条弧中的流量大于0；
  - 只需讨论单发点，单收点的网络

➢只需讨论单发点，单收点的网络

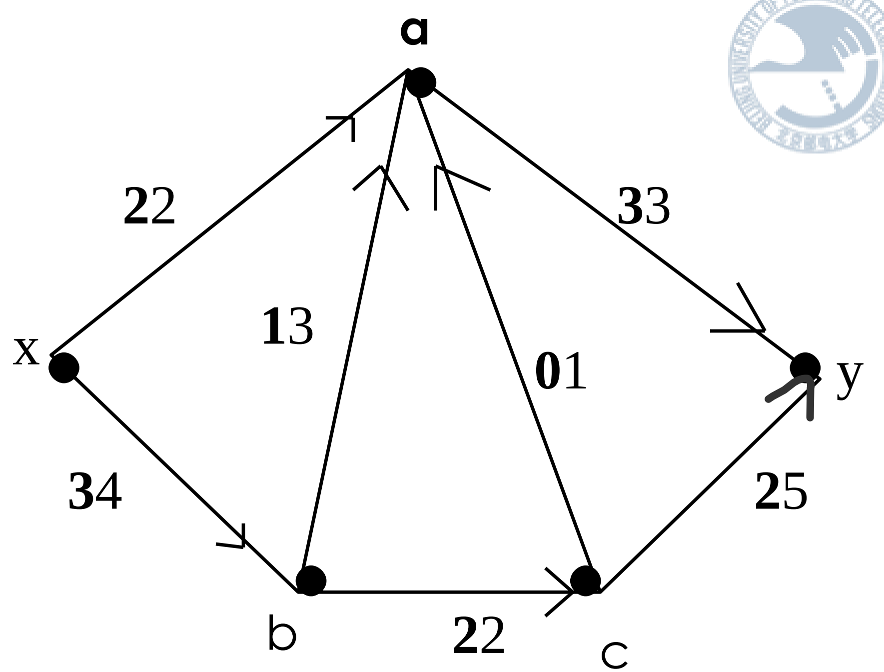


➢转化前后网络有相同的最大流的值

割

- 网络N只有单发点x及单收点y, 顶点子集S $\subset$V(D), 如果x$\in$S, y$\in$ $\bar{S}$, 称K=(S, $\bar{S}$)为N中的**割(cut)**
- 称 $capK = c(S,\bar{S}) = \sum_{\alpha \in K} c(\alpha)$ 为割 K的**容量**
- **称K为网络的最小割(minimum cut)** $\Leftrightarrow$ 不存在割K'使 cap K' < cap K。

例图



(flow, capacity)

- Val f = 5 共有（8）个割：
- S={x}, K1={xa,xb} cap(K1)=6
- S={x,a}, K2={xb,ay} ,7
- S={x,b},K3={xa,ba,bc},7
- S={x,c},K4={xa,xb,ca,cy},12
- S={x,a,b},K5={bc,ay},5 最小割
- S={x,a,b,c}, K6={ay,cy},8

定理6.2

- **定理6.2** 对N中任一流 f 及任一割$(S, \bar{S})$，有

$$valf = f^{+}(S) - f^{-}(S)$$

- 证明：注意到

$$f^{+}(v) - f^{-}(v) = \begin{cases} valf, v = x; \\ 0, v \in S \setminus \{x\} \end{cases}$$

因此有:

$$valf = \sum_{v \in S} (f^{+}(v) - f^{-}(v)) = f^{+}(S) - f^{-}(S)$$

定理6.3

- 称弧a为：

f-零的	$\Leftrightarrow$	$f(a) = 0$ ；
f-正的	$\Leftrightarrow$	$f(a) > 0$ ；
f-不饱和的	$\Leftrightarrow$	$f(a) < c(a)$ ；
f-饱和的	$\Leftrightarrow$	$f(a) = c(a)$ 。

定理6.3 对N中任一流 f及任一割K =(S, $\bar{S}$)有 **val f ≤ cap K**；上式等号成立 ⇔ K =(S, $\bar{S}$)中每弧为f-饱和的， 且($\bar{S}$,S) 中每弧为 f-零的。

