第八章

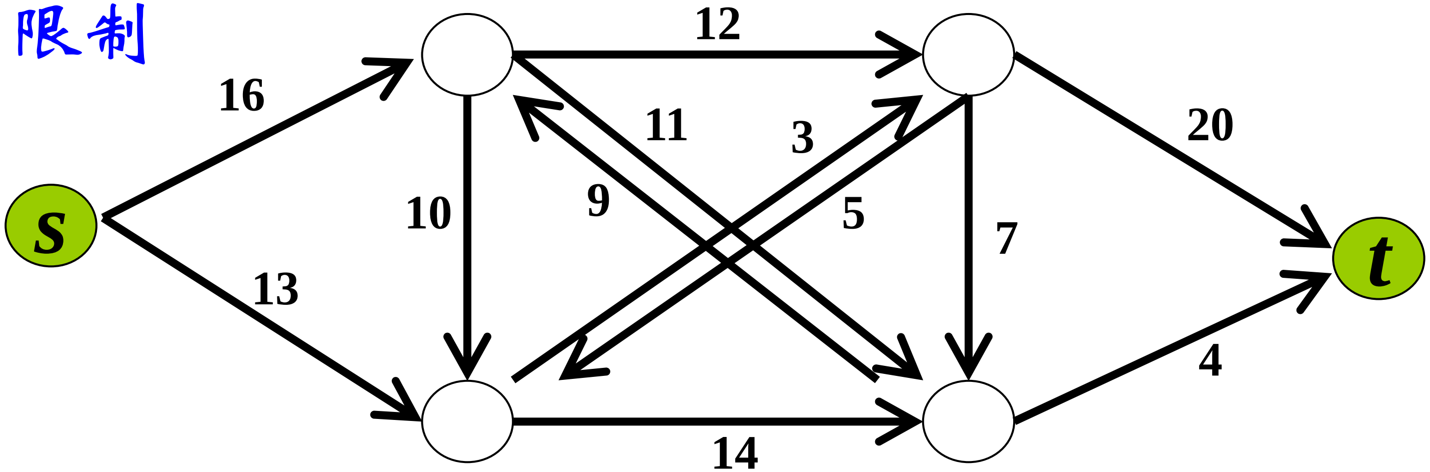
流网络算法

- 8.1 基本概念与问题定义
- 8.2 Ford-Fulkerson方法

8.1 基本概念与问题定义

应用背景

- 很多实际问题可以建模为流网络
  - 装配线上物件的流动
  - 电网中电流的流动
  - 通信网络中信息的流动
  - 道路交通网络中的运输活动
  - ……
- 一个源节点s、一个汇点t，由源节点流向汇点
  - 流量守恒
  - 容量限制



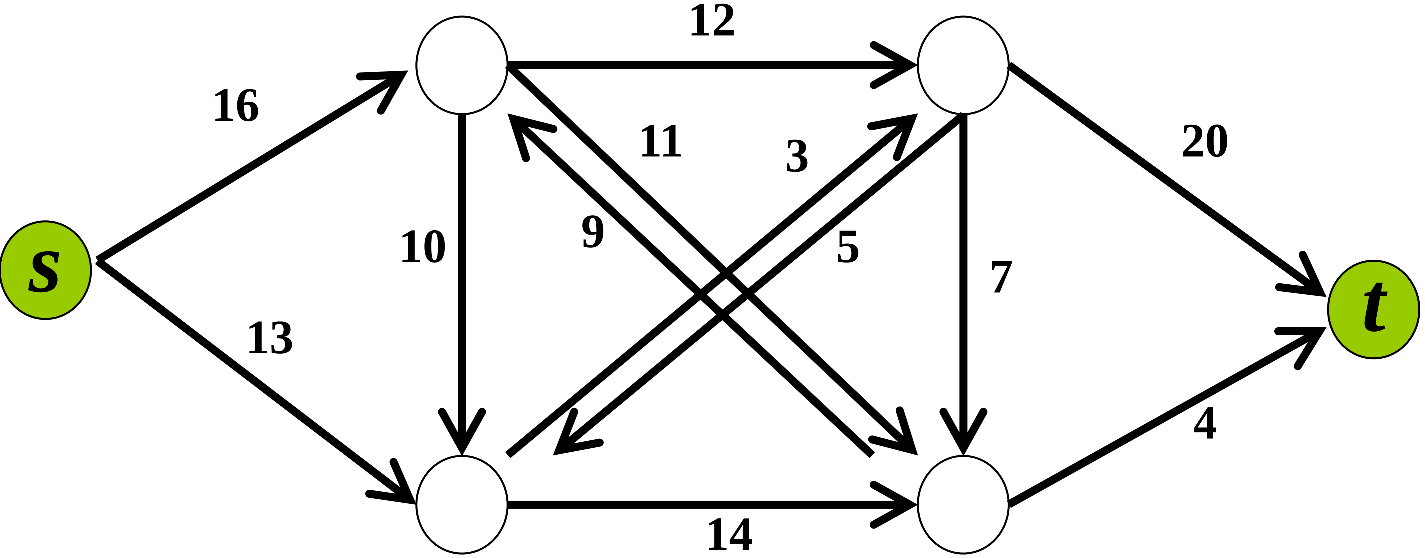
- 流网络

  是一个无自环的有向图$G=(V, E)$，

  (1) 图中的每条边$(u, v)\in E$有一个非负的容量值$c(u, v)\geq 0$。
  if $(u, v)\notin E$ $c(u, v)=0$；

  (2) 有两个特殊结点$s, t\in V$，s称为源结点(source)，t称为汇点(sink)

  (3) For $\forall v\in V$，存在一个s到t经过v的路径$s\Rightarrow v\Rightarrow t$.



流网络是连通的

除源结点外，每个结点都至少有一条进入的边，所以$|E|\geq|V|-1$

基本概念

- 流(Flow)

设$G(V,E)$是一个流网络，c是容量函数，s源结点，t是汇点。
G中的流是一个实值函数f：$V\times V\rightarrow R$，满足下列性质：

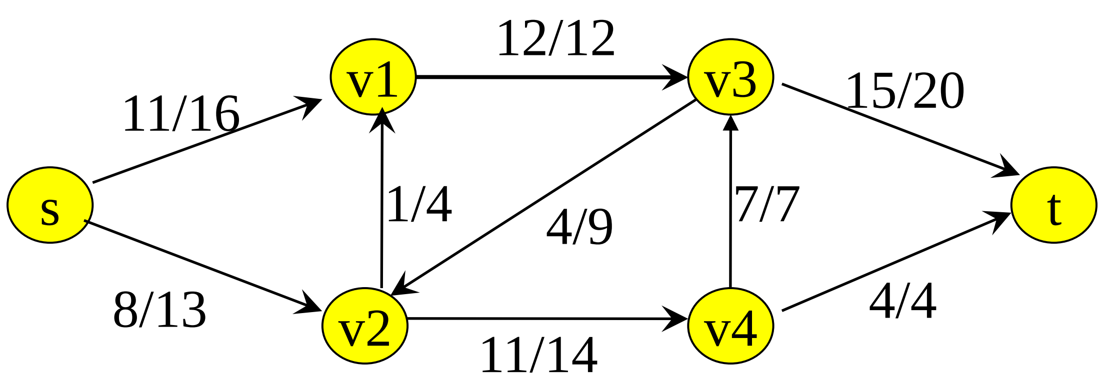
(1) 容量约束：$\forall u,v\in V$, $0\le f(u,v)\le c(u,v)$；

(2) 流量守恒：$\forall u\in V-\{s,t\}$, 有

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

称$f(u,v)$为从结点u到结点v的流

当$(u,v)\notin E$时，从结点u到v之间没有流，因此$f(u,v)=0$.



