网络流

- Comzyh博客

网络流问题

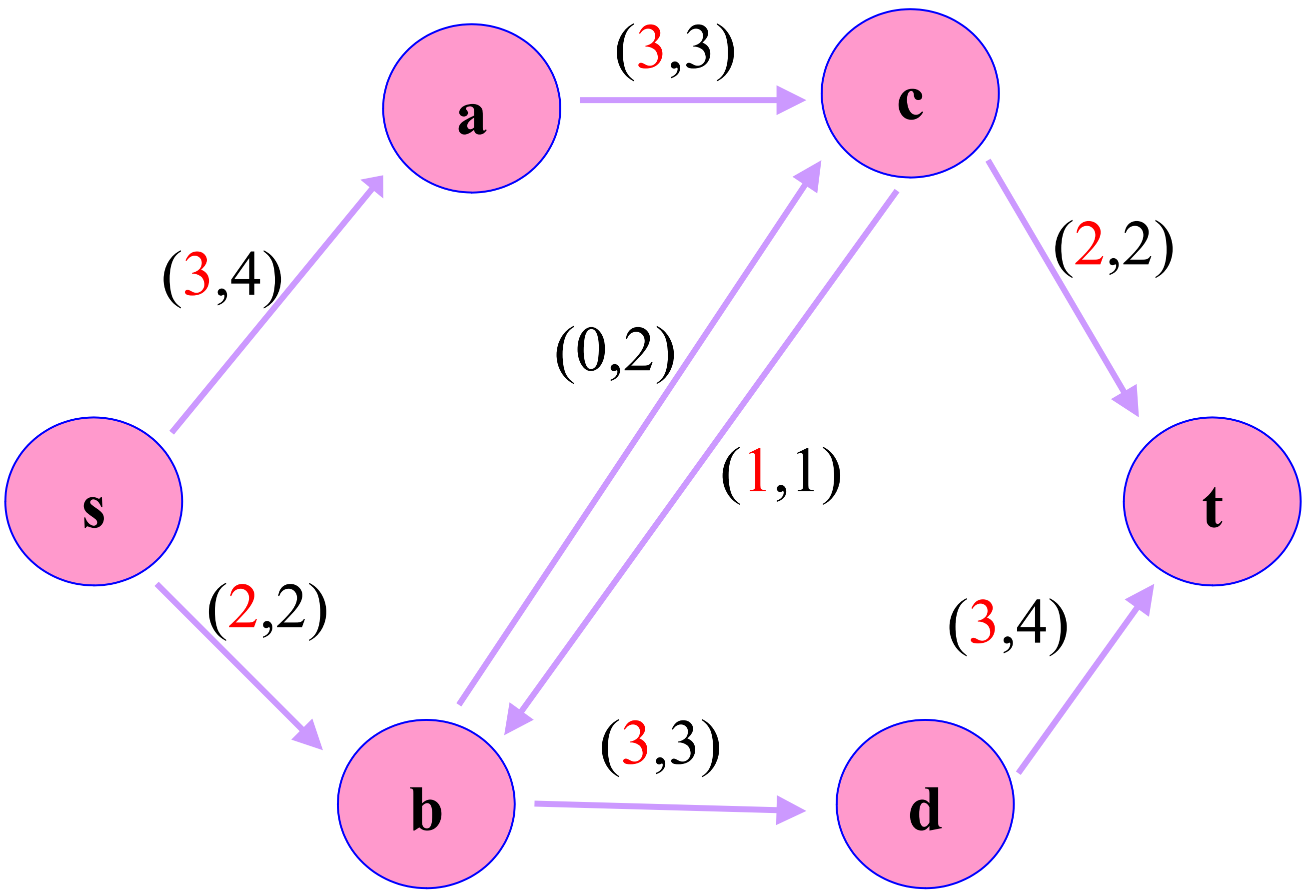
由于人类对自然资源的消耗，人们意识到大约在 2300 年之后，地球就不能再居住了。于是在月球上建立了新的绿地，以便在需要时移民。令人意想不到的是，2177 年冬由于未知的原因，地球环境发生了连锁崩溃，人类必须在最短的时间内迁往月球。现有 n 个太空站位于地球与月球之间，且有 m 艘公共交通太空船在其间来回穿梭。每个太空站可容纳无限多的人，而每艘太空船 i 只可容纳 H[i] 个人。每艘太空船将周期性地停靠一系列的太空站，例如：(1，3，4) 表示该太空船将周期性地停靠太空站 134134134…。每一艘太空船从一个太空站驶往任一太空站耗时均为 1。人们只能在太空船停靠太空站 (或月球、地球) 时上、下船。初始时所有人全在地球上，太空船全在初始站。试设计一个算法，找出让所有人尽快地全部转移到月球上的运输方案。

网络流问题

由文件 input.txt 提供输入数据。文件第 1 行有 3 个正整数 n（太空站个数）， m（太空船个数）和 k（需要运送的地球上的人的个数）。其中 1<=m<=13, 1<=n<=20,1<=k<=50 。接下来的 n 行给出太空船的信息。第 i+1 行说明太空船 pi 。第 1 个数表示 pi 可容纳的人数 Hpi；第 2 个数表示 pi 一个周期停靠的太空站个数 r， 1<=r<=n+2；随后 r 个数是停靠的太空站的编号 (Si1,Si2,…,Sir)，地球用 0 表示，月球用 -1 表示。时刻 0 时，所有太空船都在初始站，然后开始运行。在时刻 1， 2， 3… 等正点时刻各艘太空船停靠相应的太空站。只有在 0,1,2… 等正点时刻才能上下太空船。

将全部人员安全转移所需的时间输出到文件 output.txt 中。如果问题无解，则输出 0 。

一个简单的例子 - 网络的最大流问题



网络可以被想象成一些输水的管道．括号内右边的数字表示管道的容量，左边的数字表示这条管道的当前流量。

最大流为 5 ？

一些符号和定义

- V 表示整个图中的所有结点的集合 .
- E 表示整个图中所有边的集合 .
- G = (V,E) , 表示整个图 .
- s 表示网络的源点 ,t 表示网络的汇点 .
- 对于每条边 (u,v), 有一个容量 c(u,v) (c(u,v)>=0)
- 如果 c(u,v)=0， 则表示 (u,v) 不存在于网络中。
- 如果原网络中不存在边 (u,v)， 则令 c(u,v)=0
- 对于每条边 (u,v), 有一个流量 f(u,v).

网络流的三个性质

1、*容量限制*：f[u,v]<=c[u,v]
2、*反对称性*： f[u,v] = - f[v,u]
3、*流量平衡*： 对于不是源点也不是汇点的任意结点，流入该结点的流量和等于流出该结点的流量和。
结合反对称性，流量平衡也可以写成：

$$\sum_{u\in V} f(v,u) = 0$$

只要满足这三个性质，就是一个合法的网络流，也称为可行流。可行流至少有一个零流
。

最大流问题

- 定义一个网络的流量（记为 |f| ）= $\sum_{v \in V} f(s,v)$
- **最大流**问题，就是求在满足网络流性质的情况下，|f| 的最大值。

弧的分类

- 若给定一个可行流 $F=(F_{ij})$, 我们把网络中 $F_{ij}=C_{ij}$ 的弧称作饱和弧，　$F_{ij}<C_{ij}$ 的弧称作非饱和弧，　$F_{ij}=0$ 的弧称作零流弧，　$F_{ij}>0$ 的弧称作非零流弧
- 若 P 是网络中联结源点 s 和汇点 t 的的一条路 (不用管边的有向性)，我们定义路的方向是从 V_s 到 V_t ，则路上的弧被分为两类：一类与路的方向一致，称为前向弧；另一类和路的方向相反，称为后向弧

残量网络

- 为了更方便算法的实现，一般根据原网络定义一个残量网络。其中 $r(u,v)$ 为残量网络的容量。
- $r(u,v) = c(u,v) - f(u,v)$
- 通俗地讲：就是对于某一条边（也称弧），还能再有多少流量经过。
- G_f 残量网络 ,E_f 表示残量网络的边集 .

