第五章 匹配与网络流Ⅱ

计算机系网络所：张小平

清華大學
Tsinghua University

主要内容

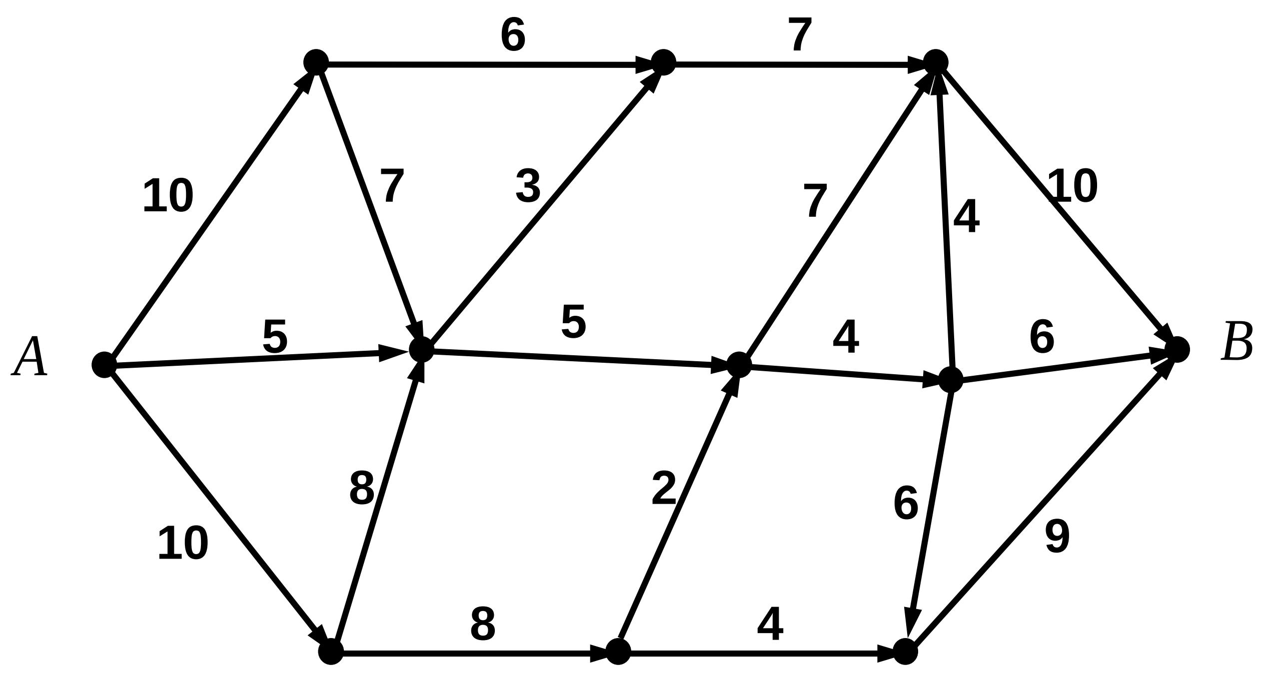
- 5.1 二分图的最大匹配
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- **5.5 网络流图**
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

网络流图

- 网络流问题：
  - 网络流理论主要用于解决一个传输网络中的运载容量与运载流量之间的关系。
  - 目的在于追求一个最大和最佳的运输方案

网络流图

例：城市间公路网建设

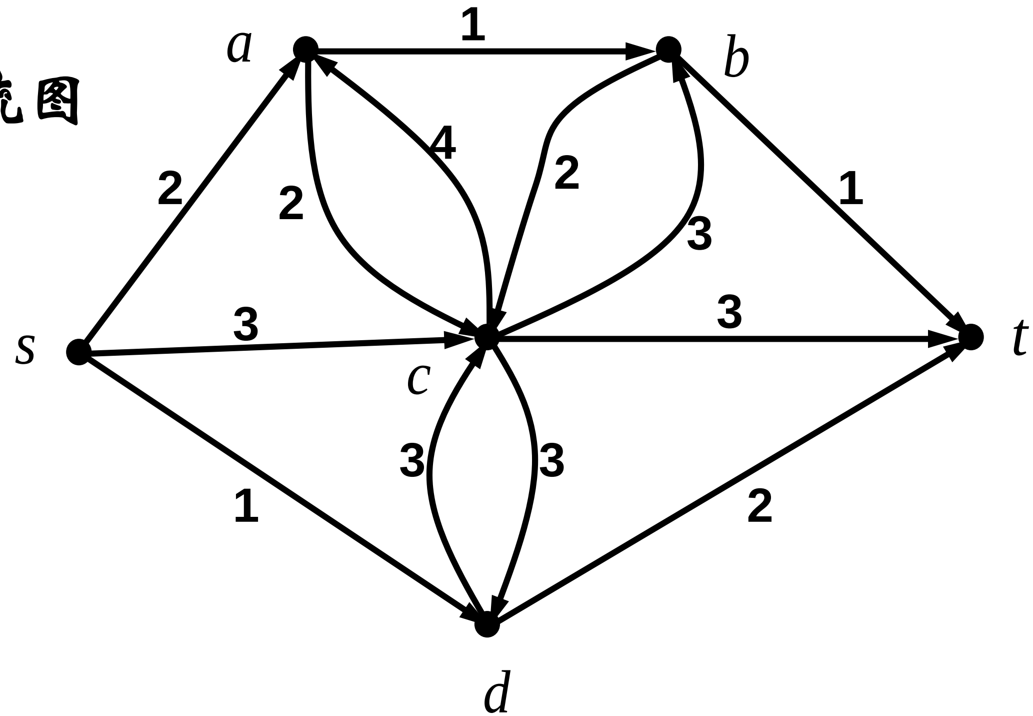


网络流图

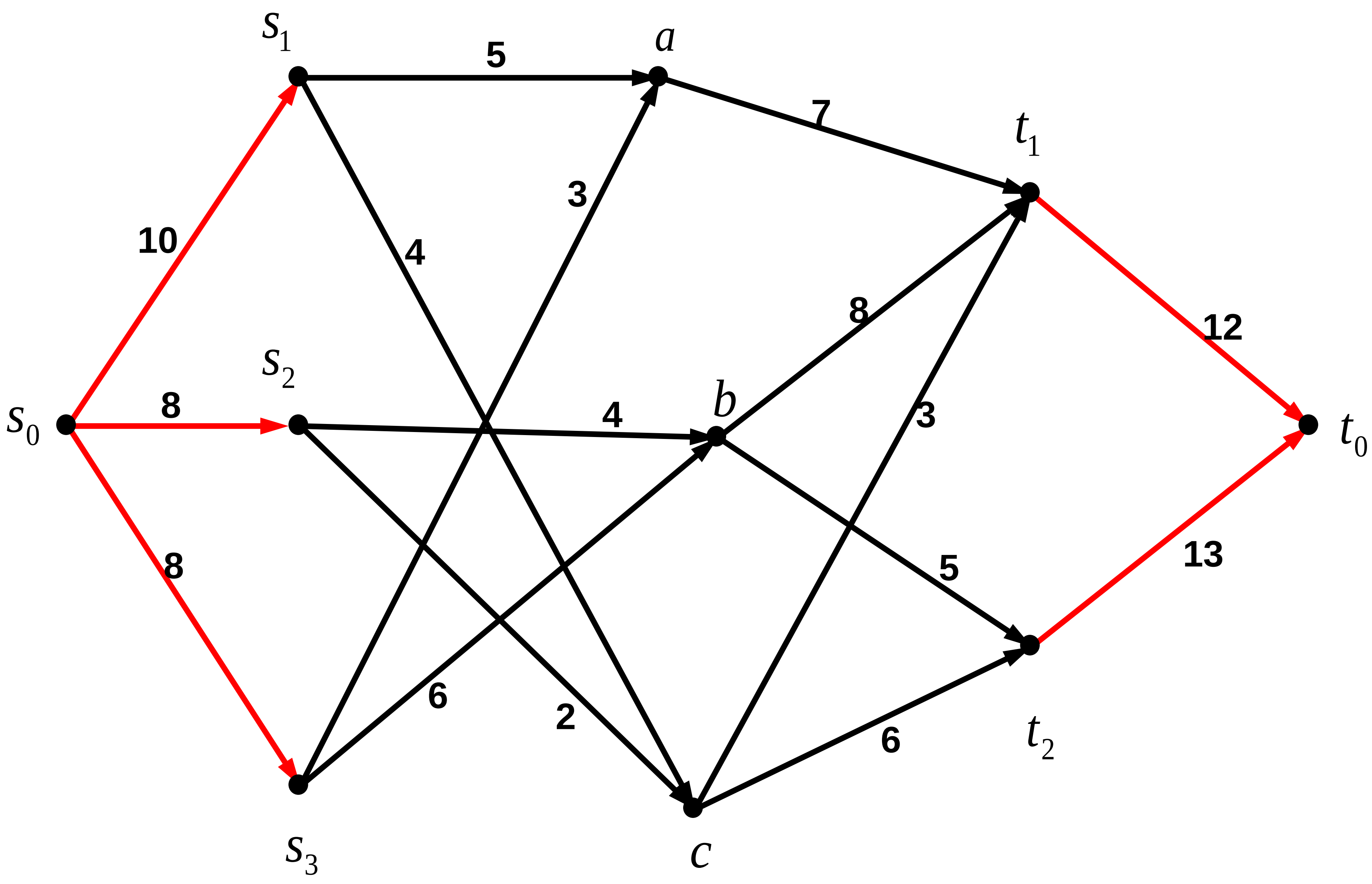
- 定义5.5.1 一个运输网络N（或称**网络流图**）是一个没有自环的有向连通图。它满足：
  - 只有一个正度为零的结点 t，称为**汇**
  - 只有一个负度为零的结点 s，称为**源**
  - 每条边 (i,j) 都有一个非负实数权 c_{ij}，称为该边的**容量**。如果结点 i 到结点 j 没有边，则 $c_{ij}=0$