一个流f的值$|f|$定义为：

$$\sum_{v\in V} f(s,v)-\sum_{v\in V} f(v,s)$$

基本概念

- 流(Flow)

设$G(V,E)$是一个流网络，c是容量函数，s源结点，t是汇点。G中的流是一个实值函数f: $V\times V\rightarrow R$，满足下列性质：

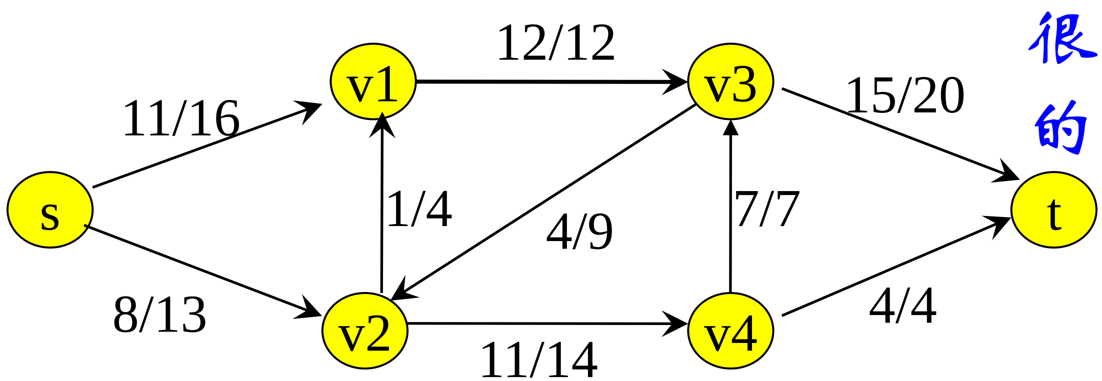
(1) 容量约束：$\forall u,v\in V$, $0 \le f(u, v) \le c(u, v)$；

(2) 流量守恒：$\forall u\in V-\{s,t\}$, 有

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

称$f(u, v)$为从结点u到结点v的流

当$(u, v)\notin E$时，从结点u到v之间没有流，因此$f(u, v)=0$.

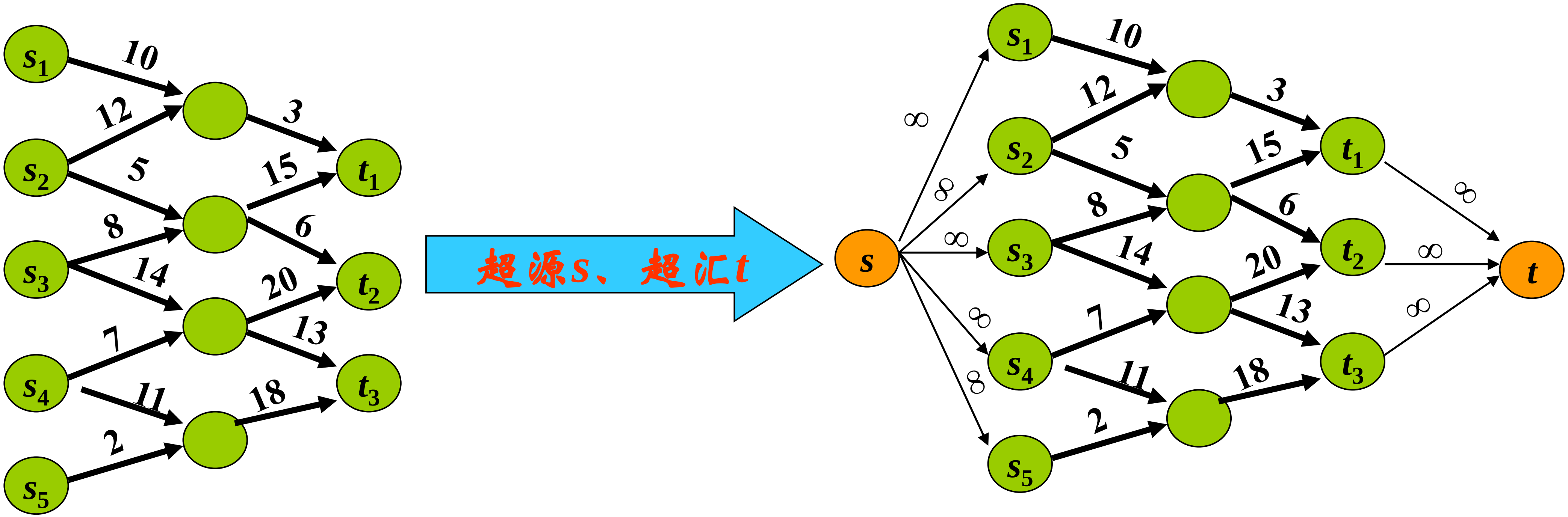


很多流网络中没有进入source的边，一个流f的值$|f|$定义为：

$$\sum_{v\in V} f(s,v)$$

基本概念

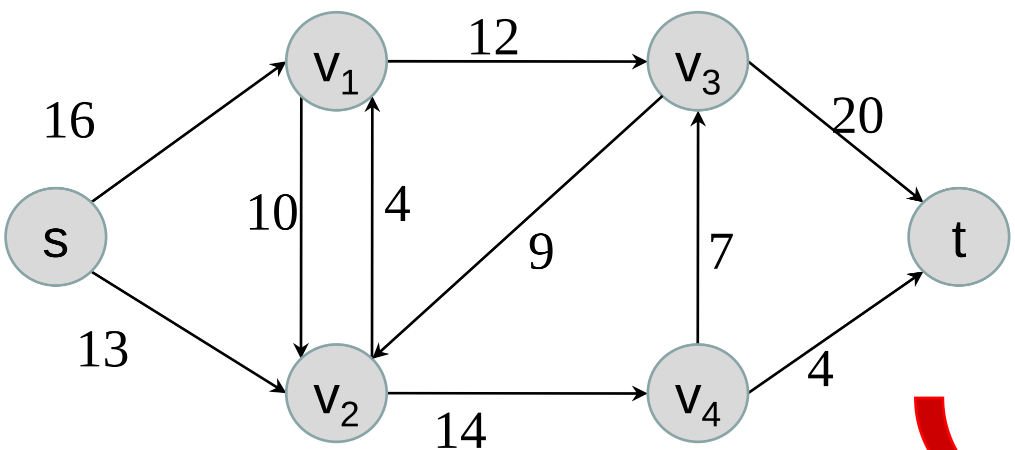
- 多源多汇的网络

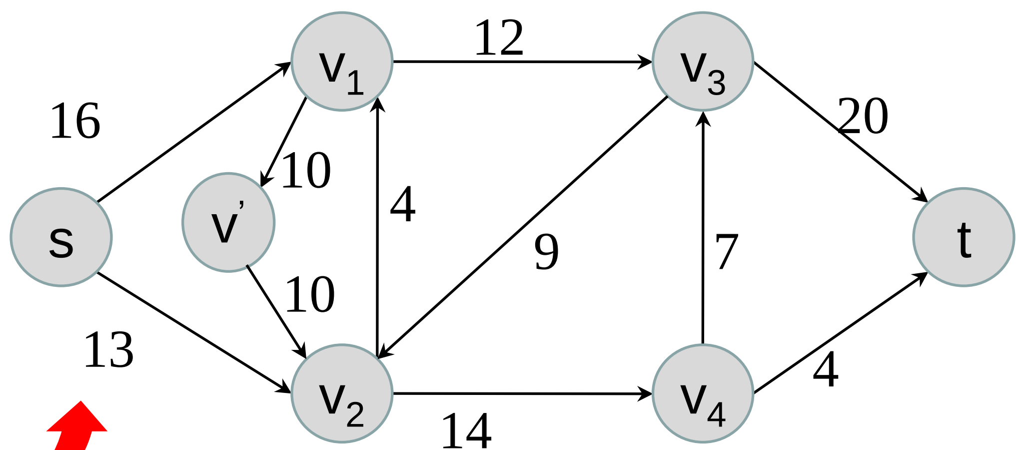


只需讨论单源单汇的网络流

一些假设

- 单源结点、单汇点流网络
- 假设：流网络中无反向边
  - 给定有向图$G=(V, E)$，如果边$(u, v)\in E$，则边$(v, u)\notin E$





最大流问题

- 问题定义
  - 输入：流网络$G=(V, E)$
  - 输出：具有最大流值的流f

直观想法

- 循环递进
  - 初始：网络上的流为0
  - 找出一条从s到t的路径p和正数a，使得p上的每一条边(u,v)的流量增加a之后仍能够满足容量约束：$f(u,v)+a \le c(u,v)$