(a,b) 表示 （流量 f, 容量 c)

图 1 原网络

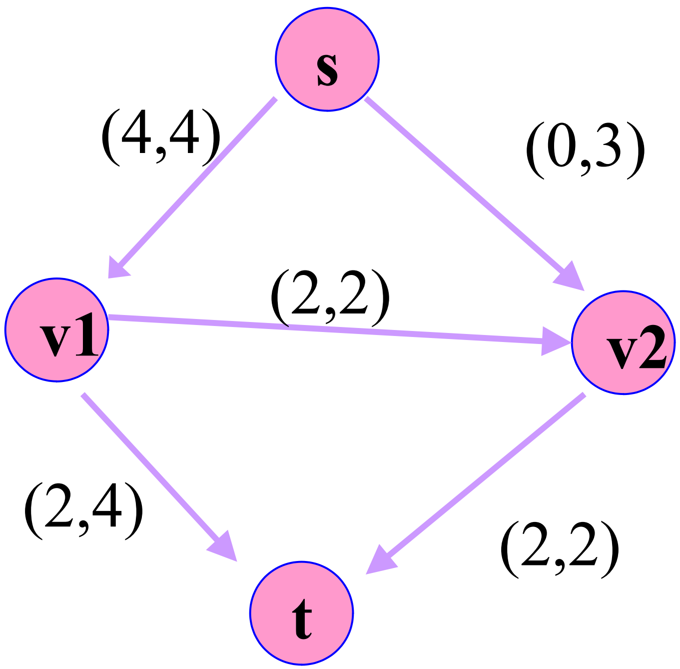
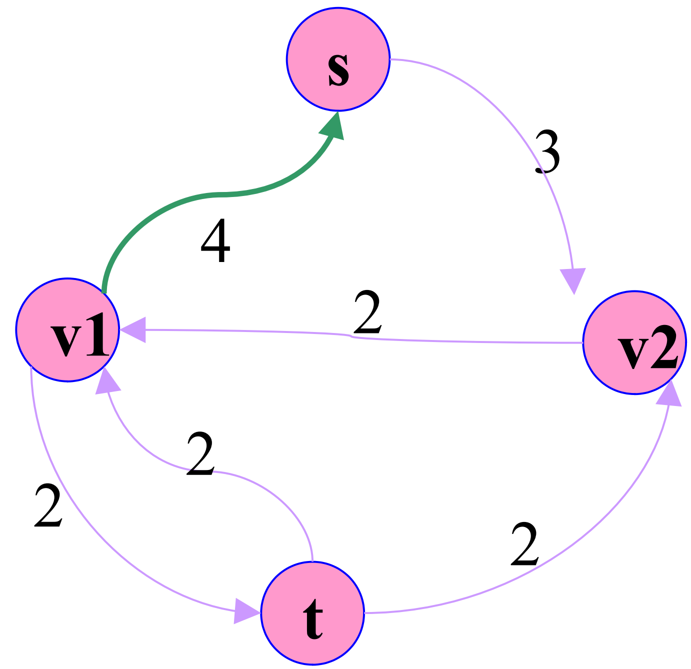
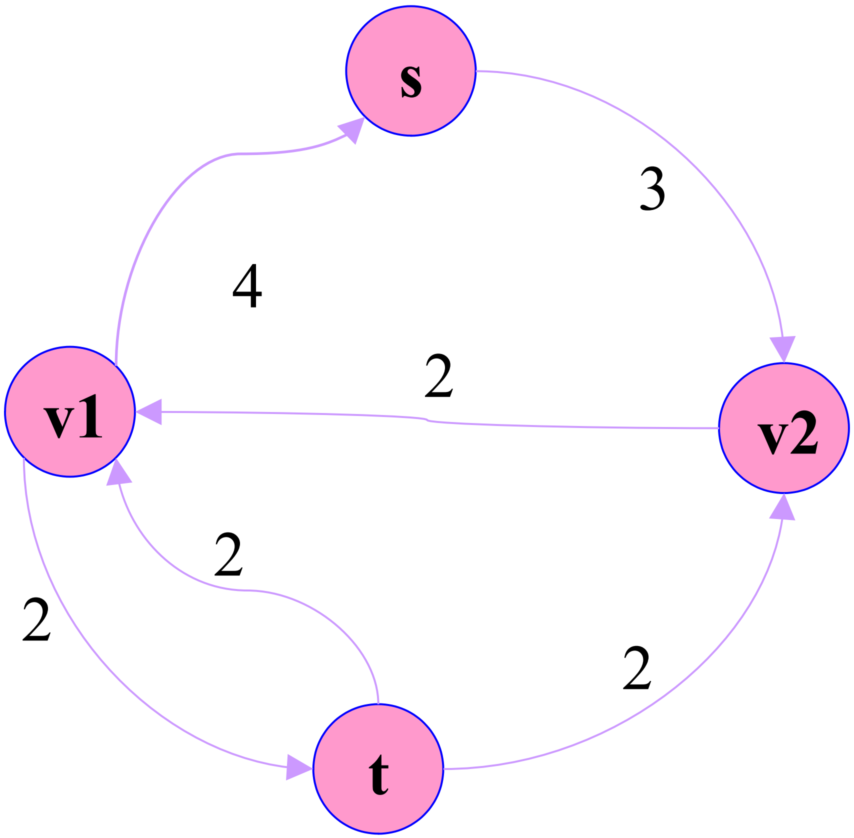


图 2 残量网络

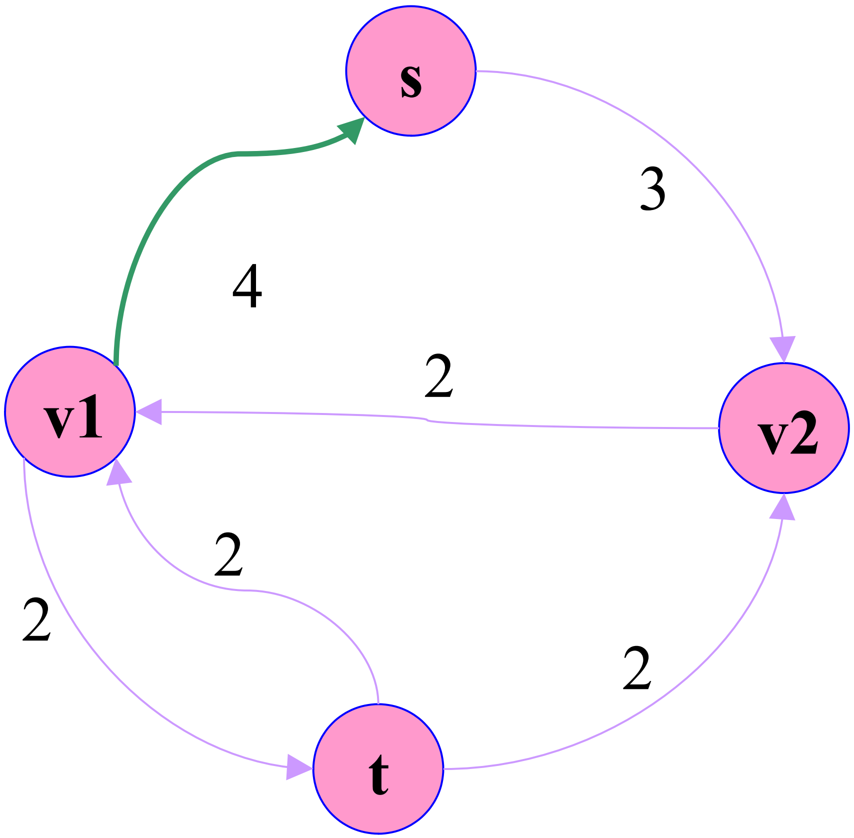


如果网络中一条边的容量为 0, 则认为这条边不在残量网络中。r(s,v1)=0, 所以就不画出来了。另外举个例子： r(v1,s) = c(v1,s) – f(v1,s) = 0 – (-f(s,v1)) = f(s,v1) = 4.