例：

如右图，就是一个网络流图



此类网络可以看作是某种产品
从产地 s 通过不同的道路到达销地 t，
边的容量可以表示该边可以通过的量，
也可以表示通过该边运输所需要的代价等



s_1
a
5
7
t_1
3
10
4
8
12
s_2
b
s_0
8
4
3
t_0
8
5
13
6
2
6
t_2
s_3
c

网络流图

- 在网络流图N中，如果每条边e_{ij}都给定一个非负实数f_{ij}（称其为边流量），满足

1. $f_{ij} \le c_{ij},\quad e_{ij} \in N$ ⟷ 每条边的流量不超过边容量
2. $\sum_j f_{ij} = \sum_k f_{ki},\quad i \ne s,t$ ⟷ 注入i结点的流量总和等于结点i发出的流量总和
3. $\sum_j f_{sj} = \sum_k f_{kt} = w$ ⟷ 源结点发出的流量总和等于汇结点接收的流量总和

那么就把这一组f_{ij}叫做该网络的**允许流**，W称为它的**流量**。

允许流分布f

Tsinghua University

网络流图

- 在网络流图N的一个允许流分布f里，满足$f_{ij}=c_{ij}$的边称为饱和边，否则就是非饱和边

- 如果一个允许流分布使得网络的流量w_0为极大，即

$$w_0=\max\sum_j f_{sj}$$

就说w_0是网络的最大流

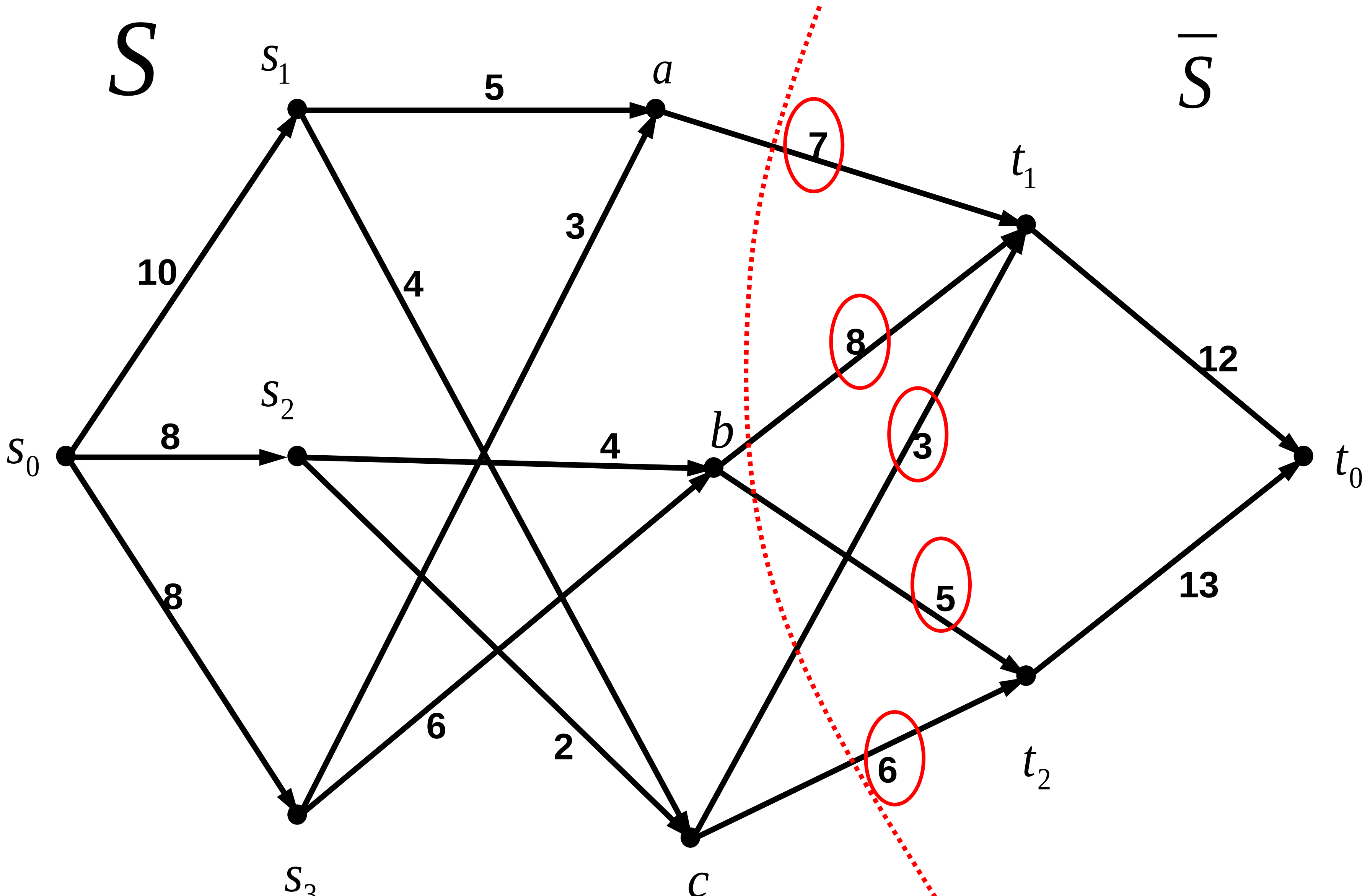
网络流图

- 定义5.5.2 设S是网络流图N=(V,E)中的一个结点集，满足

  1. $s \in S$
  2. $t \in \bar{S}$, $\bar{S} = V - S$

  则全部有向边 (i,j), $i \in S$, $j \in \bar{S}$ 的集合称为N的一个**割切**，记为 $(S,\bar{S})$，$(S,\bar{S})$ 中各边的容量之和称为该**割切的容量**，记为 $c(S,\bar{S})$，即

$$C(S,\bar{S}) = \sum_{(i,j)\in(S,\bar{S})} c_{ij}$$



S
$\overline{S}$
s_0
s_1
s_2
s_3
a
b
c
t_0
t_1
t_2
10
8
8
5
4
3
4
6
2
7
8
3
5
6
12
13

网络流图

- 定理5.5.1 网络的最大流量小于等于最小的割切容量，即

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

证明：

– 设 f 是给定网络的任一容许流分布，则满足

$$\sum_j f_{sj} = w \qquad (1)$$

$$\sum_j \left(f_{ij} - f_{ji}\right) = 0 \qquad i \ne s,t \qquad j \in V \qquad (2)$$

网络流图

- 任给一个割切 $(S,\overline{S})$，满足 $s \in S, t \in \overline{S}$。由(1)(2)式可知，流量从源S发出后，将不会凭空衰减或消失，保持守恒，即对于一个包含源 s 的任一集合，其整体对外注出的流量将等同于W，即

$$\sum_{\substack{i \in S \\ j \in \overline{S}}} \left(f_{ij} - f_{ji} \right) = w$$

网络流图

$$\sum_{\substack{i\in S\\ j\in \overline{S}}}\left(f_{ij}-f_{ji}\right)=w$$

故 $$w=\sum_{\substack{i\in S\\ j\in \overline{S}}}\left(f_{ij}-f_{ji}\right)\le\sum_{\substack{i\in S\\ j\in \overline{S}}}\left(f_{ij}\right)\le\sum_{\substack{i\in S\\ j\in \overline{S}}}c_{ij}=C\left(S,\overline{S}\right)$$