- 证明：由容量约束知

$$f_+(S) \le \text{cap } K \quad (1)$$

$$f_-(S) \ge 0 \quad (2)$$

再由定理6.2知定理 $valf = f^+(S) - f^-(S)$

知 valf ≤ cap K；又

(1) 中等号成立 ⇔ （S, $\bar{S}$）中每弧为f-饱和的；

(2) 中等号成立 ⇔ （$\bar{S}$, S）中每弧为f-零的；

从而定理的第二个结论也成立。 #

推论

- 显然，若 f* 为最大流，K为最小割，则

  $$\mathrm{val} f^* \leq \ \mathrm{cap}\ K。$$

- 推论6.1 设流 f 及割K，使val f = cap K，则f为最大流，K为最小割。

最大流最小割定理

北京郵電大學 Beijing University of Posts and Telecommunications

- 设f为网络N中的流，P为N中一条路（不一定是有向路），

  令

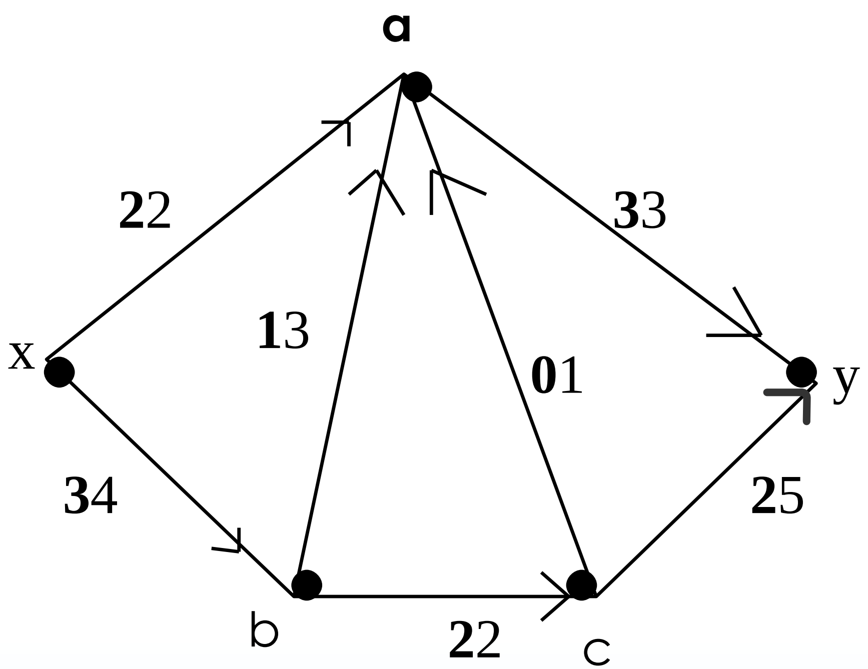
$$l(a)=\begin{cases} c(a)-f(a) & a\text{为}P\text{的顺向弧} \\ f(a) & a\text{为}P\text{的反向弧} \end{cases}$$

$$l(P)=\min_{a\in A(P)} l(a)$$

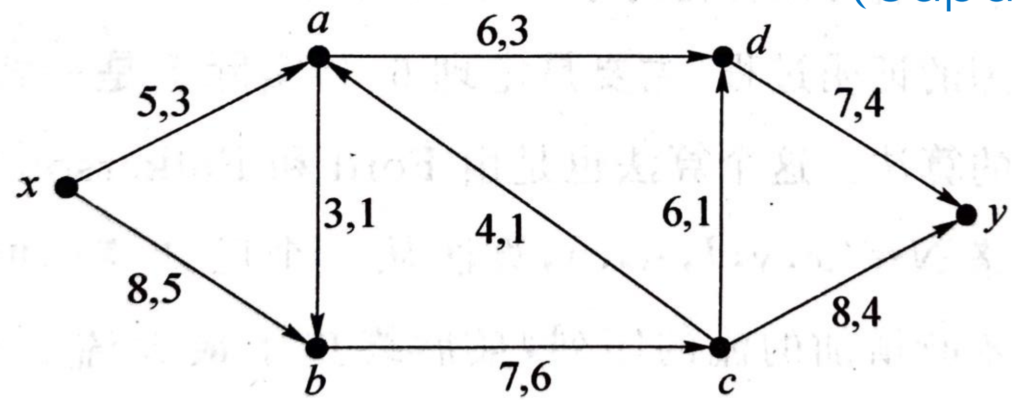
称路P为f-饱和的 $\Leftrightarrow$ l(P) = 0 ；

f-不饱和的 $\Leftrightarrow$ l(P) > 0 ；

f-可增路（f-incrementing path）P $\Leftrightarrow$ P是以发点x为起点，以收点y为终点的 f-不饱和路 。



- 路xbacy: f-饱和路？
- 有没有f-可增路？



- 路 xacy: 不饱和？f可增路？L(xacy)=?