- 从残量网络中可以清楚地看到：
- 因为存在边 (s,v2) = 3, 我们知道从 S 到 v2 还可以再增加 3 单位的流量；
- 因为存在边 (v1,t) = 2, 我们知道从 v1 到 t 还可以再增加 2 单位的流量。

为什么要建立后向弧



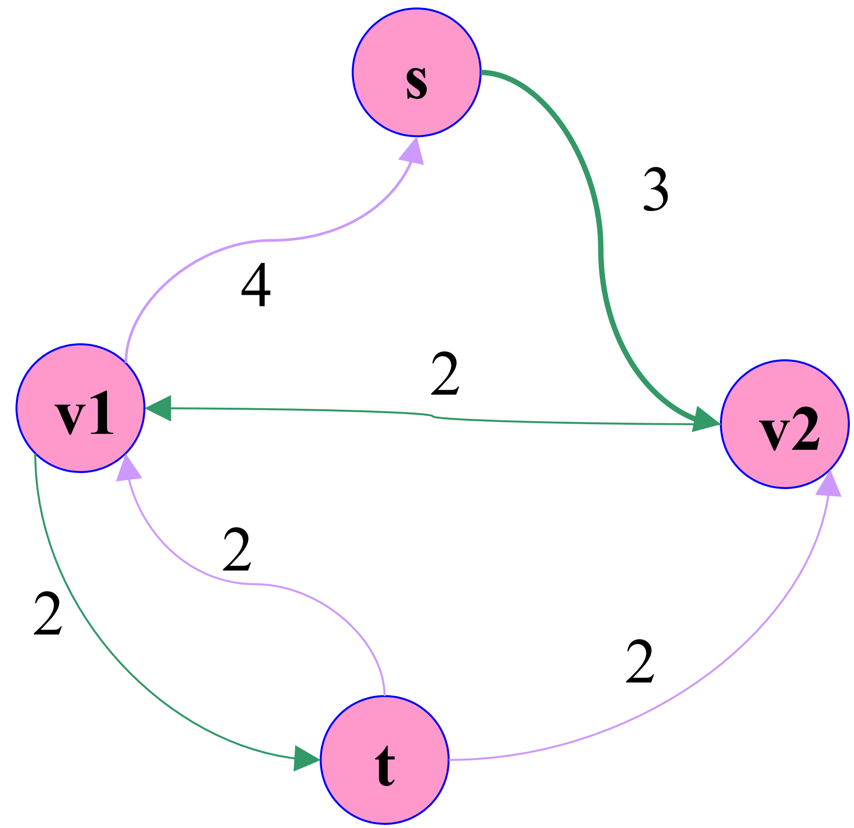
- 其中像 (v1,s) 这样的边称为后向弧，它表示从 v 1 到 s 还可以增加 4 单位的流量。
- 但是从 v1 到 s 不是和原网络中的弧的方向相反吗？显然“从 v1 到 s 还可以增加 4 单位流量”这条信息毫无意义。那么，有必要建立这些后向弧吗？

为什么要建立后向弧

- 显然，例 1 中的画出来的不是一个最大流。
- 但是，如果我们把 s -> v2 -> v1 -> t 这条路径经过的弧的流量都增加 2, 就得到了该网络的最大流。
- 注意到这条路径经过了一条后向弧 :(v2,v1) 。
- 如果不设立后向弧，算法就不能发现这条路径。
- 从本质上说，后向弧为算法纠正自己所犯的错误提供了可能性，它允许算法取消先前的错误的行为 (让 2 单位的流从 v1 流到 v2)

可改进路（增广路）

- **可改进路定义：在残量网络中的一条从 s 通往 t 的路径，其中任意一条弧 (u,v)，都有 r[u,v]>0 。（每一条前向弧都是非饱和弧，每一条后向弧都是非零流弧）**
- **绿色的即为一条可改进路。**



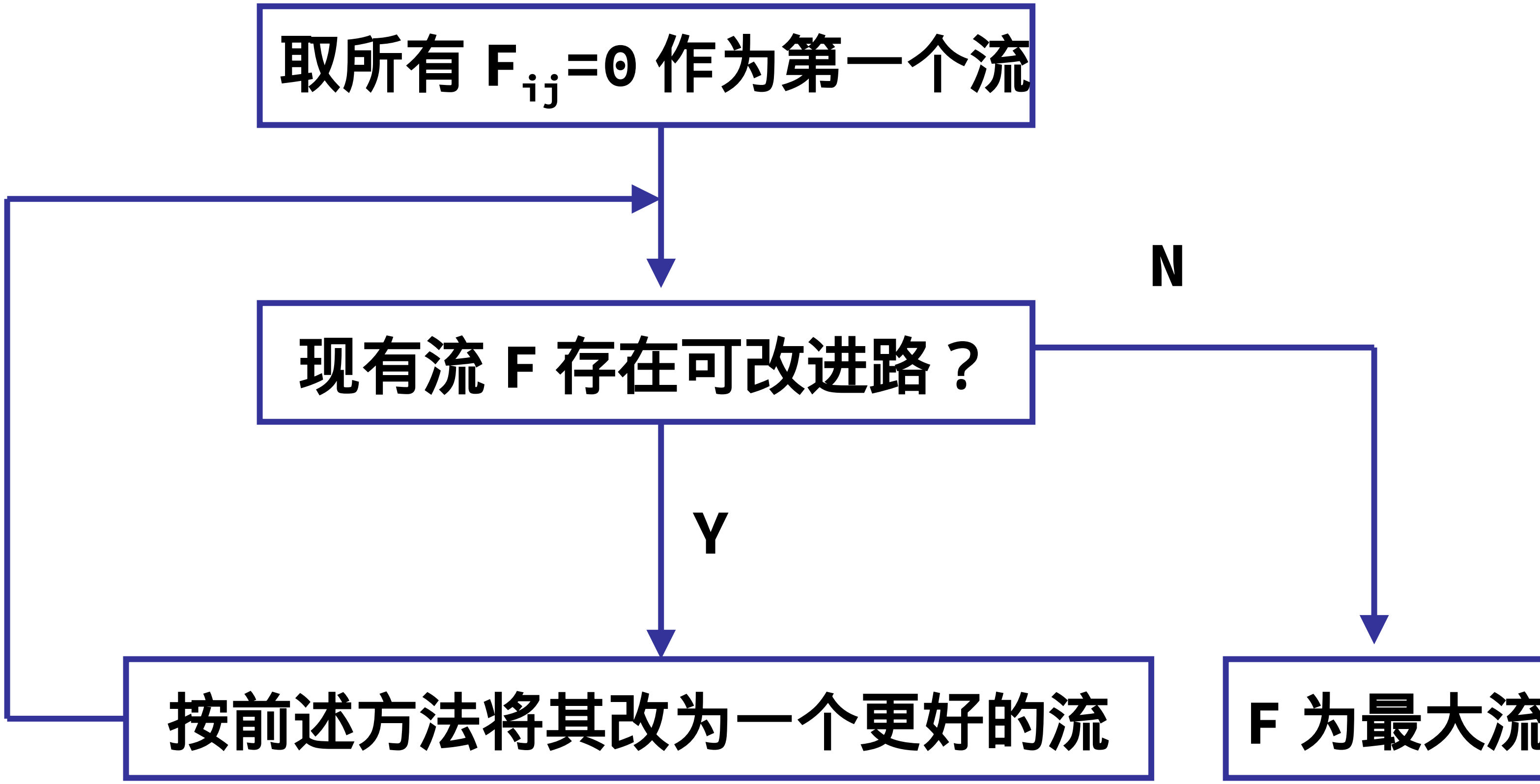
可改进路算法

- 可改进路算法：每次用 BFS 找一条可改进路，然后沿着这条路径修改流量值（实际修改的是残量网络的边权），使得总流量变得更大，修正的方法是：

  1、不属于可改进路 P 的弧一概不变

  2、对于可改进路 P 上的所有前向弧加上 a，后向弧减去 a，这里 a 是一个可改进量。a 既要尽量大，又要保证变化后 $0<=F_{ij}<=C_{ij}$（满足容量限制和平衡条件）。因此 a=min(min(C 前向弧 ij － F 前向弧 ij),min(F 后向弧 ij))。
- 如果不存在 Vs 到 Vt 的可改进路，算法停止，此时的流就是最大流。