由于允许流分布与割切的任意性，定理得证！

网络流图

- 定理5.5.1 网络的最大流量小于等于最小的割切容量，即

$$\max(w) \le \min C(S, \overline{S})$$

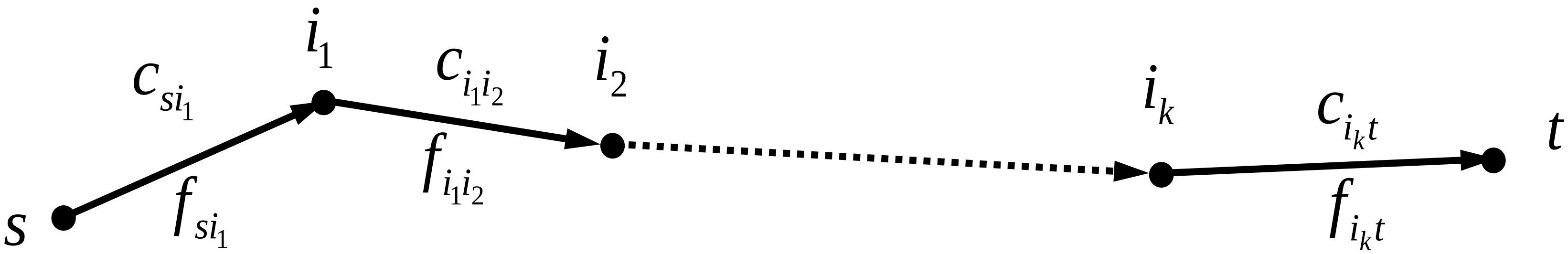
网络流图

思考：

- 如果网络的容许流并不是最大流，就一定存在从 s 到 t 的增流路径。

- 如何找到增流路径？

- 首先我们需要认识什么样的路径是增流路径！

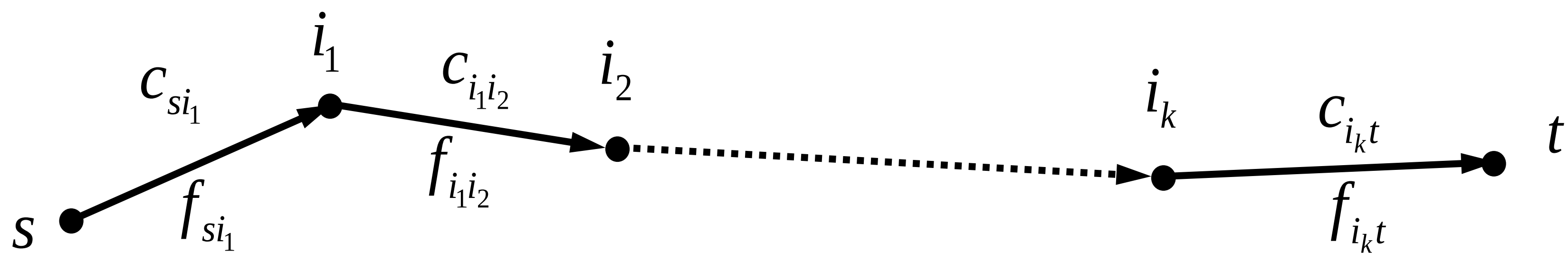


路径 P_{st} ，每条边的方向都是从 i_j 到 i_{j+1} ，称为向前边

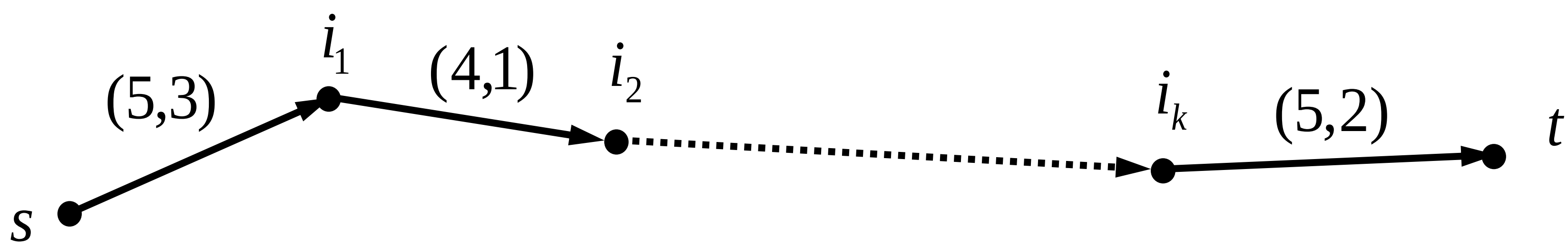
假如每条边上都有 $f_{ij} < c_{ij}$ ，则令 $\delta = \min\limits_{e_{ij} \in P_{st}} \left(c_{ij} - f_{ij} \right)$ ，

此时令每条边的流量都增加 δ，

结果仍然是网络的容许流分布，但是流量增加了 δ



为标识清晰，后面用二元组(a,b)表示每条边当前的流量状态，a表示该边的容量，b表示当前的流量，即：$a = c_{ij}$ ，$b = f_{ij}$

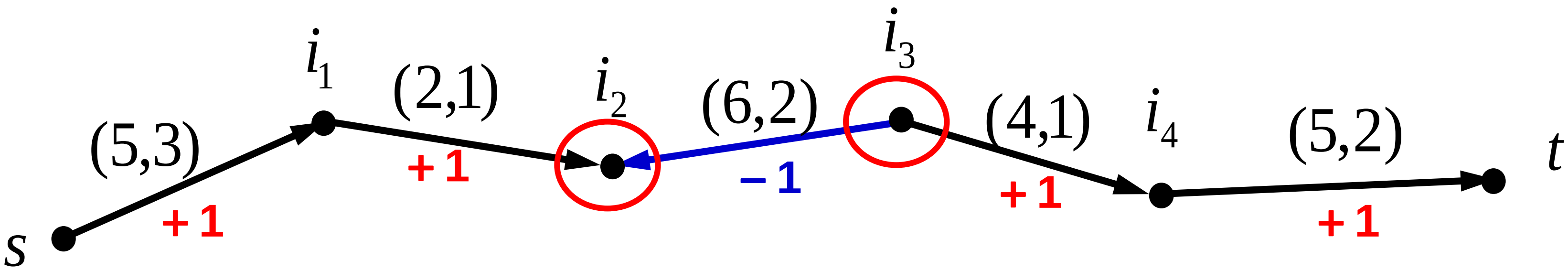


网络流图

思考：

- 从 s 到 t 的所有由向前边组成的路径有可能成为增流路径

- 如果从 s 到 t 的路径夹杂着向后边，是否有可能成为增流路径？



假如每条向前边上都有 $f_{ij} < c_{ij}$ ，则令 $\delta_1 = \min\limits_{e_{ij} \in P_{st}} \left(c_{ij} - f_{ij}\right)$ ，