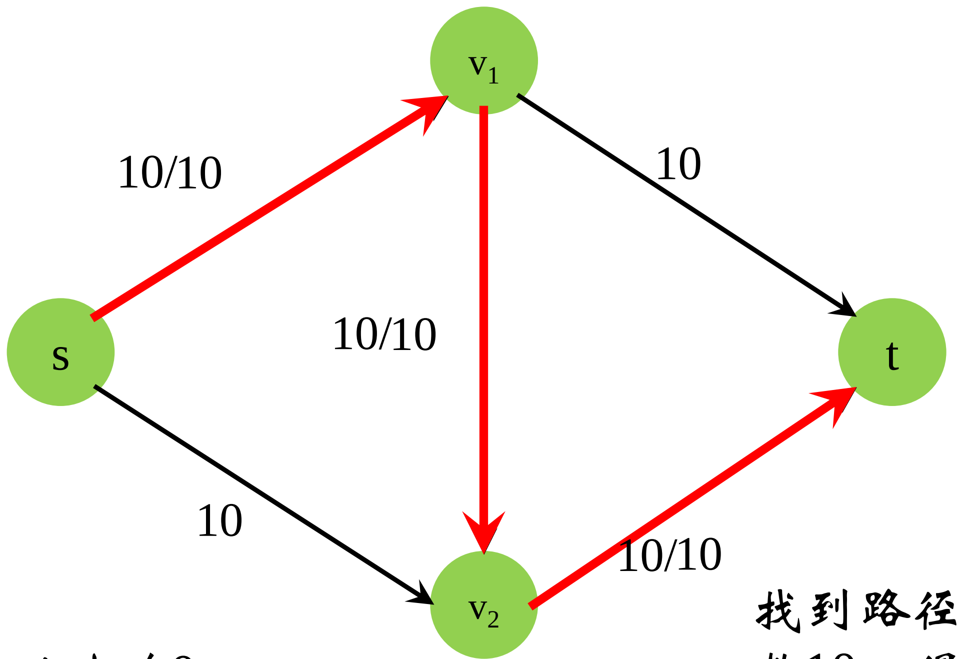
    //将p上的每条边的流量增加a，得到一个更大的流
  - 重复执行第二步，直到找不到满足约束条件的路径.

关键在于：

1. 如何找路径p，以便得到更大的流?
2. 如何判断循环结束的条件?

即：当循环结束时，所得到的流一定是最大流么？

无法找到更大的流，也无法找到最大流！



初始：网络上的流为0

$$f(s,v_1)=0$$
$$f(v_1,v_2)=0$$
$$f(v_2,t)=0$$
$$f(s,v_2)=0$$
$$f(v_1,t)=0$$

找到路径$s\rightarrow v_1\rightarrow v_2\rightarrow t$和正数10，得到一个更大的流：

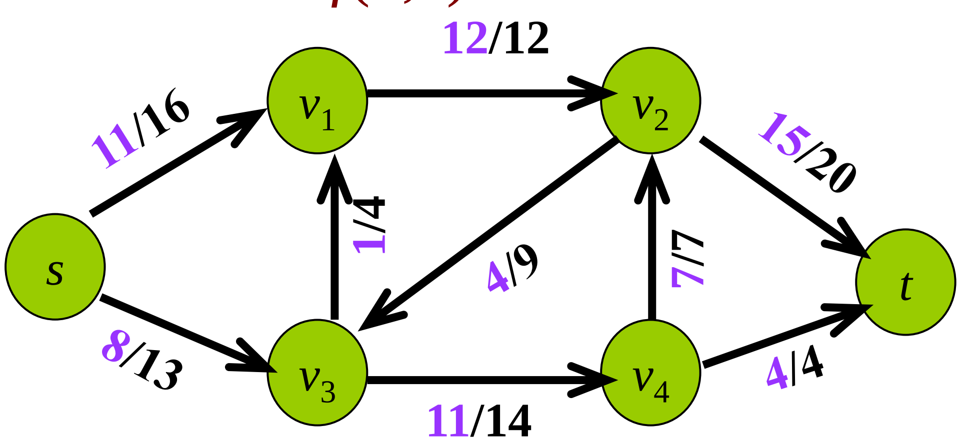
$$f(s,v_1)=10$$
$$f(v_1,v_2)=10$$
$$f(v_2,t)=10$$
$$f(s,v_2)=0$$
$$f(v_1,t)=10$$

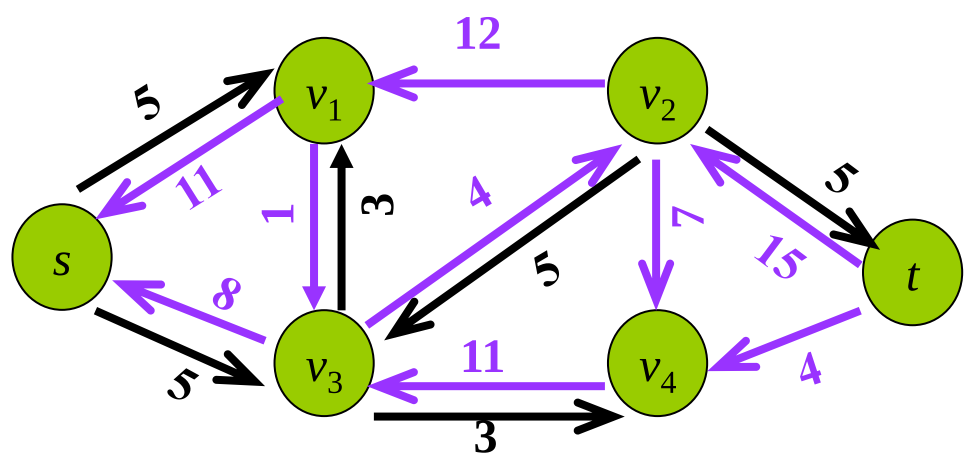
8.2 Ford-Fulkerson方法

- 如何找路径p，以便得到更大的流？
- 如何判断循环结束的条件？

相关概念

- 在一个关联的剩余网络中寻找一条增广路径
- 剩余网络(Residual network)
  - 给定流网络$G(V,E)$和一个流f，则由f诱导的G的剩余网络为$G_f=(V, E_f)$，其中E_f为
    - 对于G中每条边(u, v)，若$c(u,v)-f(u,v)>0$，则$(u, v) \in E_f$，且$c_f(u,v)=c(u,v)-f(u,v)$　(称$c_f(u,v)$为剩余容量residual capacity)
    - 对于G中每条边(u, v)，若$f(u,v)>0$，在G_f中构造边(v, u)，且$c_f(v, u)=f(u,v)$





E_f中的边要么是E中原有的边，要么是其反向边，因此$|E_f|\le 2|E|$

相关概念

- 剩余网络类似于一个容量为c_f的流网络，但包含反向边
- 也可以把流网络G看成是一个当前流f=0的剩余网络
- 可以在剩余网络中定义一个流
- 设f'是剩余网络G_f中的流，f'同样满足容量限制和流量守恒，f'也有大小

相关概念

引理. 给定流网络$G(V,E)$和其中的流f，流f'是剩余网络G_f中的流，则通过f'提升f可以得到如下的流:

$$(f\uparrow f')(u,v)=\begin{cases} f(u,v)+f'(u,v)-f'(v,u) & \text{若 } (u,v)\in E \\ 0 & \text{否则} \end{cases},$$

而且$|f\uparrow f'|=|f|+|f'|$.

需要证明$f\uparrow f'$是一个流，而且$f\uparrow f'$的大小为$|f|+|f'|$.

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & \text{若 } (u,v) \in E \\ 0 & \text{否则} \end{cases}$$

证明. 往证容量限制, 即$0 \le (f \uparrow f')(u,v) \le c(u,v)$.

对于任何$(u,v) \in E$, $c_f(u,v) = c(u,v) - f(u,v)$, $c_f(v,u) = f(u,v)$.

因此有$f'(u,v) \le c_f(u,v) = c(u,v) - f(u,v)$, $f'(v,u) \le c_f(v,u) = f(u,v)$.

故而,
$$\begin{aligned}(f \uparrow f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) \\ &\ge f(u,v) + f'(u,v) - f(u,v) = f'(u,v) \ge 0\end{aligned}$$

同时,
$$\begin{aligned}(f \uparrow f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) \\ &\le f(u,v) + f'(u,v) \le f(u,v) + c_f(u,v) \\ &\le f(u,v) + c(u,v) - f(u,v) = c(u,v)\end{aligned}$$

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v)+f'(u,v)-f'(v,u) & 若\ (u,v)\in E \\ 0 & 否则 \end{cases}$$

证明. 往证流量守恒, 即对于任何$u \in V-\{s,t\}$, 有$\sum_{v\in V}(f\uparrow f')(u,v) = \sum_{v\in V}(f\uparrow f')(v,u)$.