修改流

- 若N中有一f-可增路P，则，易见，f 不是最大流。这时，可沿P输送一附加流 而得一新流 f′：

$$f'(a)=\begin{cases} f(a)+\iota(P) & a\text{为}P\text{的顺向弧} \\ f(a)-\iota(P) & a\text{为}P\text{的反向弧} \\ f(a) & \text{其它} \end{cases}$$

容易验证， f′也是一个流，而且流值为valf +l(P)

并称 f′为基于P的修改流（revised flow based on P）

定理6.4: N中的流 f为最大流iff N不含f-可增路。

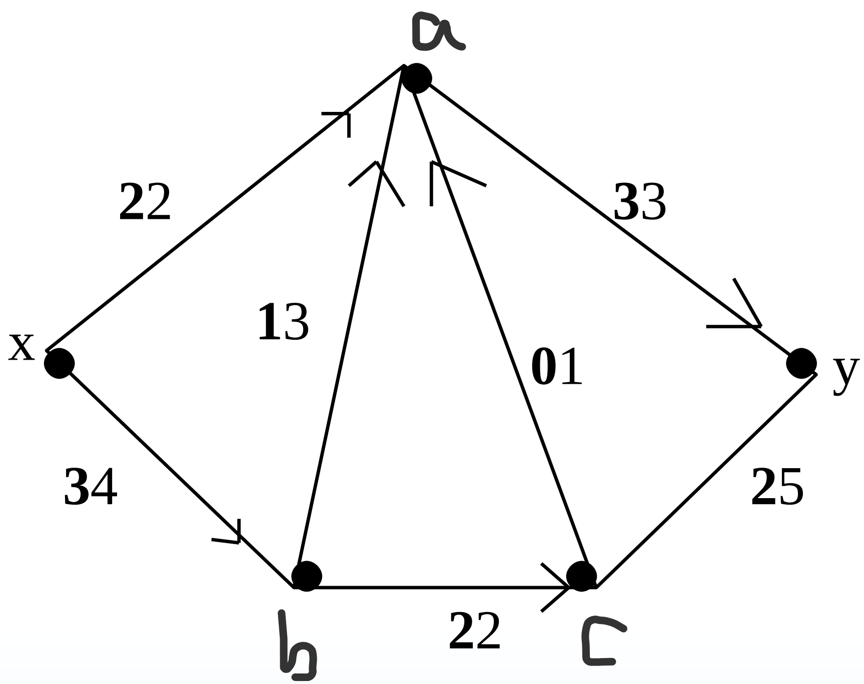
- 证明：$\Rightarrow$ 反证，若N中含一f-可增路P，则f不是最大流，因为基于P的修改流f'的值更大。

  $\Leftarrow$：令 $S = \{ v \mid \exists$ f-不饱和（x, v）-路$\}$，

  则显然有 $x \in S$，$y \in \bar{S}$；$\therefore K = (S, \bar{S})$ 为割。

  注意到，这时$(S, \bar{S})$的任一弧 $a = (u,v)$一定是f-饱和的。否则，由于$u \in S$，存在f-不饱和(x,u) -路Q 。因此Q可通过a延伸为f-不饱和(x,v) -路，从而 $v \in S$，矛盾。类似地，$(\bar{S},S)$中任一弧 一定是f-零的。

  因此，由定理6.3知，$\mathrm{val} f = \mathrm{cap}\, K$ 。从而由推论6.1知，f为最大流，同时K为最小割。 #



S = { v | ∃ f-不饱和（x, v）-路}={x,b,a}

K = (S, $\bar{S}$) 为割

这时(S, $\bar{S}$)的任一弧 a = (u,v)一定是f-饱和 的。

最大流最小割定理

- 定理6.5(Ford & Fulkerson 1956)

  在任一网络中，最大流的值等于最小割的容量。

求最大流的算法

- 原理：

1. 以一已知流 f（例如，零流）作为开始。
2. 系统搜索 f-可增路P。若P不存在，停止（f-为最大流）。
3. 若P存在，求出基于P的修改流f′，令 f ← f′，并转到1。

系统搜索f-可增路P标号法 （通过标号‘生长’f-不饱和树T）

- 开始，标x以l(x) = ∞ 。 （此后，在T的生长过程中，T中每个顶点 将标 以 l(v) = ι (Pv)，其中Pv是T中唯一的(x, v)- 路。）

(1) 若 a =(u,v)为f-不饱和弧，且u已标号， 而v未曾，则标v以

$$l(v) = \min\{ l(u),\ c(a) - f(a) \}。$$

(2) 若 a =(v,u)为f-正的，且u已标号，而v未曾，则标v以

$$l(v) = \min\{ l(u),\ f(a) \}$$

上述标号过程一直进行到：或者y已标号（“breakthrough”，找到了f-可增路）；或者所有已标号顶点都已扫描，但无顶点可再标号（f为最大流）。

标号程序 (labelling method,Ford & Fulkerson,1957)