假如每条向后边上都有 $f_{ij} > 0$ ，则令 $\delta_2 = \min f_{ij}$ ，

令 $\delta = \min\left(\delta_1, \delta_2\right)$ ，则路径中的流量可增加δ，

结果仍然是网络的容许流分布，但是流量增加了δ

思考：网络流图中还有其他什么类型的增流路径？

网络流图中只有这两种类型的增流路径！

网络流图

- 定理5.5.1 网络的最大流量小于等于最小的割切容量，即

$$\max(w) \le \min C(S, \overline{S})$$

网络流图

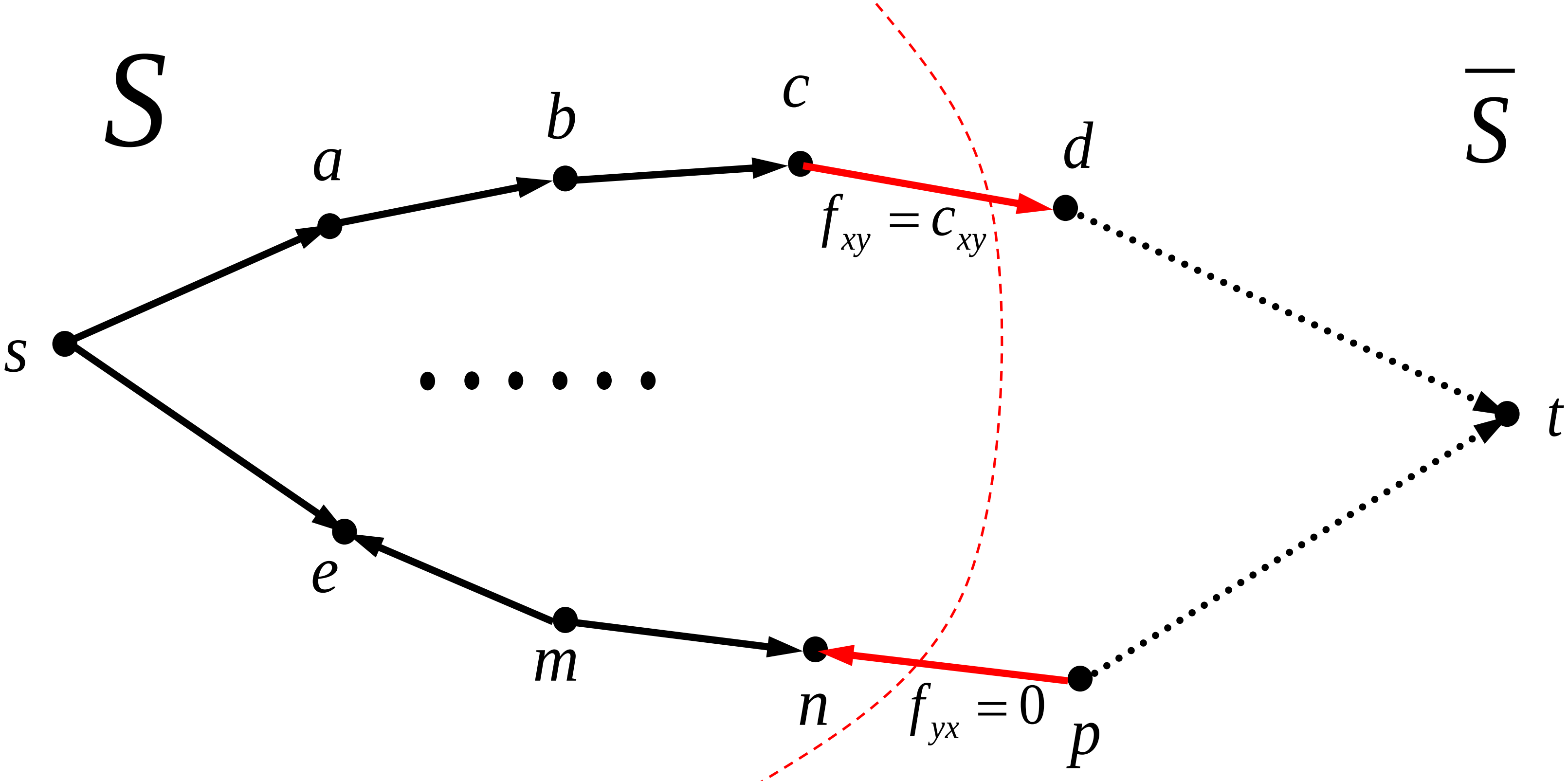
- 定理5.5.2 网络流图N中，其最大流量等于其最小割切的容量，即

$$\max(w) = \min C(S, \overline{S})$$

证明：

- 设网络中已存在一个最大的容许流 f
- 定义一个割切 $(S, \overline{S})$ 如下：
  1. $s \in S$
  2. 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) 是向后边且 $f_{yx} > 0$，则 $y \in S$



$$w = \sum_{\substack{x \in S \\ y \in \overline{S}}} \left(f_{xy} - f_{yx} \right) = \sum_{\substack{x \in S \\ y \in \overline{S}}} c_{xy} = C\left(S, \overline{S}\right)$$

$$\max(w) \le \min C(S, \overline{S}) \qquad \max(w) = \min C(S, \overline{S})$$

- 在此方法下，一定会有 $t \notin S$ ，否则从s到t将存在一条增流路径，与f是最大容许流分布矛盾
- 故 $\overline{S} \neq \Phi$。
- 据该割切的定义方法，对于任意跨于S和 $\overline{S}$ 的

边(x,y)： 若 (x,y)是向前边，必定有 $f_{xy} = c_{xy}$

若 (x,y)是向后边，必定有 $f_{yx} = 0$

- 此时流量为最大流量w，且满足

$$w = \sum_{\substack{x \in S \\ y \in \overline{S}}} \left(f_{xy} - f_{yx} \right) = \sum_{\substack{x \in S \\ y \in \overline{S}}} c_{xy} = C\left(S, \overline{S}\right)$$

- 又根据定理5.5.1 $\max(w) \le \min C(S, \overline{S})$
- 故 $\max(w) = \min C(S, \overline{S})$ 证毕！

网络流图

- 定理5.5.2 网络流图N中，其最大流量等于其最小割切的容量，即

$$\max(w) = \min C(S, \overline{S})$$

主要内容

- 5.1 二分图的最大匹配
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- **5.6 Ford-Fulkerson最大流标号算法**
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

Ford-Fulkerson最大流标号算法

- Ford和Fulkerson最先给出了计算运输网络最大流量的标号算法，该算法就是以定理5.5.2证明中给出的方法为基础，包含了两个过程：
  - 标号过程
  - 增流过程
  - 算法不断循环进行这两个过程，直到不存在增流路径为止。

Ford-Fulkerson最大流标号算法

- 标号过程：
  - 对图中每个结点都进行二元组标号 $(v^{\pm},\delta)$，其中，v表示标号从哪个结点来，δ 表示可增流的量，“±”上标表示正向流还是反向流
  - 源结点 s 给定标号为 $(-,\infty)$
  - 若$e=(u,v)$，且 $f(e)<c(e)$，则标号方向为正，v 得到标号 (u^{+},δ_v)，其中 $\delta_v=\min\left(\delta_u,c(e)-f(e)\right)$
  - 若 $e=(v,u)$，且 $f(e)>0$，则标号方向为负，v 得到标号 (u^{-},δ_v)，其中 $\delta_v=\min\left(\delta_u,f(e)\right)$