对于给定的u, 令$V_1=\{v|v\in V,(u,v)\in E\}$, $V_2=\{v|v\in V,(v,u)\in E\}$, 由于流网络G中没有双向边, $V_1\cap V_2=\emptyset$. 对于$v\notin V_1$, 有$f(u,v)=0$, $(f\uparrow f')(u,v)=0$; 对于$v\notin V_2$, 有$f(v,u)=0$, $(f\uparrow f')(v,u)=0$. 而且对于$v\notin V_1\cup V_2$，(u,v)和(v,u)一定不在G_f中，故而$f'(u,v)=0$, $f'(v,u)=0$.

对于$u\in V-\{s,t\}$, 由于流f满足流量守恒, 于是有$\sum_{v\in V} f(u,v)=\sum_{v\in V} f(v,u)$. 进一步有
$\sum_{v\in V_1} f(u,v) = \sum_{v\in V} f(u,v) = \sum_{v\in V} f(v,u) = \sum_{v\in V_2} f(v,u)$.

与此同时, 流f'也满足流量守恒, 于是有$\sum_{v\in V} f'(u,v)=\sum_{v\in V} f'(v,u)$. 因此, 进一步有
$\sum_{v\in V_1} f'(u,v)+\sum_{v\in V_2} f'(u,v) = \sum_{v\in V_1\cup V_2} f'(u,v) = \sum_{v\in V} f'(u,v) = \sum_{v\in V} f'(v,u) = \sum_{v\in V_1\cup V_2} f'(v,u) = \sum_{v\in V_1} f'(v,u)+\sum_{v\in V_2} f'(v,u)$, 于是可得$\sum_{v\in V_1} f'(u,v) - \sum_{v\in V_1} f'(v,u) = \sum_{v\in V_2} f'(v,u) - \sum_{v\in V_2} f'(u,v)$.

基于以上推导, 对于任何$u\in V-\{s,t\}$, $\sum_{v\in V}(f\uparrow f')(u,v) = \sum_{v\in V_1}(f\uparrow f')(u,v) = \sum_{v\in V_1}(f(u,v)+f'(u,v)-f'(v,u)) = \sum_{v\in V_1} f(u,v)+\sum_{v\in V_1} f'(u,v) - \sum_{v\in V_1} f'(v,u) = \sum_{v\in V_2} f(v,u) + \sum_{v\in V_2} f'(v,u)\ 0 - \sum_{v\in V_2} f'(u,v) = \sum_{v\in V_2}(f\uparrow f')(v,u) = \sum_{v\in V}(f\uparrow f')(v,u)$.

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & \text{若 } (u,v) \in E \\ 0 & \text{否则} \end{cases}$$

证明. 令$V_1 = \{v|(s,v) \in E\}$,令$V_2 = \{v|(v,s) \in E\}$, 可知$V_1 \cap V_2 = \emptyset$,
而且$V_1 \cup V_2 \subseteq V$.

$$|f \uparrow f'| = \sum_{v \in V} (f \uparrow f')(s,v) - \sum_{v \in V} (f \uparrow f')(v,s)$$

$$= \sum_{v \in V_1} (f \uparrow f')(s,v) - \sum_{v \in V_2} (f \uparrow f')(v,s)$$

$$= \sum_{v \in V_1} (f(s,v) + f'(s,v) - f'(v,s)) - \sum_{v \in V_2} (f(v,s) + f'(v,s) - f'(s,v))$$

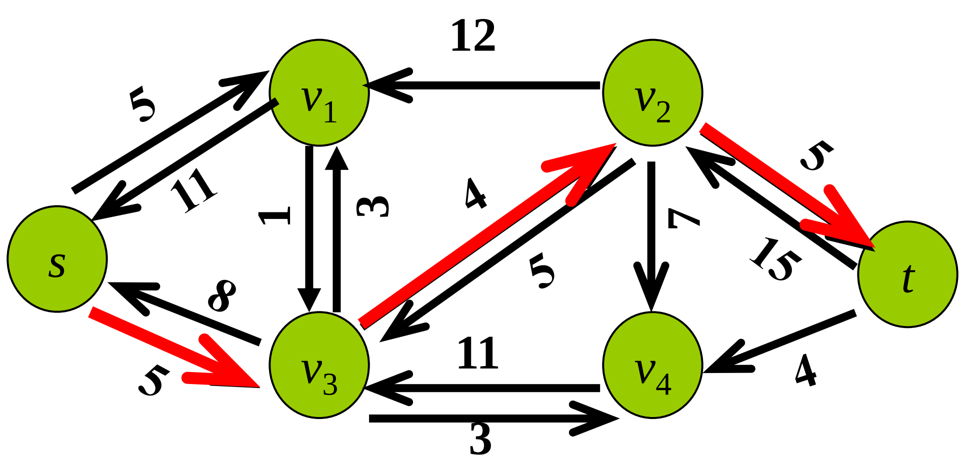
$$= \sum_{v \in V_1} f(s,v) + \sum_{v \in V_1} f'(s,v) - \sum_{v \in V_1} f'(v,s) - \sum_{v \in V_2} f(v,s) - \sum_{v \in V_2} f'(v,s) + \sum_{v \in V_2} f'(s,v)$$

$$= \sum_{v \in V_1} f(s,v) - \sum_{v \in V_2} f(v,s) + \sum_{v \in V_1} f'(s,v) + \sum_{v \in V_2} f'(s,v) - \sum_{v \in V_1} f'(v,s) - \sum_{v \in V_2} f'(v,s)$$

$$= \underbrace{\boxed{\sum_{v \in V_1} f(s,v) - \sum_{v \in V_2} f(v,s)}}_{|f|} + \underbrace{\boxed{\sum_{v \in V_1 \cup V_2} f'(s,v) - \sum_{v \in V_1 \cup V_2} f'(v,s)\,.}}_{|f'|}$$

相关概念

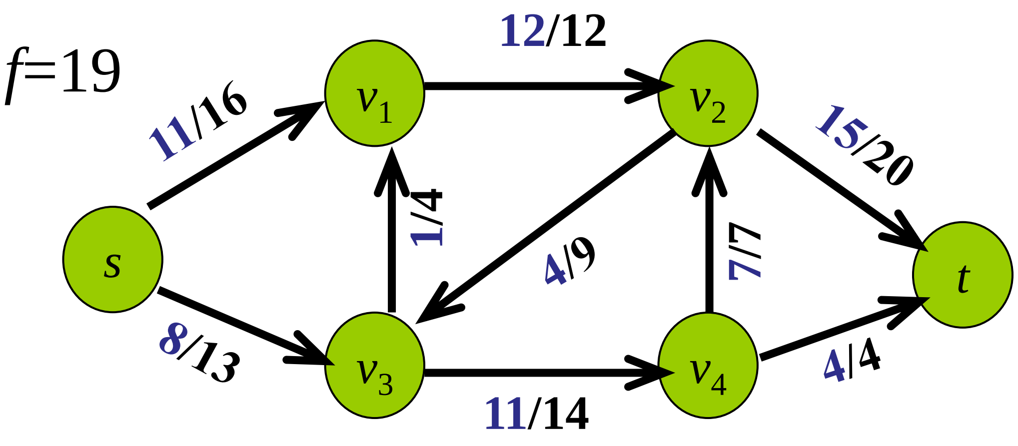
- 增广路径
  - 剩余网络中的由源结点s到汇点t的一条路径p
- 增广路径p的剩余容量
  - $c_f(p)=\min\{c_f(u, v) : (u, v)$属于路径$p\}$
    - 表示了该路径能够增加的流的最大值