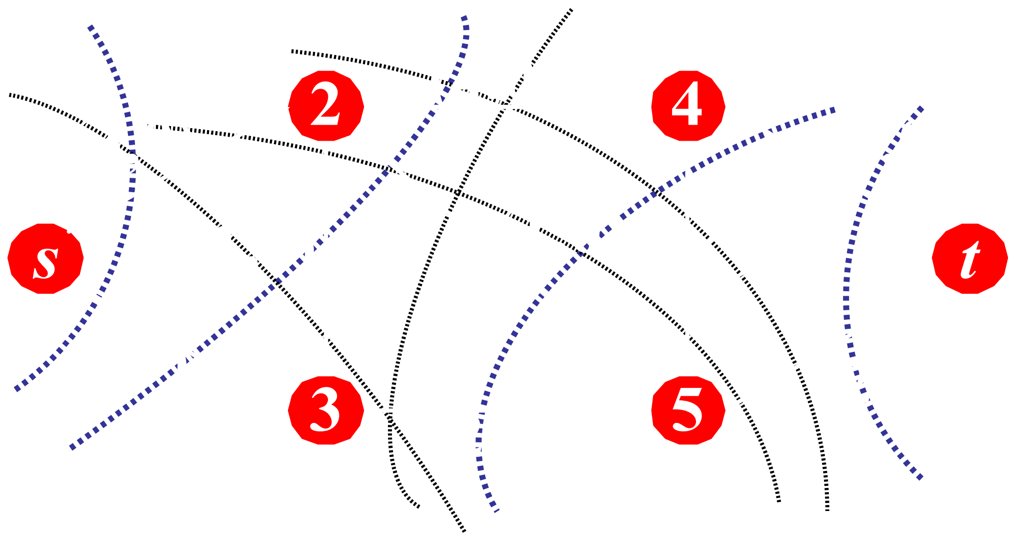
可改进路算法



如何很快找到可改进路？

截集的定义

- 一个截集 (S,T) 由两个点集 S,T 组成 .
- S+T = V
- s 属于 S.
- t 属于 T.



- 一种定义 S 的方法：

$$令V_s \in S, 若V_i \in S且F_{ij} < C_{ij}, 则令V_j \in S;$$

$$若V_i \in S且F_{ji} > 0, \quad 则令V_j \in S;$$

- 一旦 V_t 进入 S 集合，就表明找到一条可改进路；如果 S 集合扩展不下去而 V_t 又尚未进入 S 集合，则说明不存在可改进路，此时，除 S 外的顶点进入 T 集合。

最大流最小截定理

- **截集间的流量和 : f(S,T) =** $$\sum_{x\in S}\sum_{y\in T} f(x,y)$$
- **即 :S 中的任意一点与 T 中的任意一点组成的所有边上的流量之和 .(边的方向为从 S 中的节点到 T 中的节点)**
- **c,r 等函数都有类似的定义 .(截集的容量和、 截集的残量网络容量和)**

- **任一个网络 D 中从 V_s 到 V_t 的最大流的流量等于分离 Vs 和 Vt 的最小截集的容量。**

最大流等价条件

- **网络流中这三个条件等价**（在同一个时刻）：
- 1、 f 是最大流
- 2、残量网络中找不到增广路径
- 3、 $|f| = c(S,T)$

结论 1

- 1.f(X,X) = 0　　　（由流量反对称性）
- 2. f(X,Y) = -f(Y,X)　　　（由流量反对称性）
- 3.f(X ∪ Y,Z) = f(X,Z) + f(Y,Z)　（显然）
- 4.f(X,Y ∪ Z) = f(X,Y) + f(X,Z)　（显然）

1、 f 是最大流
2、残量网络中找不到增广路径
3、 |f| = c(S,T)

- 1 -> 2 证明：显然 . 假设有增广路径 , 由于增广路径的容量至少为 1, 所以用这个增广路径增广过后的流的流量肯定要比 f 的大 , 这与 f 是最大流矛盾 .

结论 2(点集总流量为零)

- 不包含 s 和 t 的点集 , 与它相关联的边上的流量之和为 0.
- 证明 : f(X,V) = $\sum_{x\in X}[\sum_{v\in V} f(x,v)]$

  = $\sum_{x\in X}[0]$ (由流量平衡)

  = 0

结论 3

- **任意割的流量等于整个网络的流量.**
- **证明:**
- **$f(S,T) = f(S,V) - f(S,S)$ (由辅助定理 1)**

$= f(S,V)$ （由辅助定理 1)

$= f(S,V) + f(S - s,V)$ （同上）

$= f(s,V)$ （由辅助定理 2)

$= |f|$ （由 $|f|$ 的定义）

结论 4

- 网络的流量小于等于任意一个割的*容量*.(注意这个与辅助定理 3 的区别 . 这里是容量)
- 即 |f| <= c(S,T)
- 证明： |f| = f(S,T) = $\sum_{x\in S}\sum_{y\in T} f(x,y)$ (由定义)

  <= $\sum_{x\in S}\sum_{y\in T} c(x,y)$ (由流量限制)

  = c(S,T)

1、 f 是最大流
2、残量网络中找不到增广路径
3、 |f| = c(S,T)

- 2 -> 3 证明：定义 S = s ∪ {v | 在残量网络中 s 到 v 有一条路径 } ; T = V- S. 则 (S,T) 是一个割 .
- | f | = f(S,T) (由辅助定理 3)
- 而且 ,r(S,T) = 0. 假设不为 0, 则在残量网络中，两个集合间必定有边相连 , 设在 S 的一端为 v, 在 T 的一端为 u. 那么 ,s 就可以通过 v 到达 u, 那么根据 S 的定义 ,u 就应该在 S 中 . 矛盾 .

所以 , | f | = f(S,T) = c(S,T) – r(S,T) = c(S,T)

1、 f 是最大流
2、残量网络中找不到增广路径
3、 |f| = c(S,T)

- 3 -> 1 证明：

  |f| <= c(S,T) （辅助定理 4)

  因为我们已经有 |f| = c(S,T), 如果最大流的流量是 |f|+d (d>0), 那么 |f|+d 肯定不能满足上面的条件 .

标号法寻求可改进路（ Ford-Fulkerson 算法）

从一个可行流 F 出发（可以设为零流），经过标号过程和调整过程。

标号过程：

网络中的顶点或者是标号点(分为已检查和未检查两种)，或者是未标号点，每个标号点分为两部分(标号从哪个顶点得到，确定可改进量 a)。

标号过程开始，总先给 V_s 标上 (0, +∞)，这时是标号而未检查的顶点，其余都是未标号点。取一个标号而未检查的标号 V_i ，对一切未标号点 V_j

(1) 若在弧(V_i, V_j)上$F_{ij}<C_{ij}$,则给V_j标号$(V_i, L(V_j))$，这里$L(V_j)=\min[L(V_i), C_{ij}-F_{ij}]$。这时$V_j$成为标号未检查的顶点。

(2) 若在弧(V_j, V_i)上$F_{ij}>0$,则给V_j标号$(-V_i, L(V_j))$，这里$L(V_j)=\min[L(V_i), F_{ij}]$。这时$V_j$成为标号未检查的顶点。

在 V_i 的全部相邻顶点都已标号后， V_i 成为标号而已检查过的顶点。重复上述步骤，一旦 V_t 被标上号，表明得到一条从 V_s 到 V_t 的可改进路 P，转入调整过程。