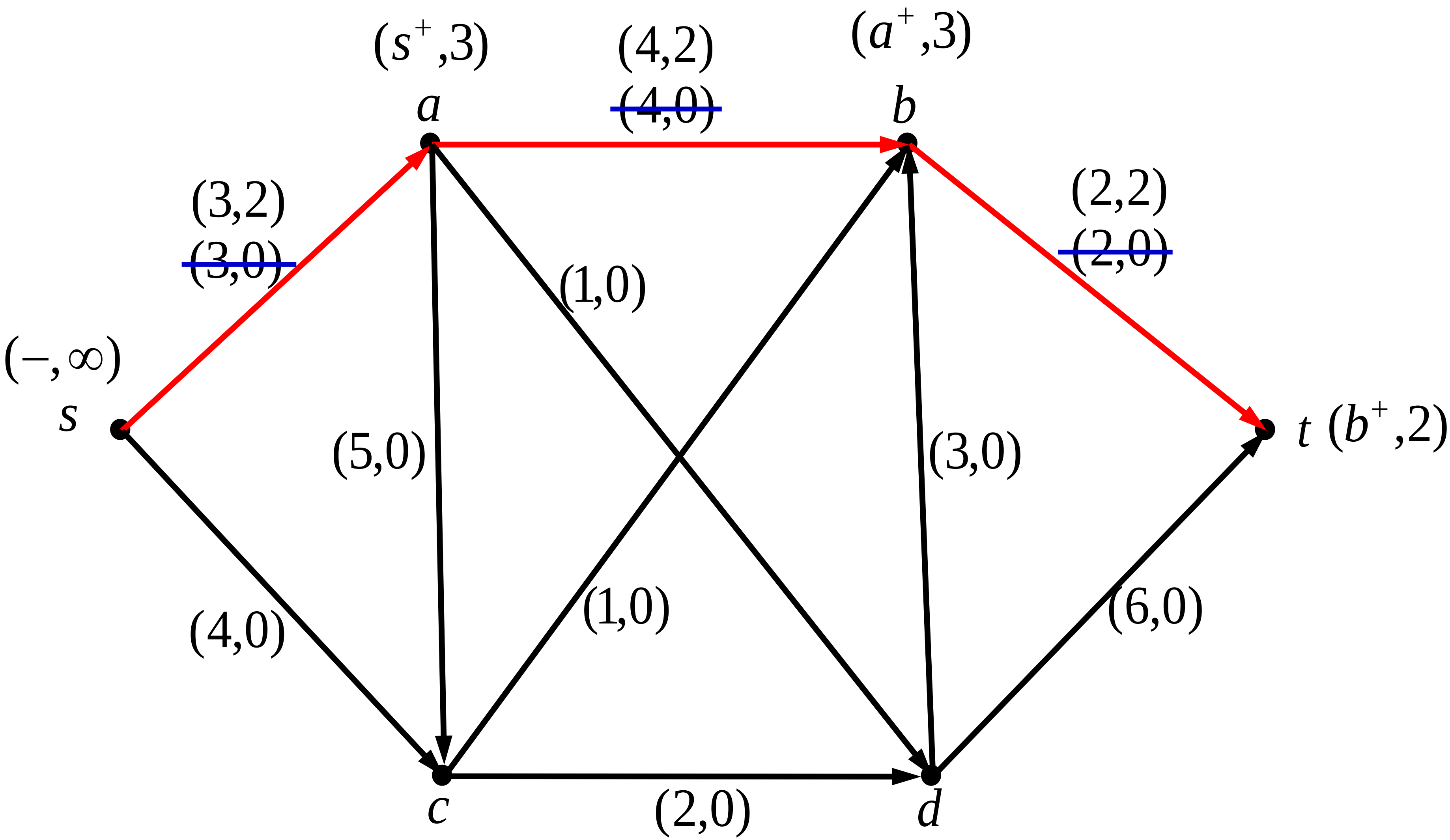
一次标号过程以标到t为终结条件

清華大學 Tsinghua University

Ford-Fulkerson最大流标号算法

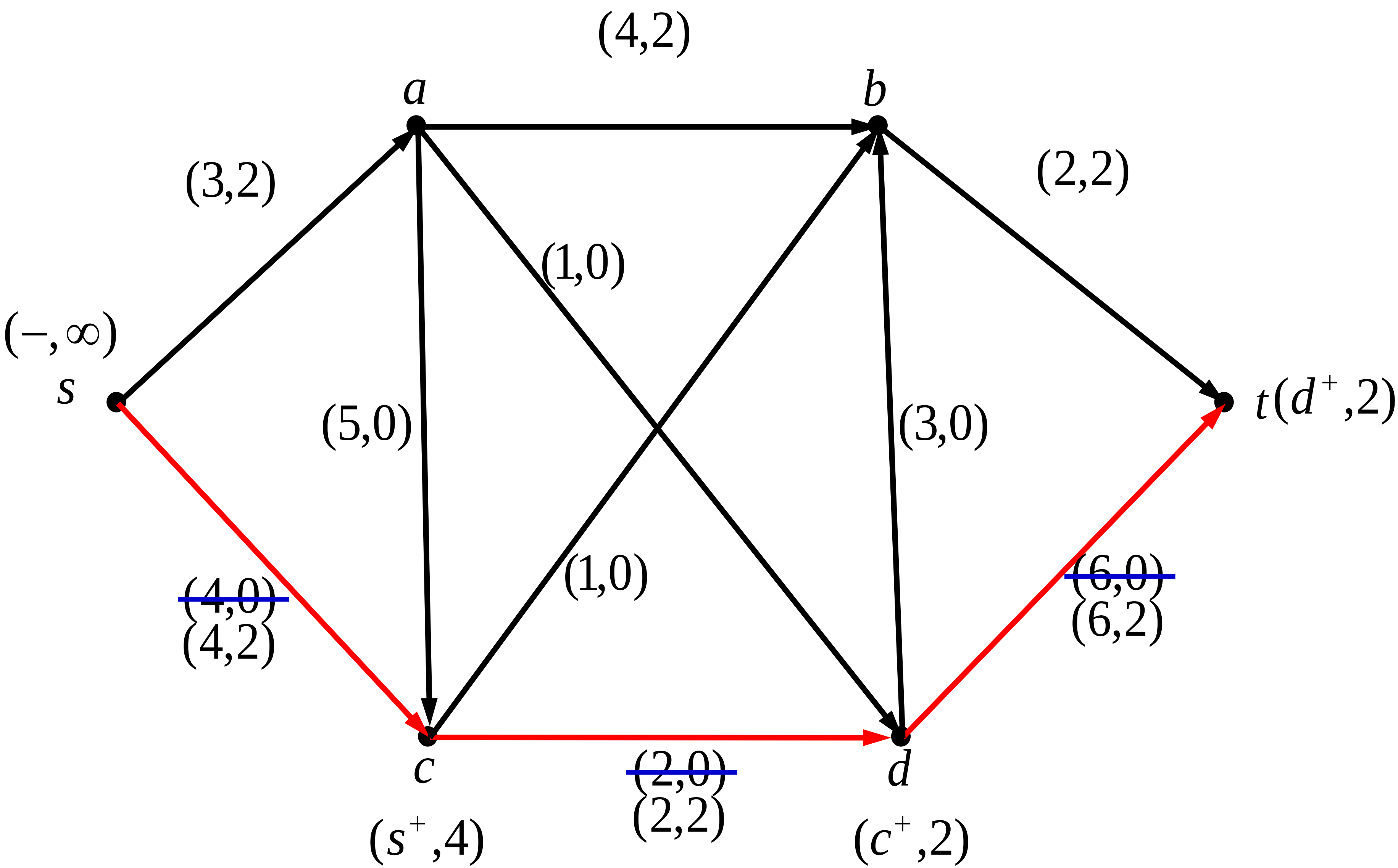
- 增流过程：
  - 对每条边都用二元组(c_{ij}, f_{ij})标记，分别表示该边的容量和实际流量
  - 在标号过程中，如果 t 得到了标号，则意味着图中存在增流路径，并可增流 δ_t 。
  - 将增流路径上的边标记进行修改，得到新的容许流分布 f' ，完成增流