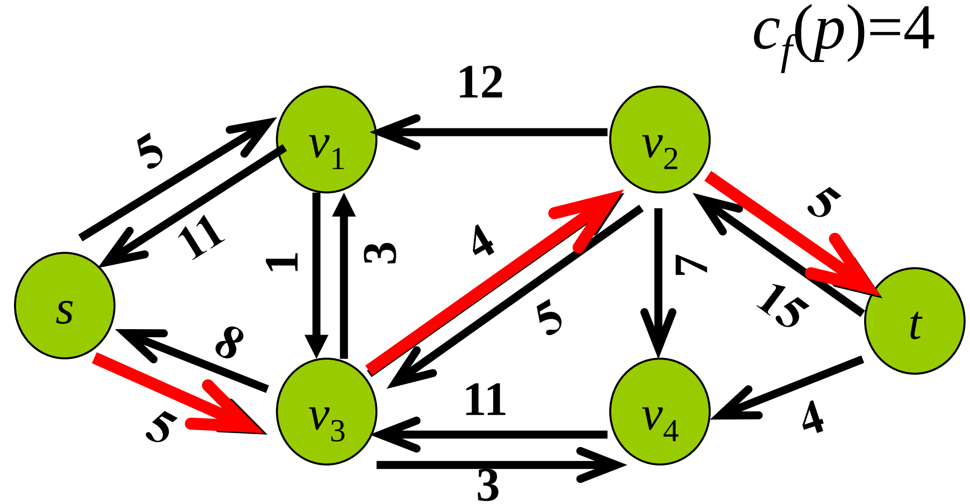
图中红色标注的路径为一条增广路径，其剩余容量为4

Ford-Fulkerson方法

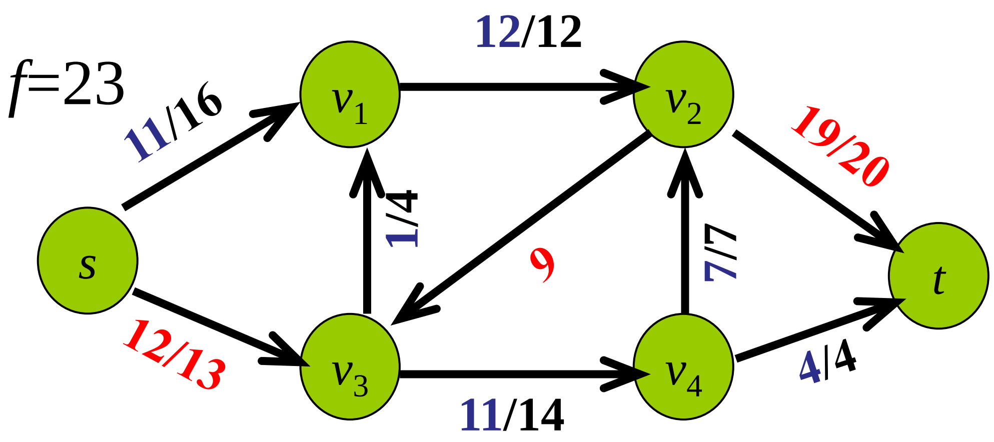
- 在剩余网络中寻找增广路径



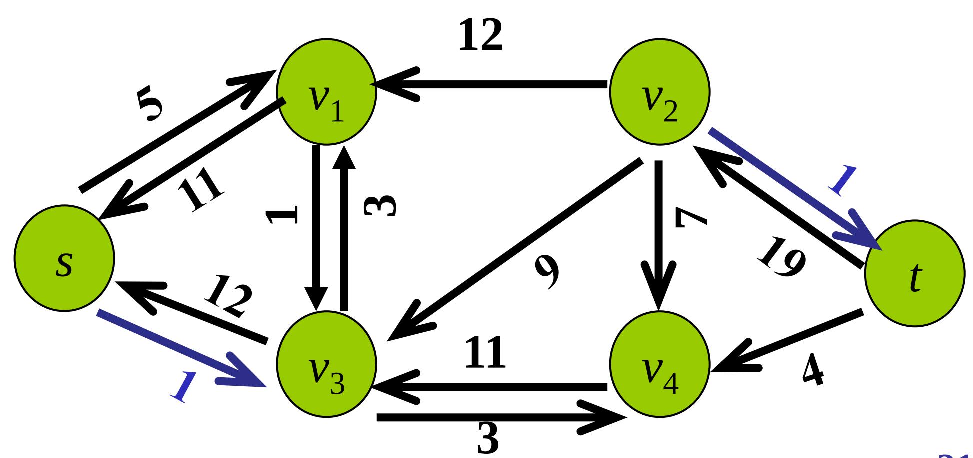
(a) 流网络G及流f



(b) 由(a)诱导的剩余网络



(c) 由增广路径得到的更大流



(d) 由(c)诱导的剩余网络

Ford-Fulkerson方法

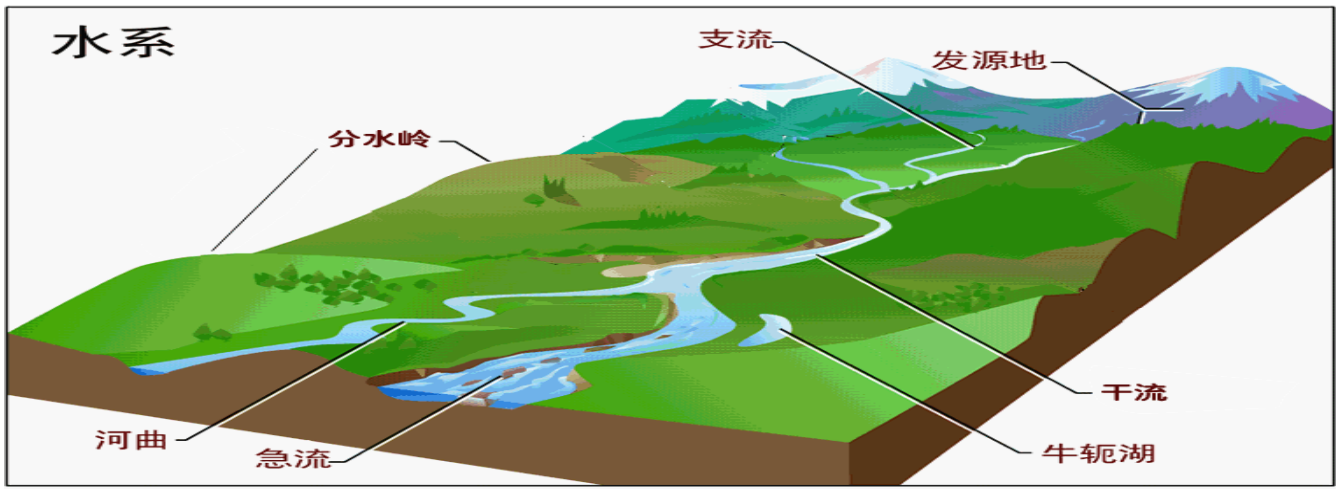
- FF算法的核心是：通过增广路径不断增加路径上的流量，直到找到一个最大流为止

问题：
如何判断算法结束时，确实找到了一个最大流？

如何判断是否已获得最大流?