标号法寻求可改进路（ Ford-Fulkerson 算法）

调整过程：

采用“倒向追踪”的方法，从 Vt 开始，利用第一个标号找出可改进路 P ，并以 Vt 的第二个标号 L (Vt) 作为改进量 a ，改进 P 上的流量。

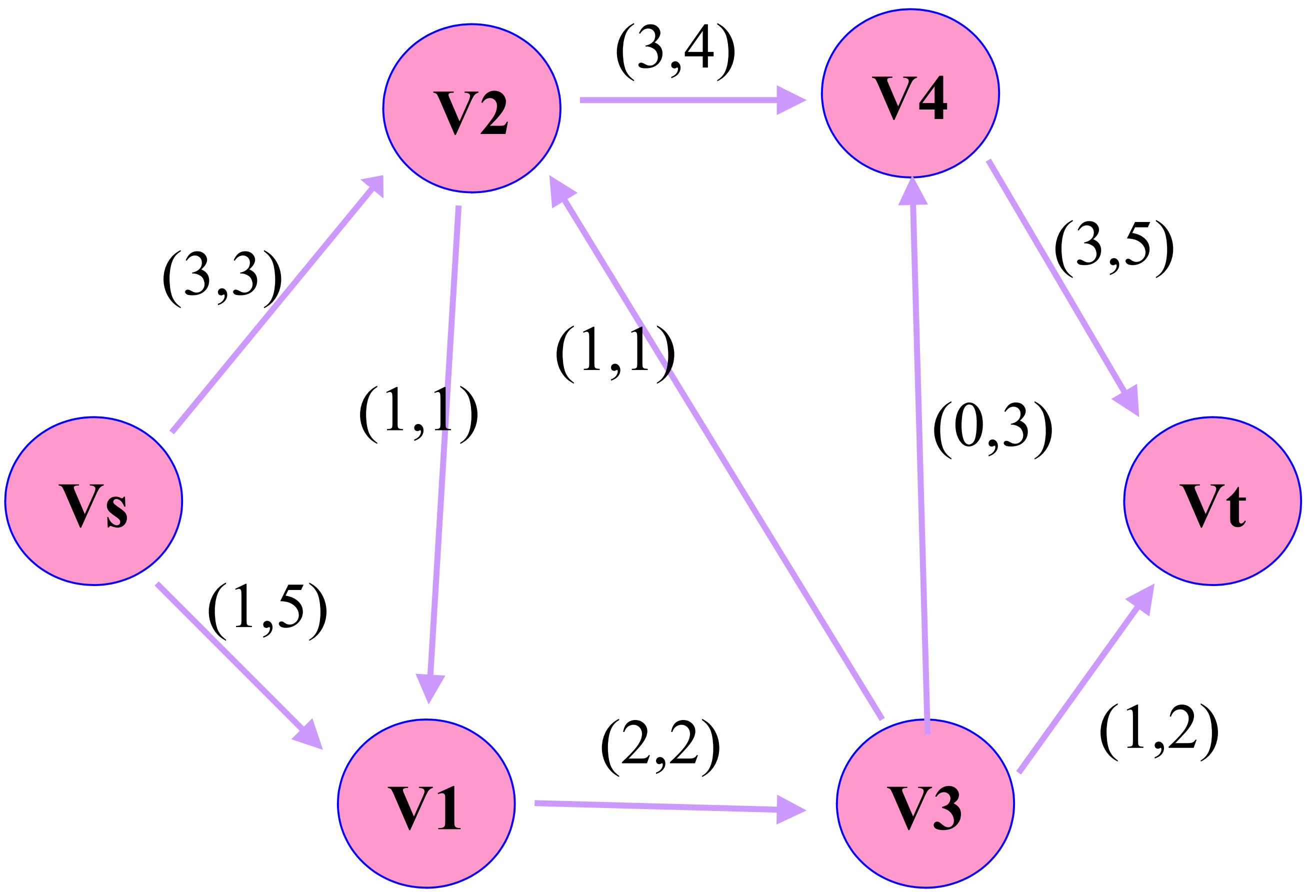
例如：设 Vt 的第一个标号为 Vk(或 -Vk) ，则弧 (Vk,Vt)(或 (Vt,Vk)) 是 P 上的弧，接下来再检查 Vk 的第一个标号，如此继续下去 ... ，直到查倒 Vs 为止。这时被找出的弧构成了 P 。

令改进量 a=L(Vt) ，即 Vt 的第二个标号。

$$F_{ij}' = \begin{cases} F_{ij} + a & (\mathrm{V_i},\mathrm{V_j}) \in \mathrm{P}^{+} \\ F_{ij} - a & (\mathrm{V_i},\mathrm{V_j}) \in \mathrm{P}^{-} \\ F_{ij} & (\mathrm{V_i},\mathrm{V_j}) \notin \mathrm{P} \end{cases}$$

去掉所有的标号，对新的可行流 F_{ij}'，重新进入标号过程。直到标号过程无法继续。

例 1 求如下网络的最大流



标号法分析例 1

1. 标号过程

(1) 首先给 Vs 标上 (0,+∞)

(2) 检查 Vs 。弧 (Vs,V2) 上， Fs2=Cs2=3 ，不满足标号条件；弧 (Vs,V1) 上， Fs1=1<Cs1=5 ，则 V1 的标号为 (Vs,L(V1)) ，其中 L(V1)=min[L(Vs),(Cs1-Fs1)]=min[+∞,5-1]=4

(3) 检查 V1 。弧 (V1,V3) 上， F13=C13=2 ，不满足标号条件；弧 (V2,V1) 上， F21=1>0 ，则给 V2 记下标号为 (-V1,L(V2)) ，其中 L(V2) ＝ min[L(V1),F21]=min[4,1]=1

标号法分析例 1

(4) 检查 V2 。弧 (V2,V4) 上， F24=3<C24=4 ，则给 V4 标号 (V2,L(V4)) ，其中 L(V4)=min[L(V2),C24-F24]=min[1,1]=1

同理，标注 V3 为 (-V2,1).

(5) 在 V3 ， V4 中任选一个进行检查，如 V4 。弧 (V4,Vt) 上， F4t=3<C4t=5 ，则给 Vt 标号 (V4,L(Vt)) ，其中 L(Vt)=min[L(V4),C4t-F4t]=min[1,2]=1