例：



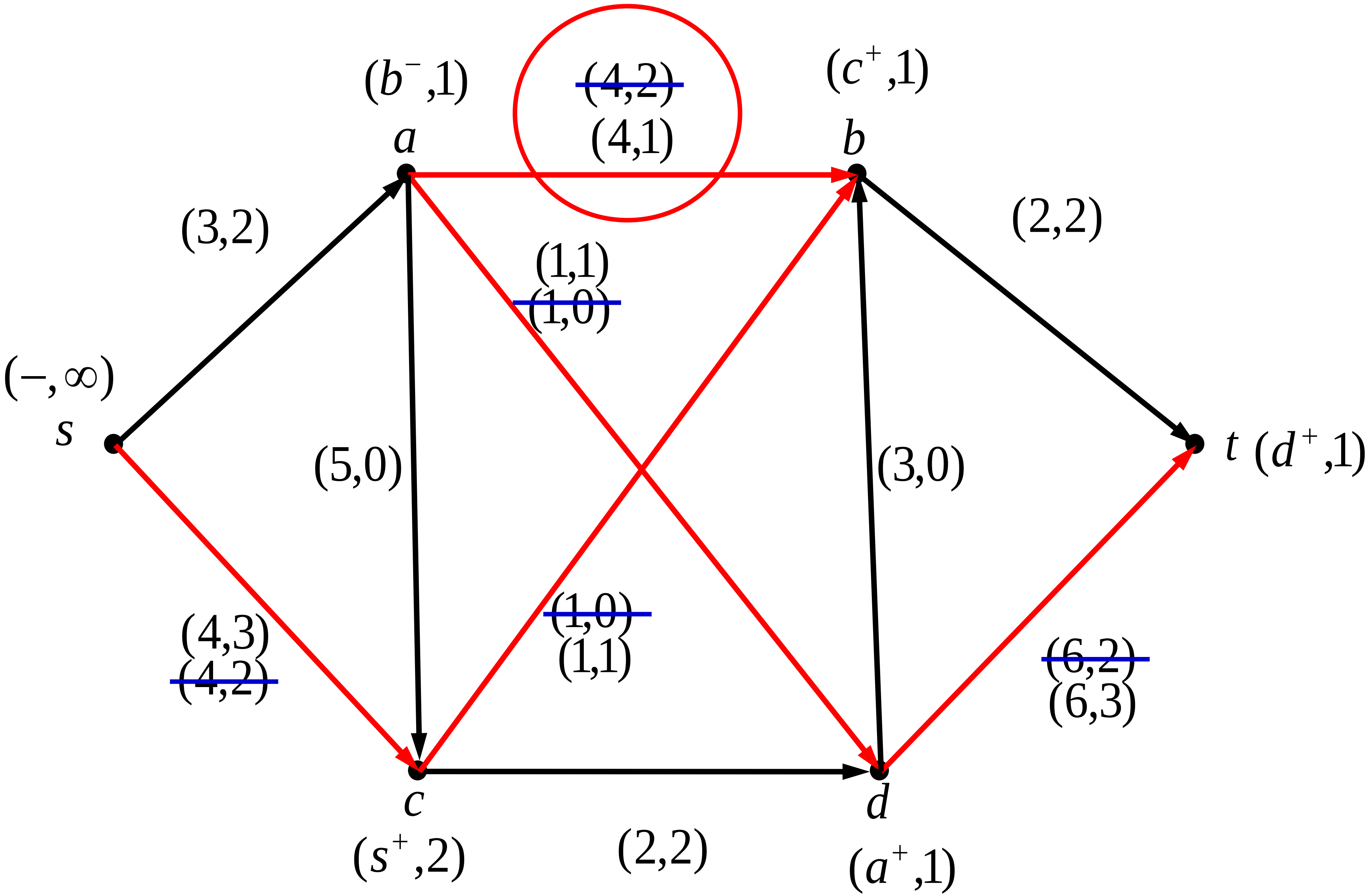
第一轮标号及增流过程结束，此时W＝2

例：



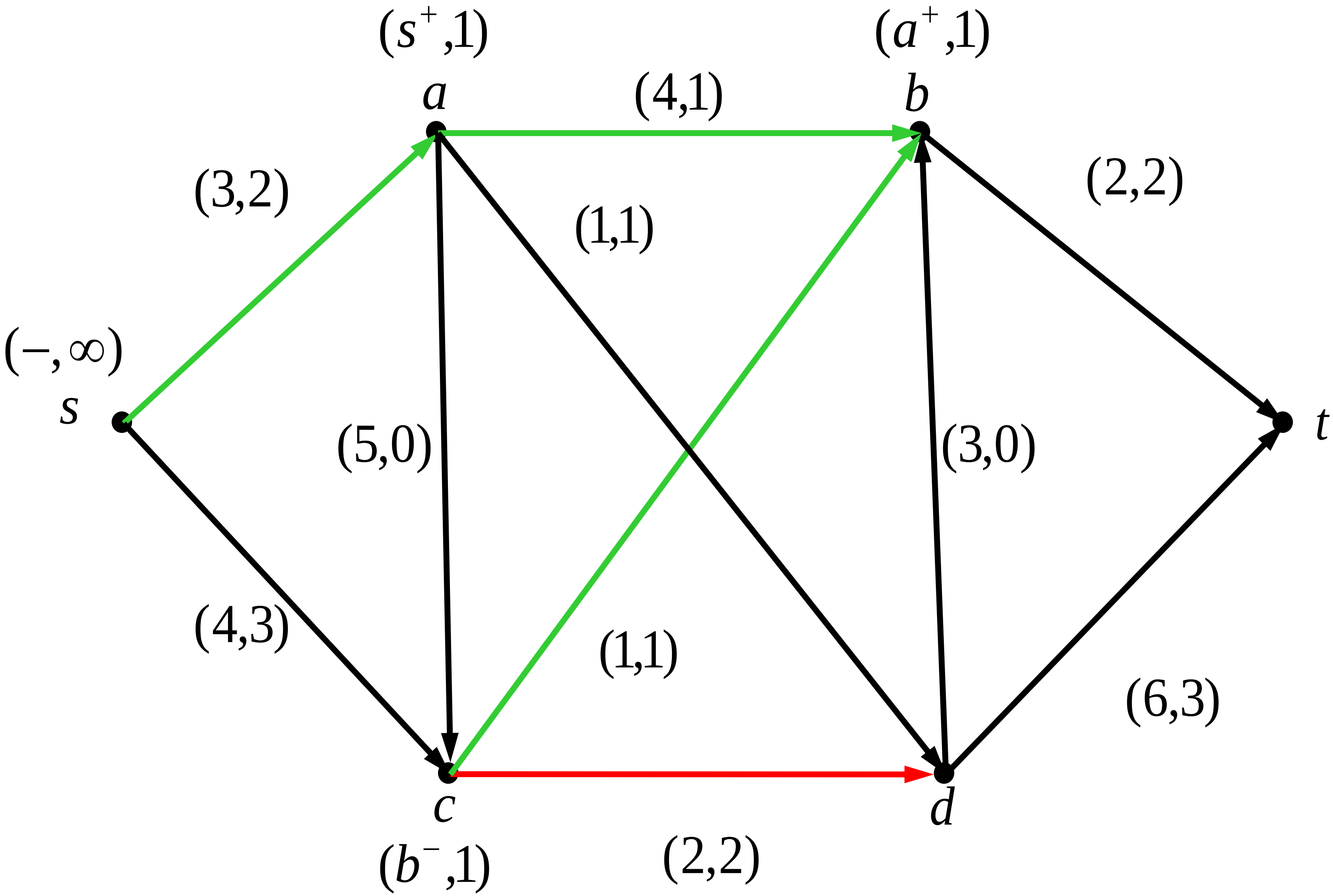
第二轮标号及增流过程结束，此时W＝4

例：

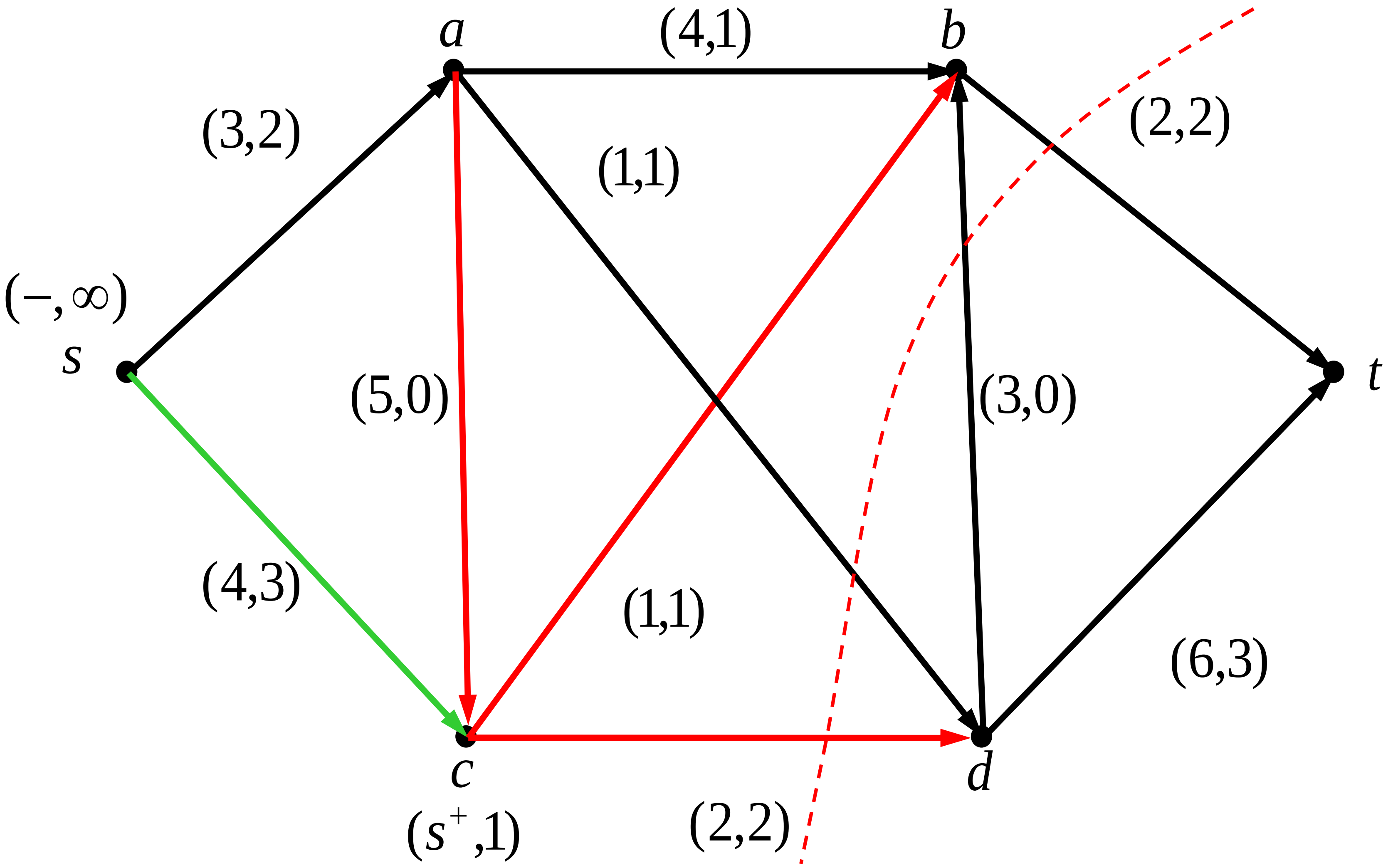


第三轮标号及增流过程结束，此时W＝5

例：



例：



$$S=\{s,a,b,c\},\quad \overline{S}=\{d,t\}\qquad (S,\overline{S})=\{(b,t),(a,d),(c,d)\},\quad C(S,\overline{S})=5$$

算法结束，此时max(w) = 5

Ford-Fulkerson最大流标号算法描述：

1. 任选一容许流分布f（一般为零流量）

标号过程：

2. 给 s 标号为 $(-,\infty)$
3. 若存在一个未标结点v，可通过正、反向标号得到标号，标之并转4，否则转7（算法结束）
4. 若$v=t$，转5，否则转3