以已给流 f（例如，零流）作为开始;

① 标x以 l(x) = ∞。扫描 （scan） x， u=x。

② 对正在扫描的（已标号）顶点u,

(i) 检查每条以u为尾的弧 a = (u,v)。如果 a 为f-不饱和的，且顶点v未标号，

则标v以 l(v) = min{ l(u), c (a) - f (a) } 。

(ii) 检查每条以u为头的弧 a = (v,u) 。如果 a 为f-正的，且顶点v未标号，

则标v以 l(v) = min{ l(u), f(a) } 。

③ 若y已标号（ ‘break through’，找到了一条f-可增路），则转到④；否则选一未曾扫描的已标号顶点进行扫描，并回到②。如果已标号顶点都已扫描过，停止（得最大流。且由已标号顶点集S，得最小割（S， $\bar{S}$））。

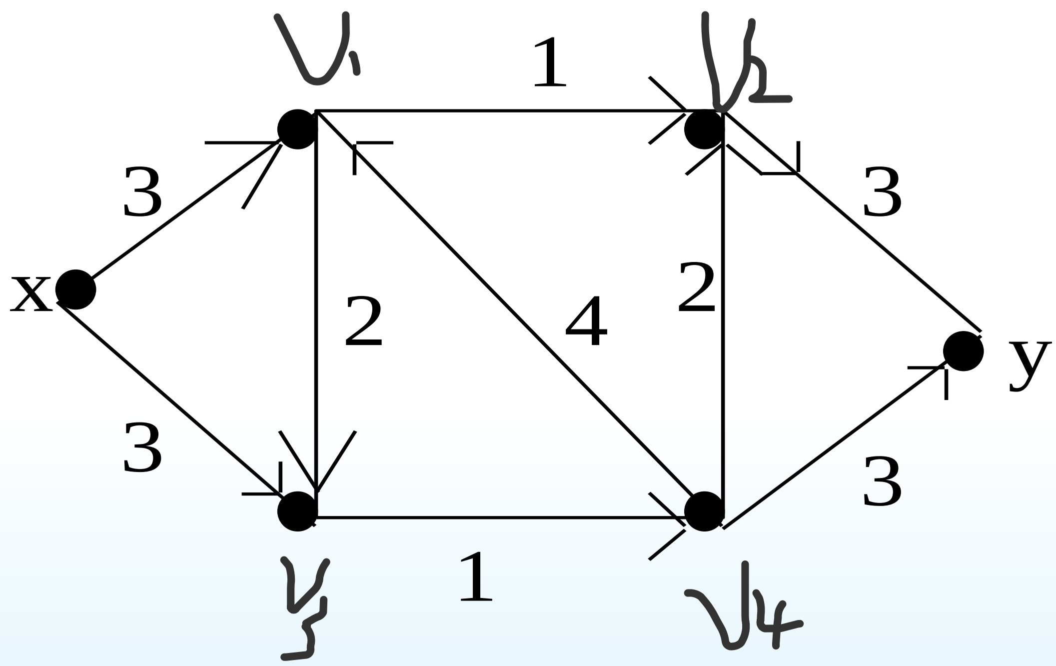
④ 找一f-可增路P，令

$$\tilde{f}(a)=\begin{cases} f(a)+l(y) & \text{当}a\text{为}P\text{上的顺向弧} \\ f(a)-l(y) & \text{当}a\text{为}P\text{上的逆向弧}, \\ f(a) & \text{其它} \end{cases} \qquad \tilde{f}\to f$$