(6) Vt 有了标号，转入调整过程。

标号法分析例 1

2. 调整过程

(1) 如图为按照标号的第一个顶点找到的一个可改进路

(2) 前向弧集合

$$P^+ = \{(V_s, V_1), (V_3, V_t)\}$$

(3) 后向弧集合

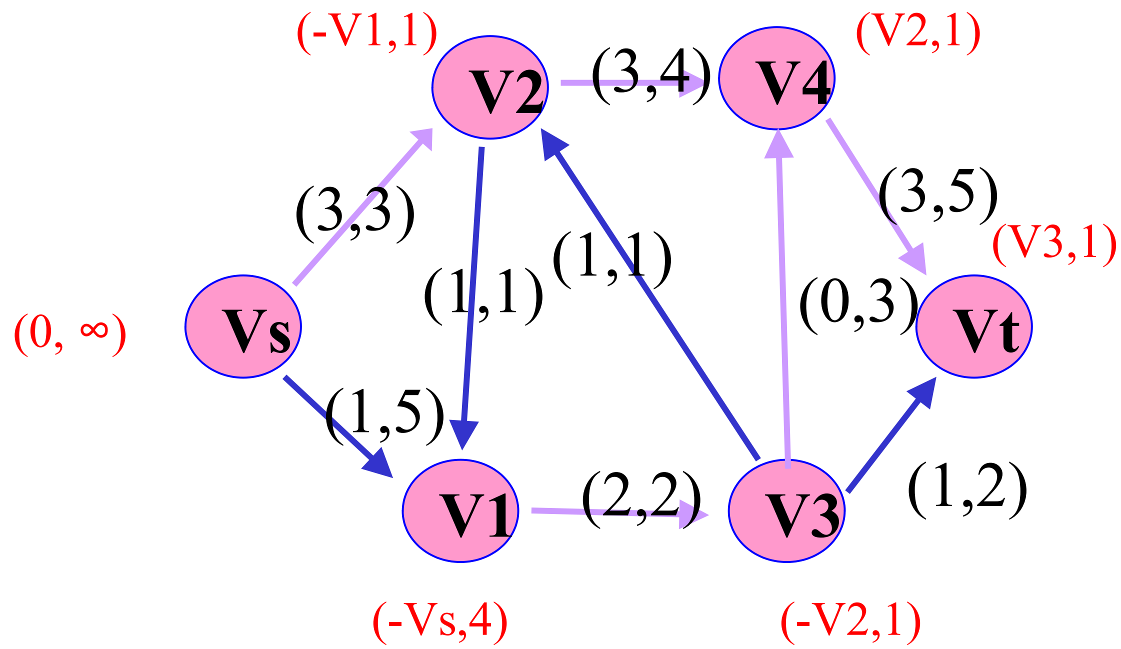
$$P^- = \{(V_2, V_1), (V_3, V_2)\}$$

(4) 按 a = 1 在 P 上调整

$$P^+ : F_{s1}' = F_{s1} + a = 2, F_{3t}' = F_{3t} + a = 2$$

$$P^- : F_{21}' = F_{21} - a = 0, F_{32}' = F_{32} - a = 0$$

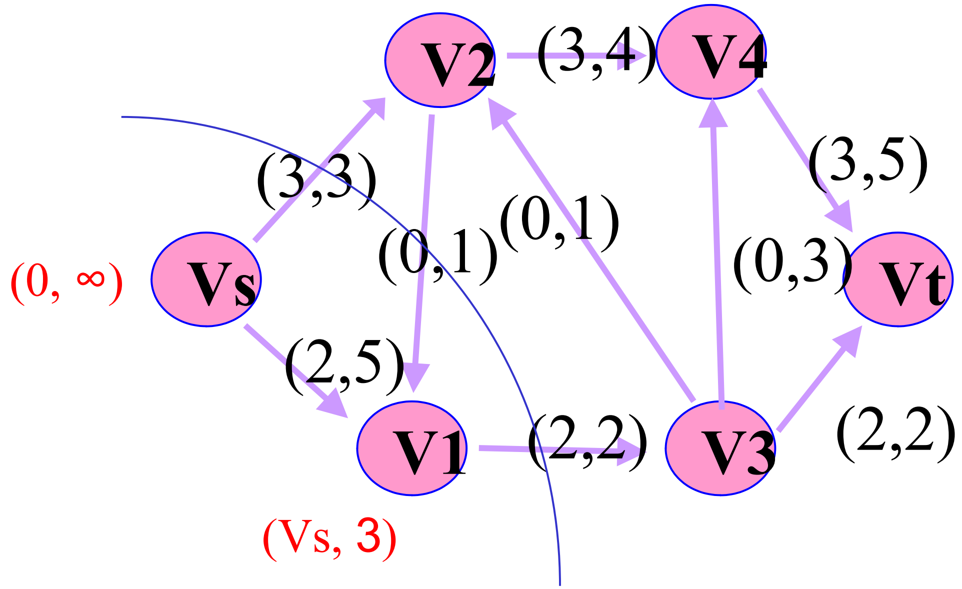
(5) 对调整后的图重新进入标号过程



标号法分析例 1

- 3. 重新开始标号过程

 当算法进行到检查 V1 时 F21=0， F13=C13 均不符合条件，标号无法继续，这时的可行流为最大流 5 。



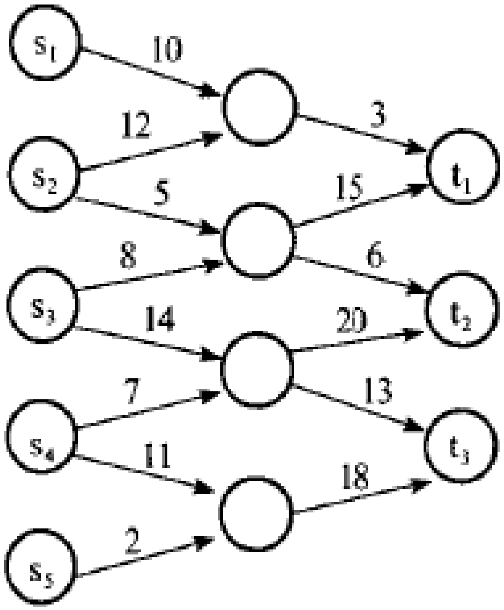
同时找到最小截集 (S,T)

S={Vs,V1},T={V2,V3,V4,Vt}

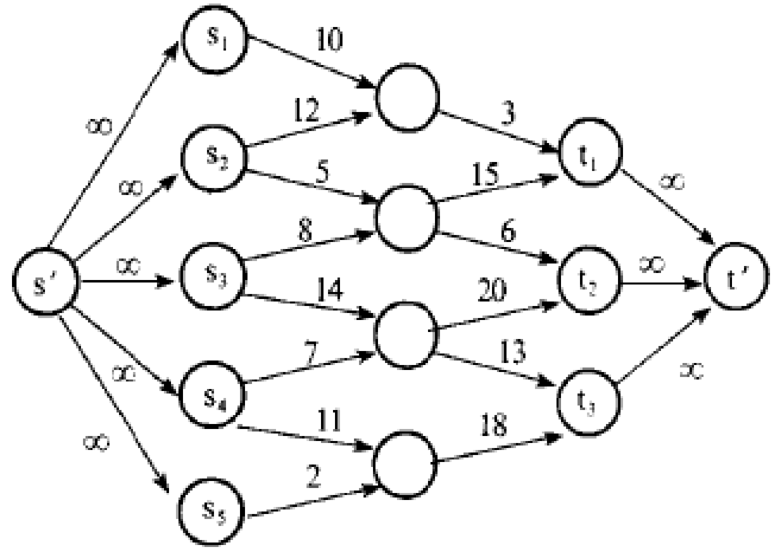
C(S,T)=5

这也验证了最大流最小截定理的正确性

多源多汇网络的最大流



(a)



(b)

- 将其转换为单源单汇的问题

容量有上下界的网络的最大流

网络中的每条弧 e 对应两个数字 B(e) 和 C(e)，分别标识弧容量的上界和下界，那么如何求满足条件 $B(e) \le F(e) \le C(e)$ 的最大流？

标号法中的网络是有上下界网络的一种特例，即 B(e)=0 。但当 B(e)>0 时有上下界的网络不一定存在可行流了。

$$s \xrightarrow{0,\ 1} a \xrightarrow{2,\ 3} t$$

那么如何判断一个有上下界的网络有无可行流呢？

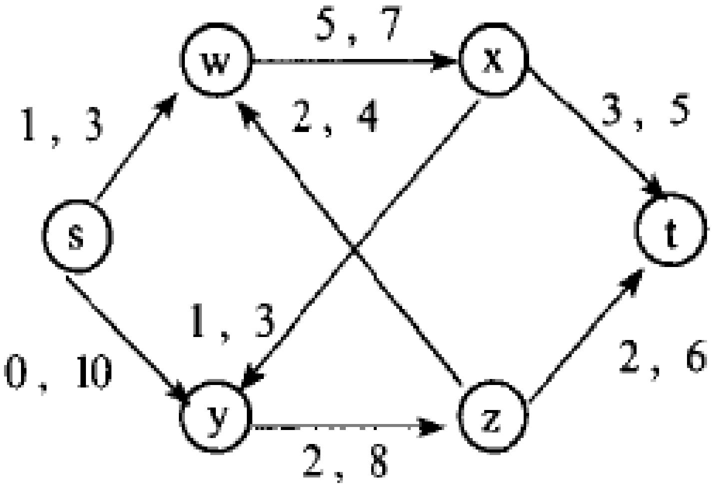
容量有上下界的网络的最大流

算法思路：将原网络转换为一个附加网络。

1. 新增加两个顶点 $\bar{s}$ 和 $\bar{t}$，分别称为附加源和附加汇
2. 对原网络 N 的每个顶点 U 加一条新弧 $e = U\bar{t}$ ，这条弧的容量为以 U 为尾的弧的容量下限之和。
3. 对原网络 N 的每个顶点 U 加一条新弧 $e = \bar{s}U$ ，这条弧的容量为以 U 为头的弧的容量下限之和。
4. 原网络的弧仍保留，弧容量修正为 C(e) -B(e)
5. 再添两条新弧 e=st,e′=ts 。其容量均为∞
6. 用标号法求此新网络的最大流，若结果能使流出 $\bar{s}$ 的一切弧都满载，则原网络有可行流 F(e) = F(e)′ + B(e) ，否则无可行流。
7. 在原网络中运用标号法将可行流放大，从而得出最大流。