增流过程：

5. 设v的标号是(d_v,δ_v)，
   1. 若 $d_v=u^+$ ，则令 $f(u,v)=f(u,v)+\delta_t$
   2. 若 $d_v=u^-$ ，则令 $f(v,u)=f(v,u)-\delta_t$
6. 若$u=s$，删去全部标号并转2，否则令$v=u$，转5
7. 结束

Ford-Fulkerson最大流标号算法

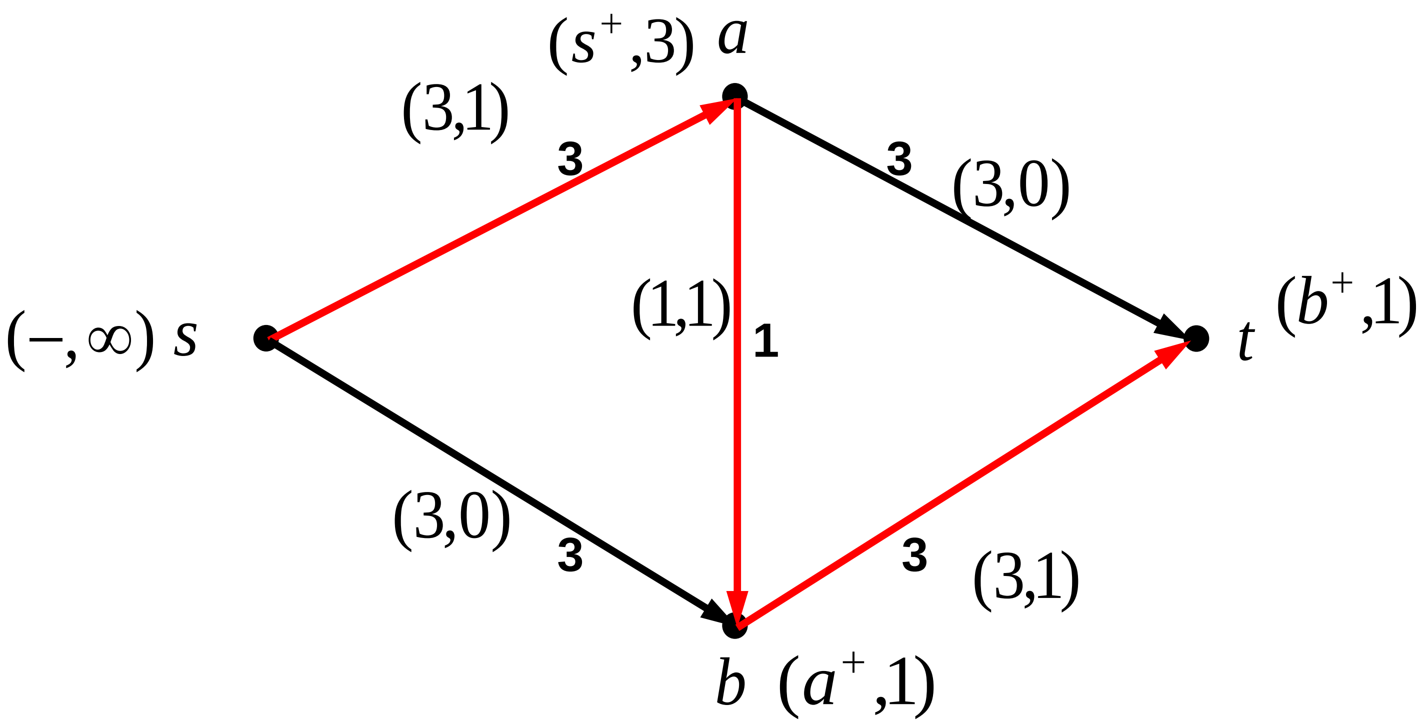
- 思考：
  - Ford-Fulkerson算法的复杂度如何？
    - 寻找增流路径

      $s \rightarrow t$ 标号过程（随机标号）

      最多经过n-1 次标号标到 t 结点
    - 修改流量分布

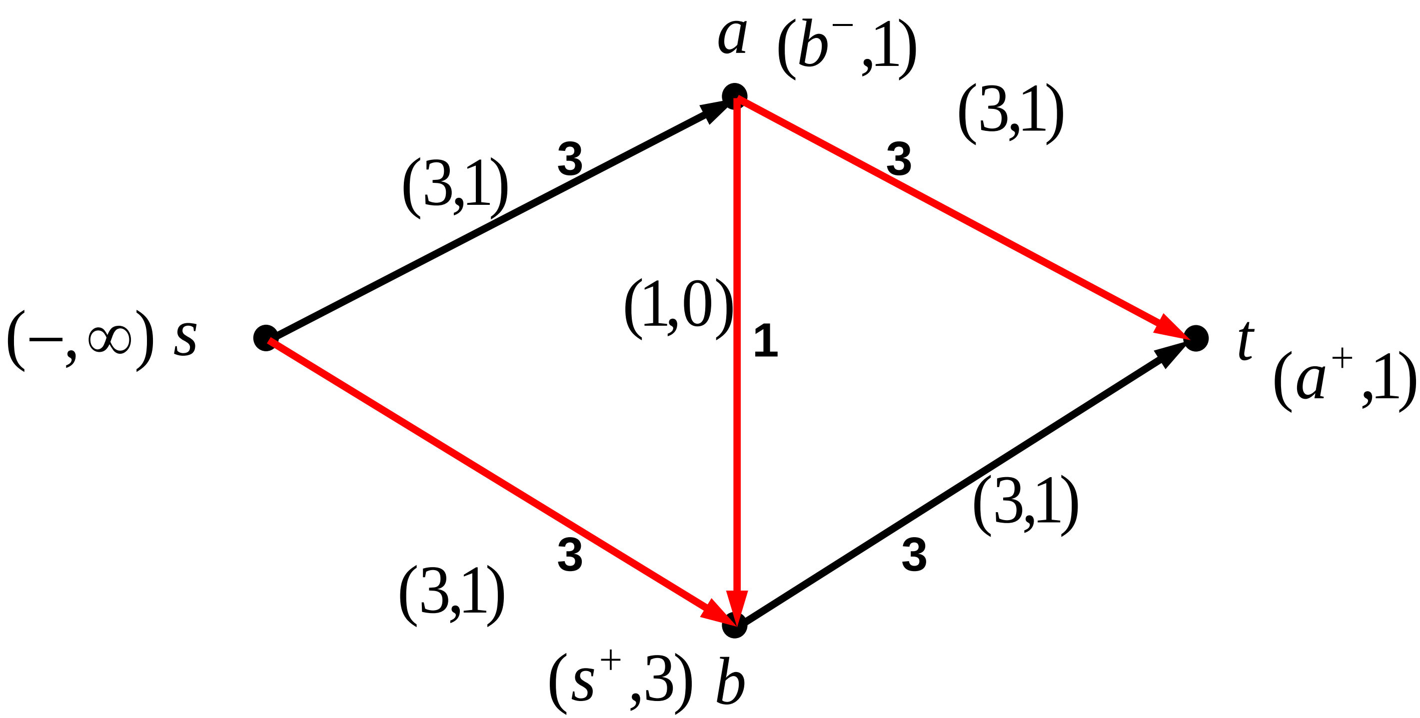
      最多m次流量值修改
    - 循环往复，直至无法找到增流路径。
    - 循环次数？