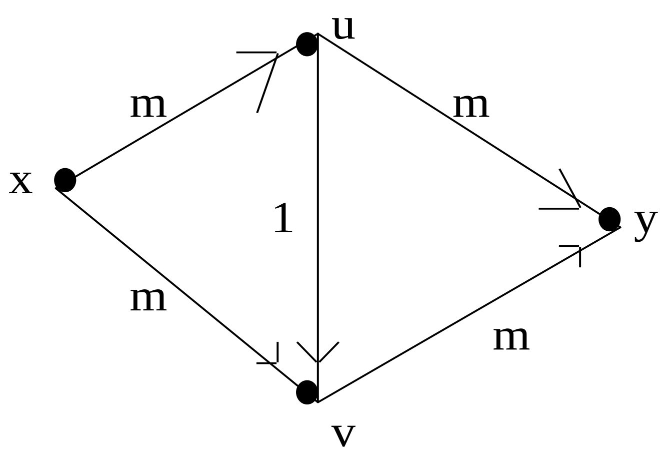
去掉全部标号，并回到① 。

例图

从零流开始，利用上述算法（找f可增路的）找下图的最大流



练习：

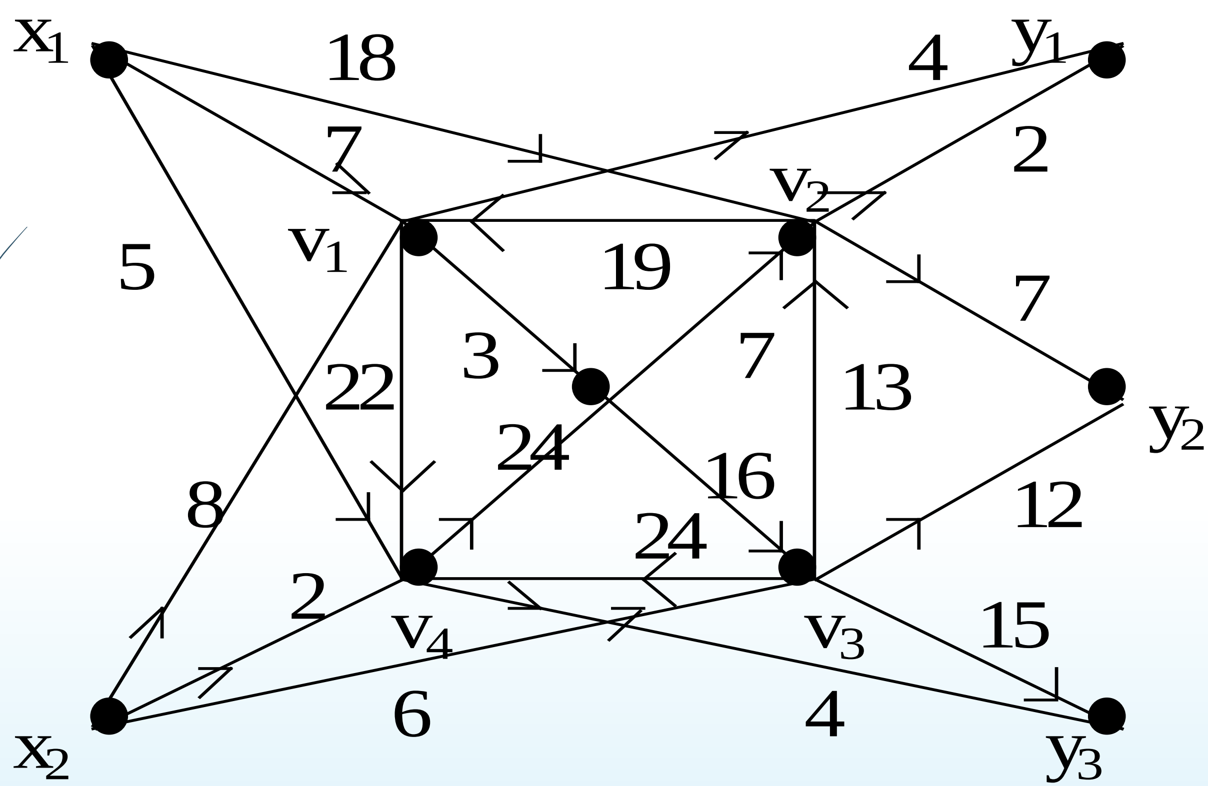


标号程序算法分析

- 上述算法（深度优先）还不是‘好’算法，例如上图中的网络，其最大流的值为**2m**。但若标号程序以零流开始，且反复地选取**xuvy**及**xvuy**为 **f-**可增路，则总共要进行**2m+1**次标号程序

- **Edmonds & Karp (1970)** 证明，若在上述标号程序中采用**first labelled first scan** （即广度优先）法则，则，可使该算法成为好算法（复杂性为**O(**$\nu\varepsilon$**2)**）

例图



最小费用流

- 不仅考虑流值达到一定量，还关心费用。
- 最小费用流：满足流值到达一定要求还要使总费用最小
- 当流不唯一时，在流值相同的流中求一个流，使得该流的总费用最小。
- 特别的，求流值最大的最小费用流的问题，称为：最小费用最大流

算法基本思想

- 最大流的标号算法过程中，考虑费用最小的可增路。

最小费用路算法

- P179 算法
- 例6-2