容量有上下界的网络的最大流

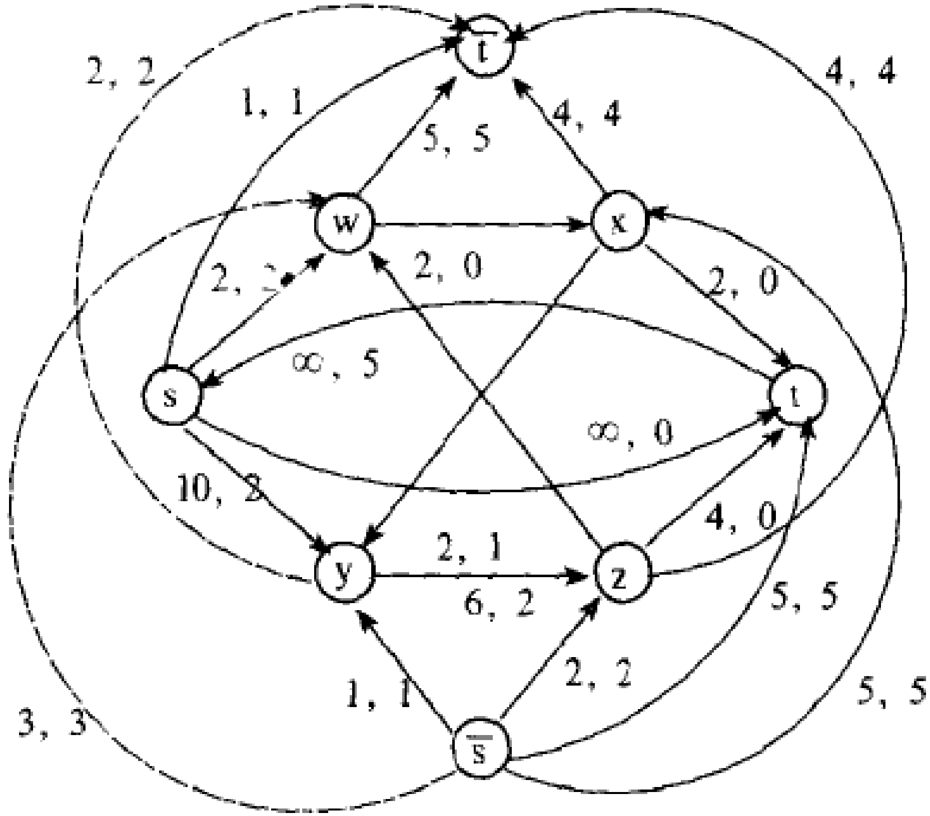
例原网络，第一个数字为 B(e)，第二个为 C(e)



容量有上下界的网络的最大流

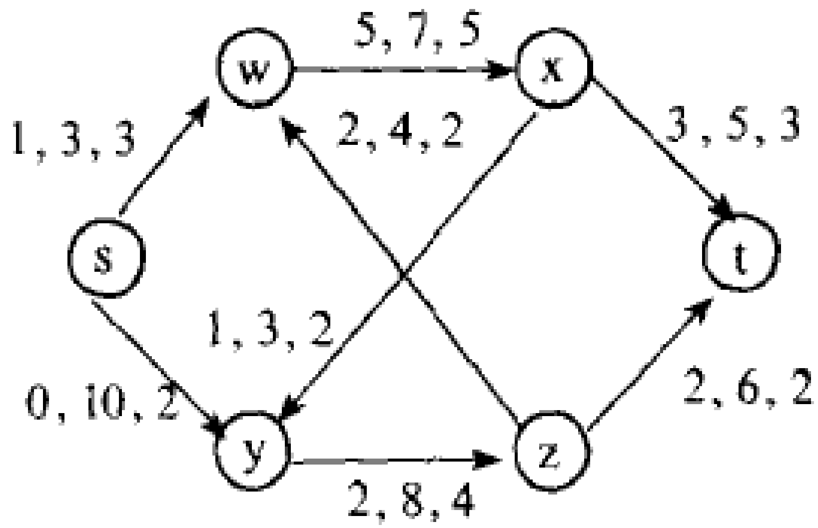
按照上述算法求得附加网络，并用标号法求此网络的最大流。

发现从 $\bar{s}$ 流出的流都满载，所以有可行流。



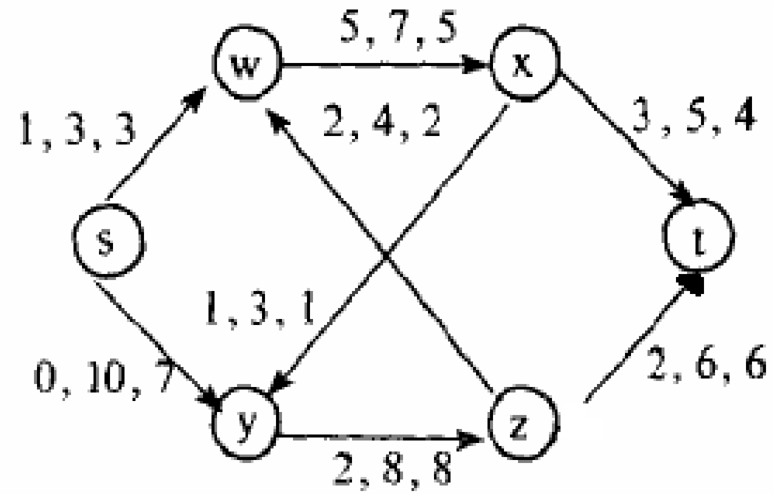
容量有上下界的网络的最大流

按照 F(e) = F(e)’ + B(e) 将此最大流还原为原网络的最大流



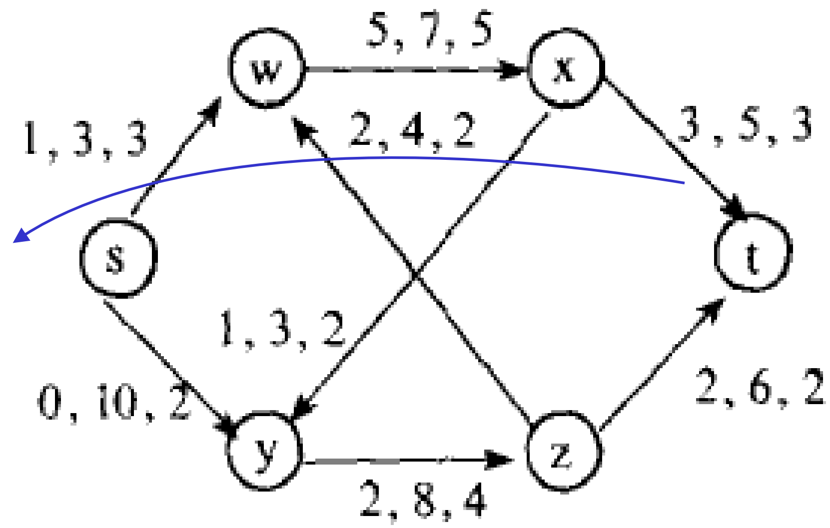
容量有上下界的网络的最大流

用标号法进行可行流放大，得到最大流 10



容量有上下界的网络的最小流

- 按照前述附加网络方法求得可行流，只不过在放大可行流时，以 t 为源点， s 为汇点进行，倒向求出的最大流为从 s 到 t 到的最小流。

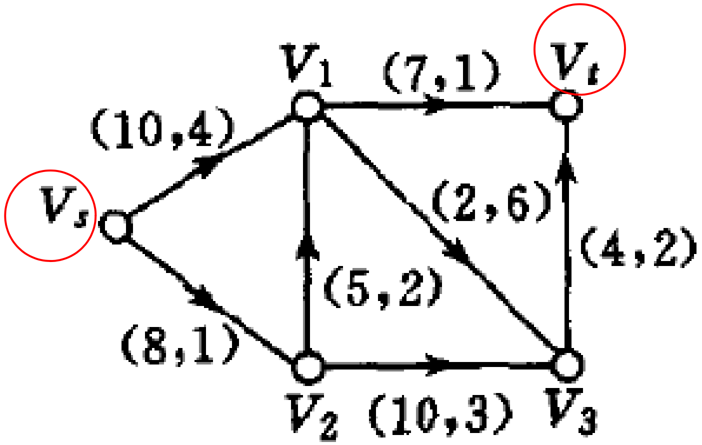


最小费用最大流问题

- **求运输量最大且费用最少的运输方案**

$$B(F) = \sum B_{ij} * F_{ij}$$

- **求一个最大流，使得此式取最小值**



最小费用最大流问题

算法思想：

若 F 是所有可行流中费用最小的，而 P 是关于 F 的所有可改进路中费用最小的，沿着 P 取调整 F 最大，则得到的流为最小费用最大流。

最小费用最大流问题

步骤：

1. 取 F(0)=0;
2. 若第 k-1 步得到最小费用流 F(k-1)，则构造赋权有向图 W(F(k-1))，在 W(F(k-1)) 中，寻求从 Vs 到 Vt 的最短路径。若不存在最短路 (即最短路为 +∞)，则 F(k-1) 为最小费用最大流；若存在最短路，则为可改进路 P，在 P 上对 F(k-1) 进行调整。
3. 其中，赋权有向图 W(F(k)) 的构造规则为：其顶点是原网络的顶点，而每条弧变为两个方向相反的弧，定义权值 W_{ij} 为

$$W_{ij}=\begin{cases} B_{ij} & 若\ F_{ij}<C_{ij} \\ +\infty & 若\ F_{ij}=C_{ij} \end{cases}$$

$$W_{ji}=\begin{cases} -B_{ij} & 若\ F_{ij}>0 \\ +\infty & 若\ F_{ij}=0 \end{cases}$$