河水的最大流量取决于

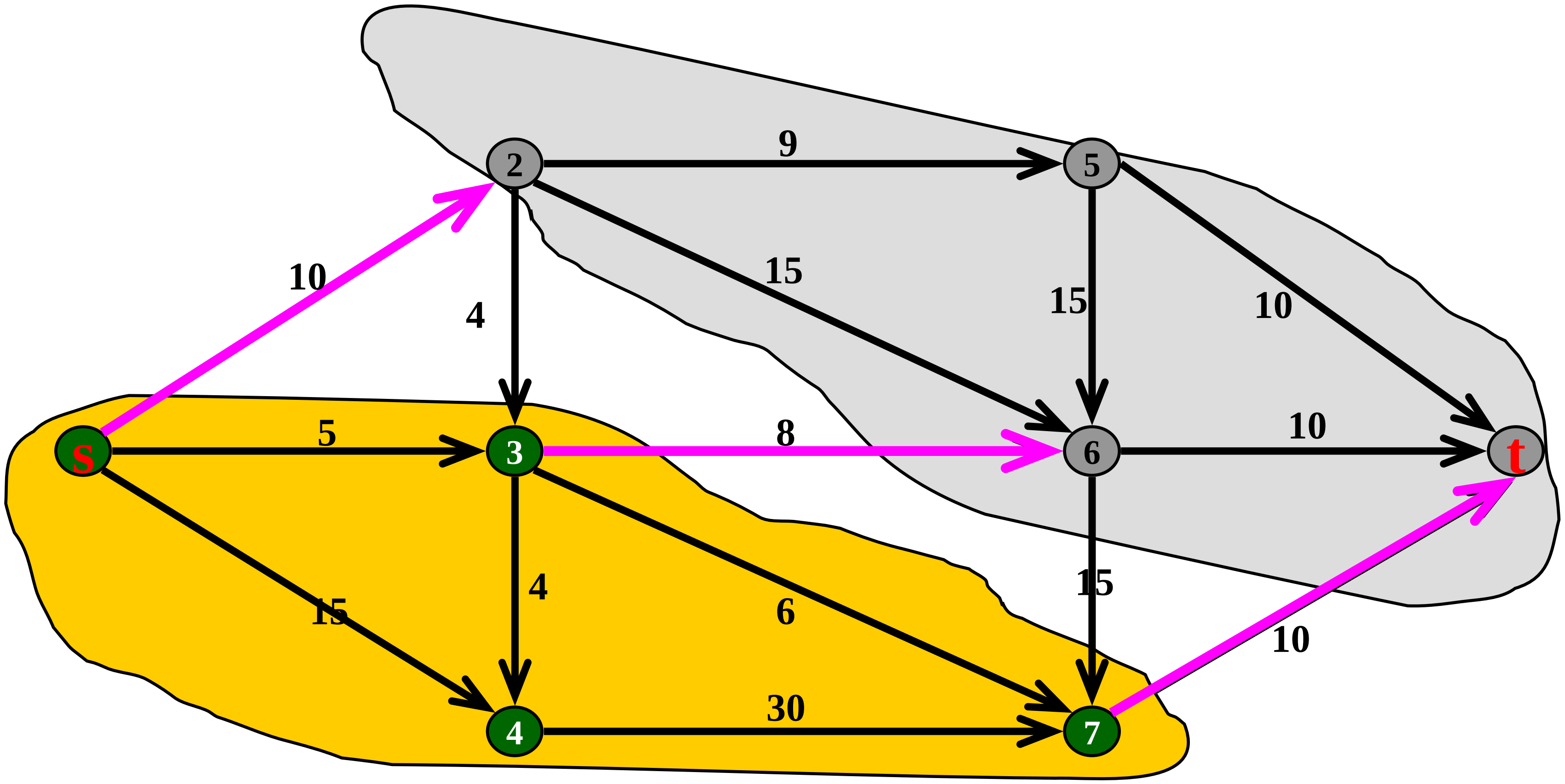
干流中河道狭窄处的通行能力



这种观察能否用于最大流问题呢?

如何判断是否已获得最大流?

从s流到t的最大流量不会超过10+8+10=28



流网络的割

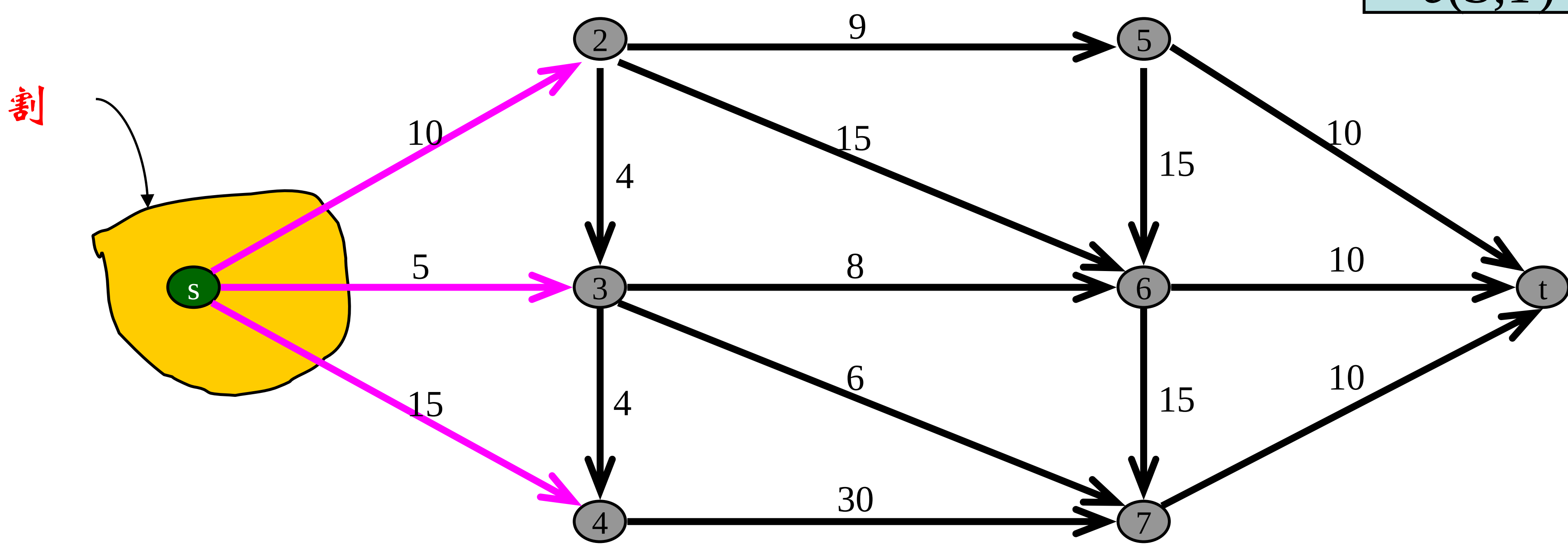
给定流网络 $G=(V,E)$,其源为s， 汇为t，

G的一个**割(cut)**是V的2-集合划分(S, T), $T=V-S$, 且$s\in S$, $t\in T$

割的**容量**定义为

$$c(S,T)=\sum_{u\in S,v\in T} c(u,v)$$

$c(S,T) = 30$

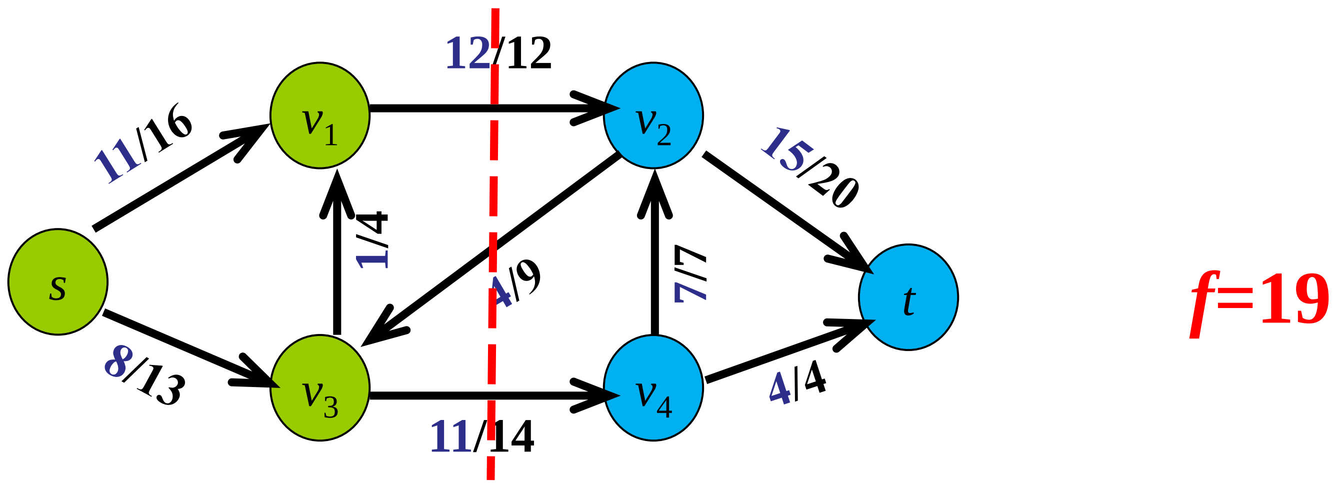


流网络的割

引理1. 设f为流网络G的一个流，该流网络的源结点为s，汇点为t，设(S,T)为流网络G的任意一个割，则横跨割(S,T)的净流量为$|f|$.

横跨割(S,T)的净流量定义为：

$$f(S,T)=\sum_{u\in S}\sum_{v\in T}f(u,v)-\sum_{u\in S}\sum_{v\in T}f(v,u)$$



流网络G及流f

$$f(S,T)=\sum_{u\in S}\sum_{v\in T} f(u,v)-\sum_{u\in S}\sum_{v\in T} f(v,u)$$

证明. 对于任一$u \in V - \{s,t\}$, 有$\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u) = 0$.