（长度为 ∞ 的可以从 $W(F)$ 中略去）

最小费用最大流问题

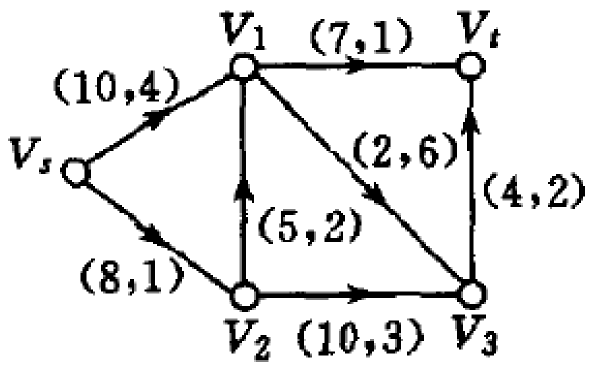
4. 在 P 上对 F(k-1) 进行调整的方法为：

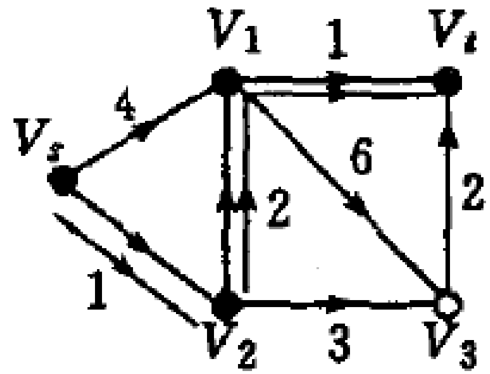
$$a = \min[\min_{P+}(C_{ij} - F_{ij}(k-1)), \min_{P-} F_{ij}(k-1)]$$

$$F_{ij}(k) = \begin{cases} F_{ij}(k-1) + a & (V_i,V_j) \in P+ \\ F_{ij}(k-1) - a & (V_i,V_j) \in P- \\ F_{ij}(k-1) & (V_i,V_j) \notin P \end{cases}$$

5. 得到新流 F(k)，再对 F(k) 重复上述步骤，直到不存在最短路径。

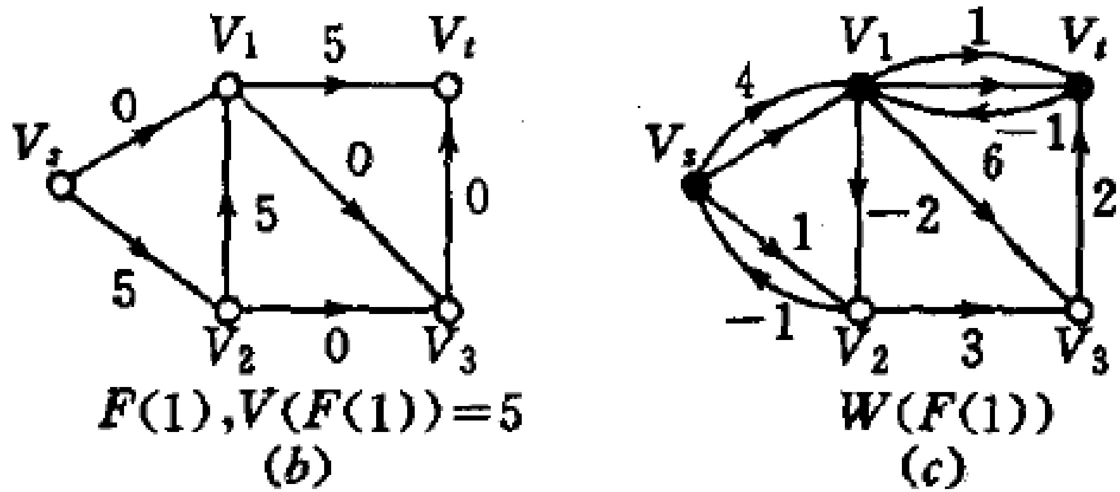
最小费用最大流问题—例



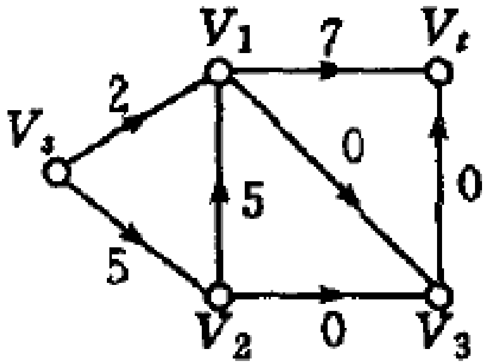


W(F(0))
(a)

最小费用最大流问题—例

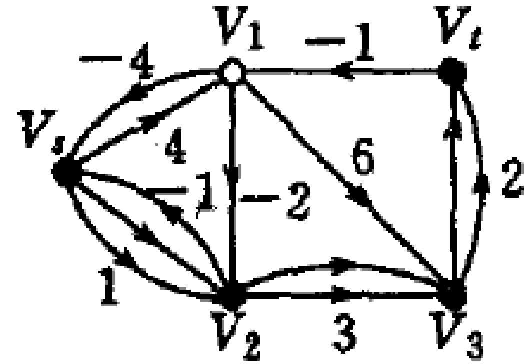


最小费用最大流问题一例



$F(2), V(F(2))=7$

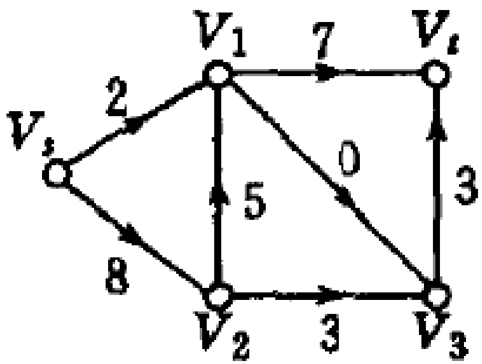
(d)



$W(F(2))$

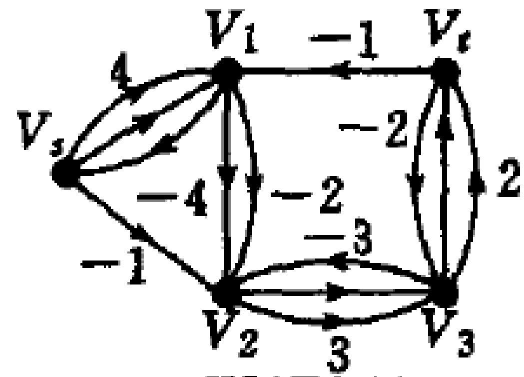
(e)

最小费用最大流问题一例



$F(3), V(F(3))=10$

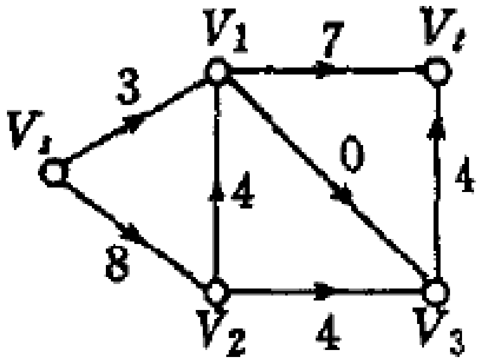
(f)



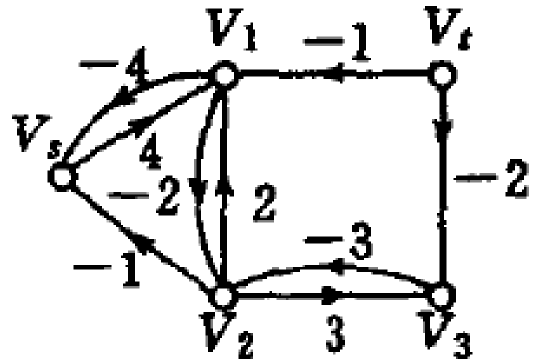
$W(F(3))$

(g)

最小费用最大流问题一例



$F(4), V(F(4))=11$
(h)



$W(F(4))$
(i)

作业

- 本课算法的实现
- 标号法
- 容量有上下界网络的最大流
- 容量有上下界网络的最小流
- 最小费用最大流