因此, $|f| = \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{u\in S-\{s\}}(\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u))$.

$$\begin{aligned}
|f| &= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{u\in S-\{s\}}\sum_{v\in V} f(u,v) - \sum_{u\in S-\{s\}}\sum_{v\in V} f(v,u) \\
&= \sum_{v\in V}\left(f(s,v) + \sum_{u\in S-\{s\}} f(u,v)\right) - \sum_{v\in V}\left(f(v,s) + \sum_{u\in S-\{s\}} f(v,u)\right) \\
&= \sum_{v\in V}\sum_{u\in S} f(u,v) - \sum_{v\in V}\sum_{u\in S} f(v,u) .
\end{aligned}$$

由于$V = S \cup T$, 并且$S \cap T = \emptyset$,

$$\begin{aligned}
|f| &= \sum_{v\in S}\sum_{u\in S} f(u,v) + \sum_{v\in T}\sum_{u\in S} f(u,v) - \sum_{v\in S}\sum_{u\in S} f(v,u) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\
&= \sum_{v\in T}\sum_{u\in S} f(u,v) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\
&\quad \cancel{+\left(\sum_{v\in S}\sum_{u\in S} f(u,v) - \sum_{v\in S}\sum_{u\in S} f(v,u)\right)} .
\end{aligned}$$

推论1. 流网络G中任意流的值不能超过G的任意割的容量.

由引理知：

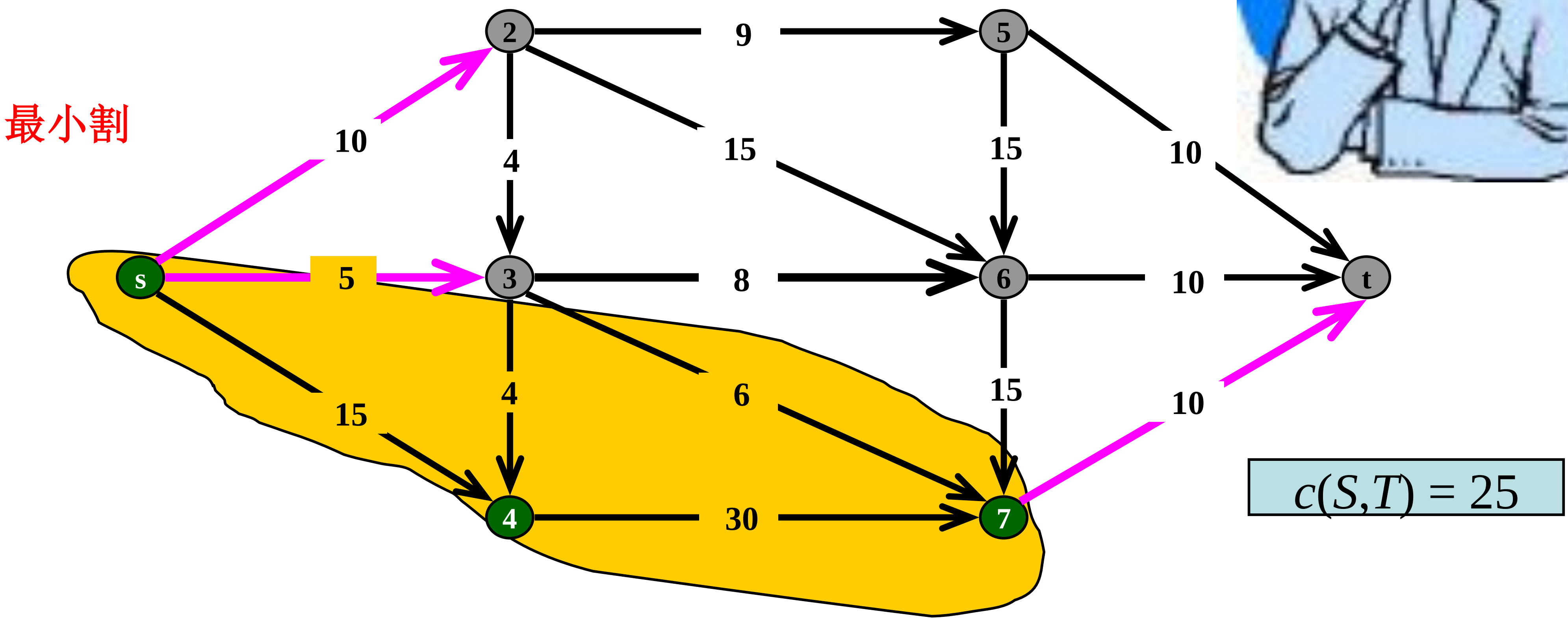
$$|f| = f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u)$$

$$\le \sum_{u\in S}\sum_{v\in T} f(u,v)$$

$$\le \sum_{u\in S}\sum_{v\in T} c(u,v) \quad = c(S,T)$$

最小割

- 一个流网络的最小割
  是指：整个网络中容量最小的割



Max-Min关系

最大流最小割定理：

设f为流网络$G(V,E)$一个流，该流网络的源结点为s，汇点为t，则下面命题等价：

1. f是G的最大流.
2. 剩余网络G_f不包含增广路径.
3. 对于G的某个划分(S, T), $|f|=c(S, T)$.

一个最大流的值实际上等于一个最小割的容量

Max-Min关系：对偶关系

最大流与最小割

最大匹配与最小覆盖

……

Max-Min关系

- 对同一问题从不同角度考虑，有两种对立的描述
  - 例如，平面中矩形面积与周长的关系

正方形： 周长一定，面积最大的矩形

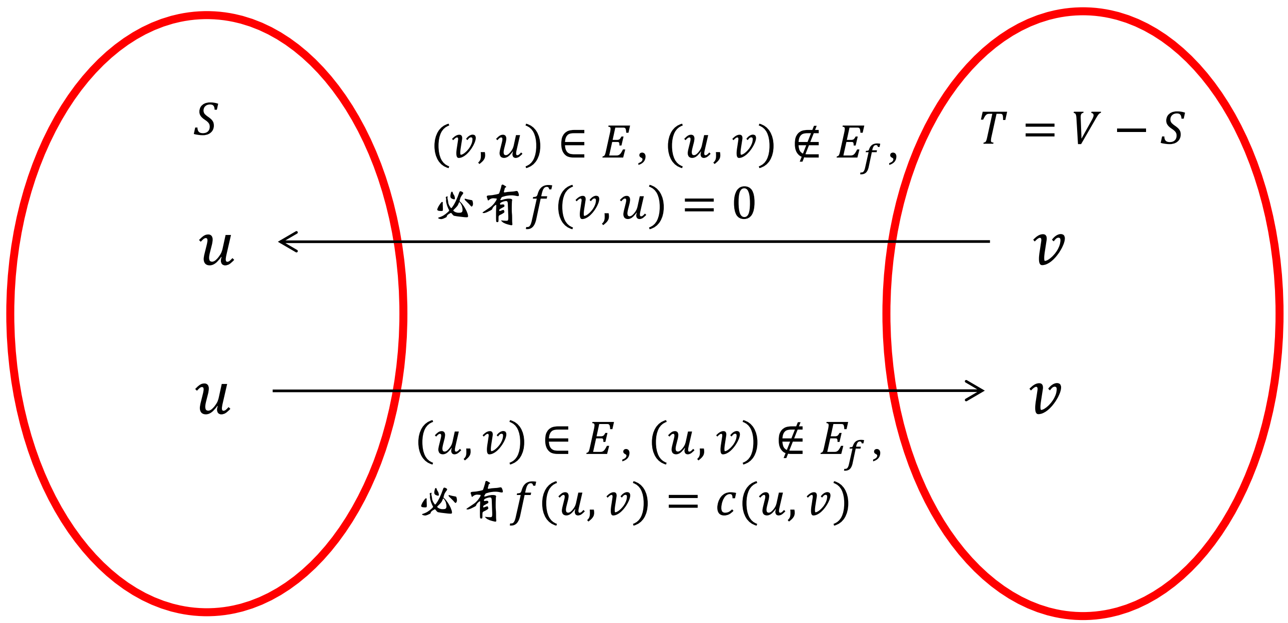
面积一定，周长最小的矩形

Max问题

Min问题

1. f是G的最大流.
2. 剩余网络G_f不包含增广路径.
3. 对于G的某个划分(S, T), $|f|=c(S, T)$.

证明概要：1→ 2，3→ 1比较直观，主要说明2→ 3



$S = \{v | v \in V, G_f$中从s到v有路径可达$\}$

$\forall u \in S, v \in T, (u,v) \notin E_f, E_f$ 是G_f的边集

$$f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u) = \sum_{u\in S}\sum_{v\in T} c(u,v) = C(S,T)$$

利用Max-Min关系求解最大流问题

1. 初始化一个可行流f
   - 0-流：所有边的流量均等于0的流
2. 不断将f增大，直到f不能继续增大为止
3. 找出一个割(S,T)使得$|f|=c(S,T)$
   - 由此断言f是最大流，而(S,T)是最小割

Max-Min关系提供了高效求解最大流-最小割问题的机制！

Ford-Fulkerson算法

算法Ford-Fulkerson(G,s,t)

Input 流网络G，源s，汇t

Output G中从s到t的最大流

1. For $\forall(u,v)\in E[G]$ do
2. $f(u,v)\leftarrow 0$
3. While G_f存在增广路径p do
4. $c_f(p)=\min\{c_f(u,v)\mid (u,v)$是$p$上的边$\}$
5. For p上的每条边(u,v) do
6. If (u,v)是流网络中的边 Then
7. $f(u,v)\leftarrow f(u,v)+c_f(p)$
8. Else
9. $f(v,u)\leftarrow f(v,u)-c_f(p)$

Ford-Fulkerson算法分析

- 正确性分析
  1. 算法输出的一定是最大流
     - 由最大流－最小割定理可得
  2. 算法可终止性
     - 假设整数容量，每次流量增加1

Ford-Fulkerson算法分析

算法Ford-Fulkerson(G,s,t)

Input 流网络G，源s，汇t

Output G中从s到t的最大流

1. For $\forall(u,v)\in E[G]$ do
2. $\quad f(u,v)\leftarrow 0$
3. While G_f存在增广路径p do
4. $\quad c_f(p)=\min\{c_f(u,v) \mid (u,v)$是$p$上的边$\}$
5. $\quad$For p上的每条边(u,v) do
6. $\quad\quad$If (u,v)是流网络中的边 Then
7. $\quad\quad\quad f(u,v)\leftarrow f(u,v)+c_f(p)$
8. $\quad\quad$Else
9. $\quad\quad\quad f(v,u)\leftarrow f(v,u)-c_f(p)$

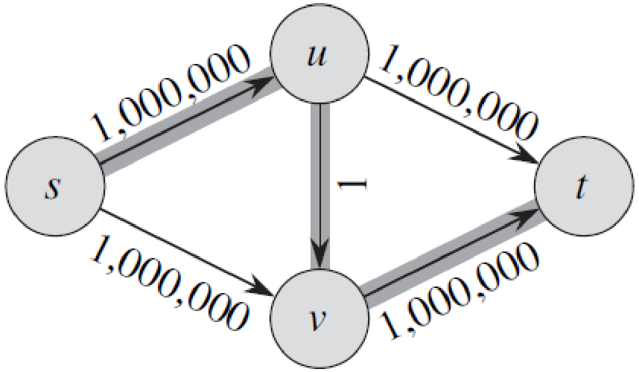
1-2步：$O(E)$

3-9步：循环次数最多为$|f^*|$

第3步在G_f中找路径（深度或宽度优先）代价$O(E)$

总的复杂度$O(|f^*||E|)$

最坏情况下，找到2000000条增广路径才能得到最大流，
最好情况下只需要2条！



如何改进Ford-Fulkerson算法？

HITWH
SE

加速增广路径的寻找

- 最短增广路径算法：始终沿着G_f中具有最少边的路径进行增广
- 最大增广路径算法：始终沿着G_f中具有最大容量的路径进行增广

Edmonds-Karp算法

- 利用宽度优先在剩余网络G_f中寻找增广路径
  - 从源结点s到汇点t的一条最短路径
  - 每条边的权重为单位距离
  - 算法复杂性：$O(|V||E|^2)$

$\delta_f(u, v)$=剩余网络G_f中从结点u到结点v的最短路径距离

Edmonds-Karp算法

引理2：如果Edmonds-Karp算法运行在流网络$G(V,E)$上，该网络的源结点为s，汇点为t，则对于所有的结点$v\in V$-{s,t}，剩余网络G_f中的最短路径距离$\delta_f(s, v)$随着每次流量的递增而单调递增。

证明：反证法，假设存在一个$v \in V - \{s,t\}$，使得$\delta_f(s,v)$减小.

令f是第一次发生最小距离减小之前的流，f'是第一次发生最小距离减小之后的流.

令v为f增长为f'的过程中，在最小距离减小的顶点中具有最小$\delta_{f'}(s,v)$，$\delta_{f'}(s,v) < \delta_f(s,v)$.

令路径$p: s \leadsto u \rightarrow v$是$G_{f'}$中从$s$到$v$的最短路径，最后一条边为$(u,v)$.

于是$\delta_{f'}(s,v) = \delta_{f'}(s,u) + 1$,而且$\delta_f(s,u) \leq \delta_{f'}(s,u)$. ($v$的选择方式)

考查边(u,v)是否在G_f中，如果在：

$\delta_f(s,v) \leq \delta_f(s,u) + 1 \leq \delta_{f'}(s,u) + 1 = \delta_{f'}(s,v)$. 矛盾！

Edmonds-Karp算法

证明：反证法，假设存在一个$v \in V - \{s,t\}$，使得$\delta_f(s,v)$减小.令f是第一次发生最小距离减小之前的流，f'是第一次发生最小距离减小之后的流.令v为f增长为f'的过程中，在最小距离减小的顶点中具有最小$\delta_{f'}(s,v)$，$\delta_{f'}(s,v) < \delta_f(s,v)$.

令路径$p: s \rightsquigarrow u \to v$是$G_{f'}$中从$s$到$v$的最短路径，最后一条边为$(u,v)$.于是有$\delta_{f'}(s,v) = \delta_{f'}(s,u) + 1$，而且$\delta_f(s,u) \le \delta_{f'}(s,u)$. 考查边$(u,v)$是否在$G_f$中，如果在：$\delta_f(s,v) \le \delta_f(s,u) + 1 \le \delta_{f'}(s,u) + 1 = \delta_{f'}(s,v)$. 矛盾！

若不在，则边(u,v)的出现必然因为f'在f中增大了(v,u)上的流.

Edmonds-Karp算法总是在最短路径上增大流，因此(v,u)必为G_f中从s到u的某条最短路径的最后一条边.

于是有$\delta_f(s,v) = \delta_f(s,u) - 1 \le \delta_{f'}(s,u) - 1 = \delta_{f'}(s,v) - 2$.

再次矛盾！

因此不存在一个$v \in V - \{s,t\}$，使得$\delta_f(s,v)$减小，引理2成立。

Edmonds-Karp算法运行时间：$O(VE^2)$

定理1：如果Edmonds-Karp算法运行在流网络$G(V,E)$上，该网络的源结点为s，汇点为t，则算法所执行的流量递增操作的总次数为$O(VE)$。

证明要点：考查增广路径上剩余容量最小的边(u,v)——关键边.

$\delta_f(s,v)=\delta_f(s,u)+1$，流$f$增大后关键边$(u,v)$将消失.

(u,v)再次出现时，是因为某次增大流f'时边(v,u)出现在增广路径上.

于是有$\delta_{f'}(s,u)=\delta_{f'}(s,v)+1$. 同时，$\delta_{f'}(s,v)\geq\delta_f(s,v)$(引理2).

因此$\delta_{f'}(s,u)-\delta_f(s,u)\geq 2$.

边(u,v)成为关键边,而后再次成为关键边,s到u的最短距离至少增长2.

s到u的最短距离最大为$|V|-1$，因此每条边(u,v)成为关键边的次数不超过$(|V|-1)/2$. 共有$|E|$条边，所有边成为关键边次数之和不超过$(|V|-1)|E|/2$.

流量递增操作次数不大于所有边成为关键边次数之和(一次递增可能有多个关键边). 因此流量递增次数为$O(VE)$.

关于期末考试

- 简答题（x分）
  - 关于基本概念、基本原理
- 计算题（25-x分）
  - 递归方程、平摊分析等内容
- 算法设计题（3×20分）
  - 分治算法、动态规划技术、贪心算法
- 算法实例题（15分）
  - 树搜索、最大流

本学期的算法课程结束了，各位同学的算法学习没有结束……