例：



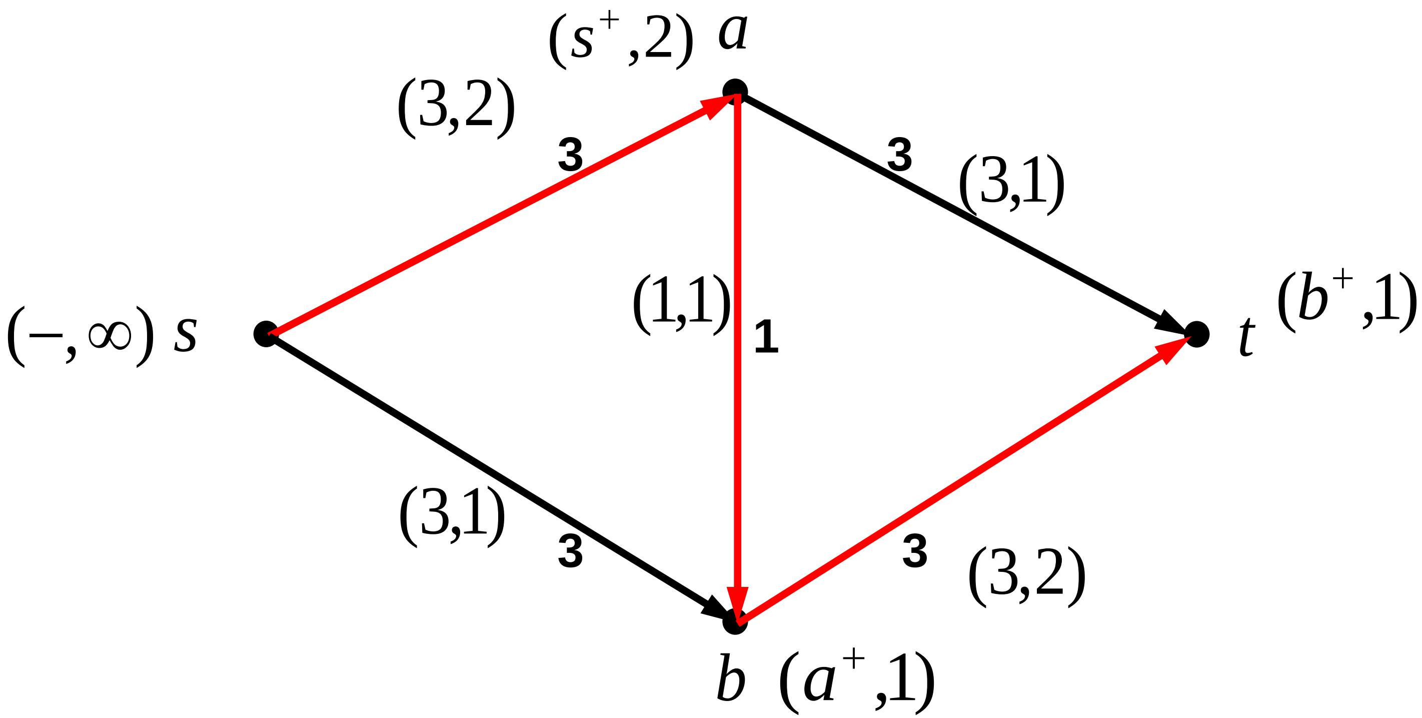
第1次增流！

例：



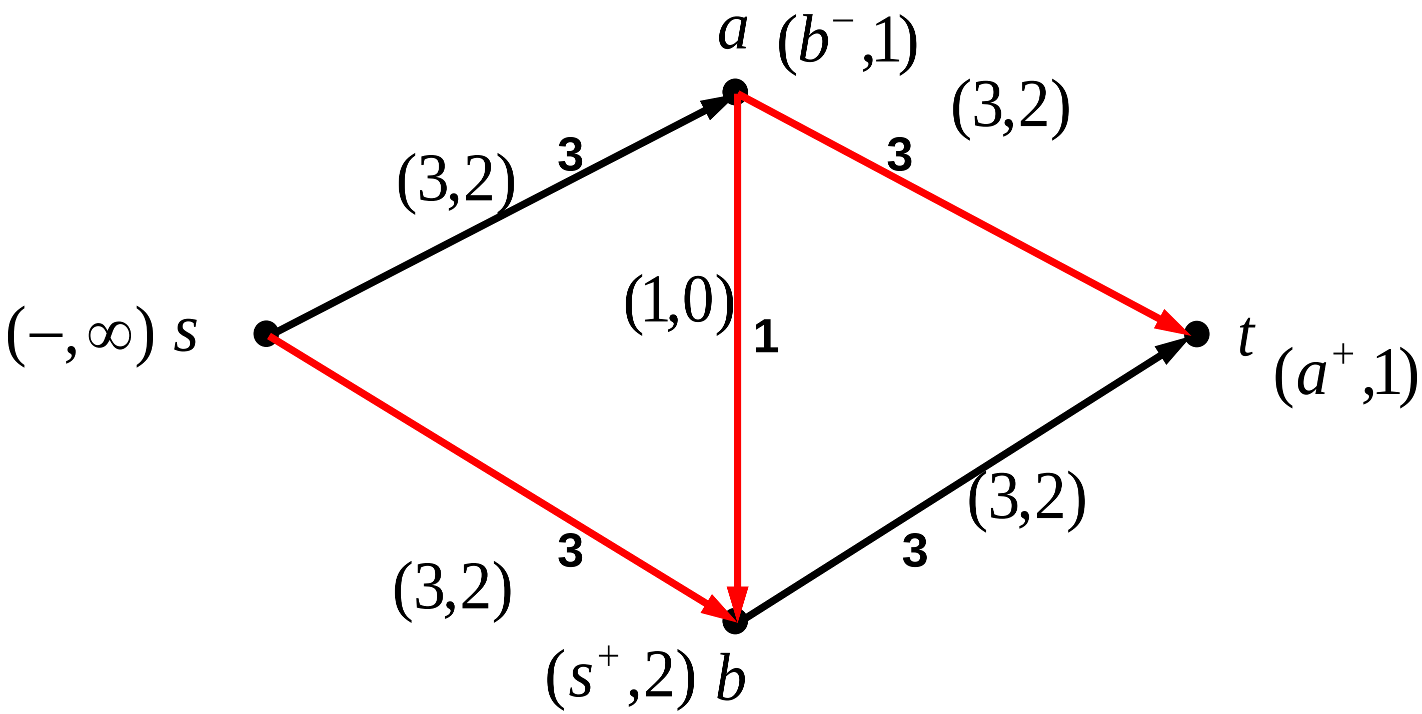
第2次增流！

例：



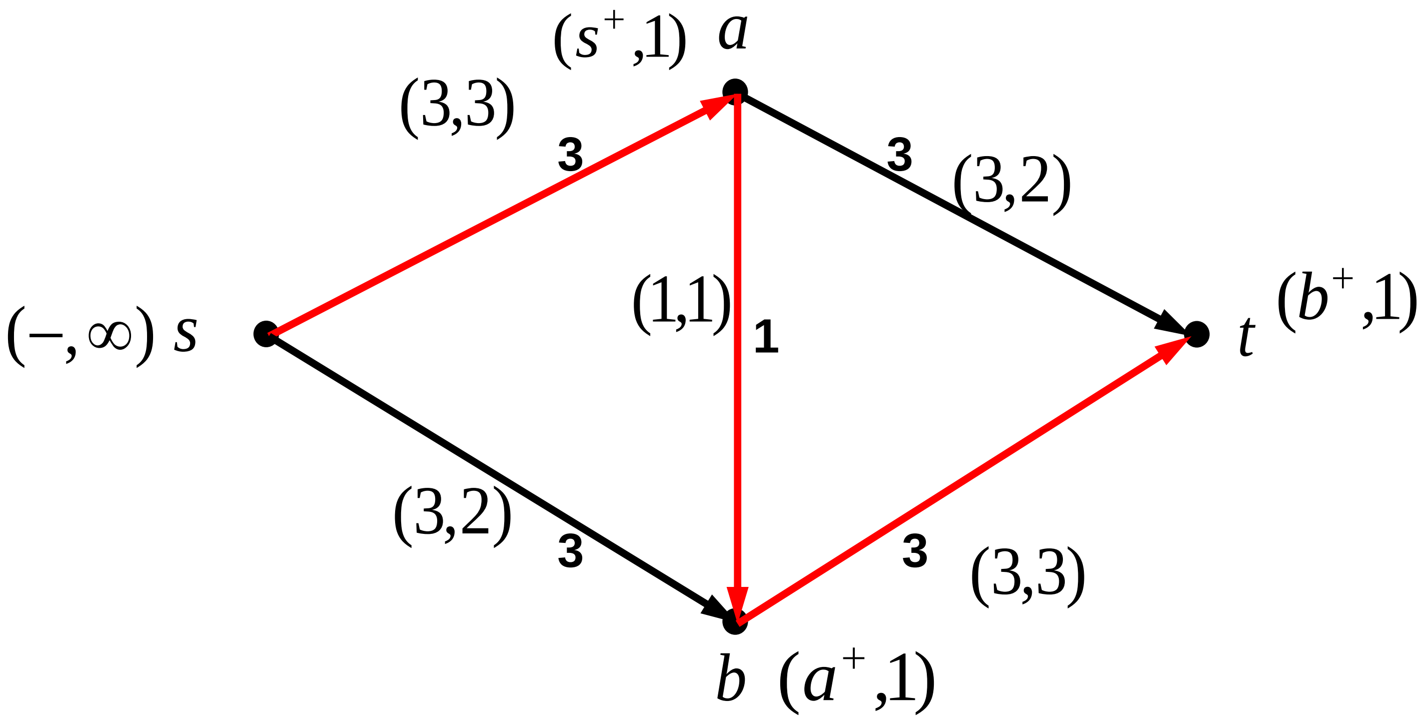
第3次增流！

例：



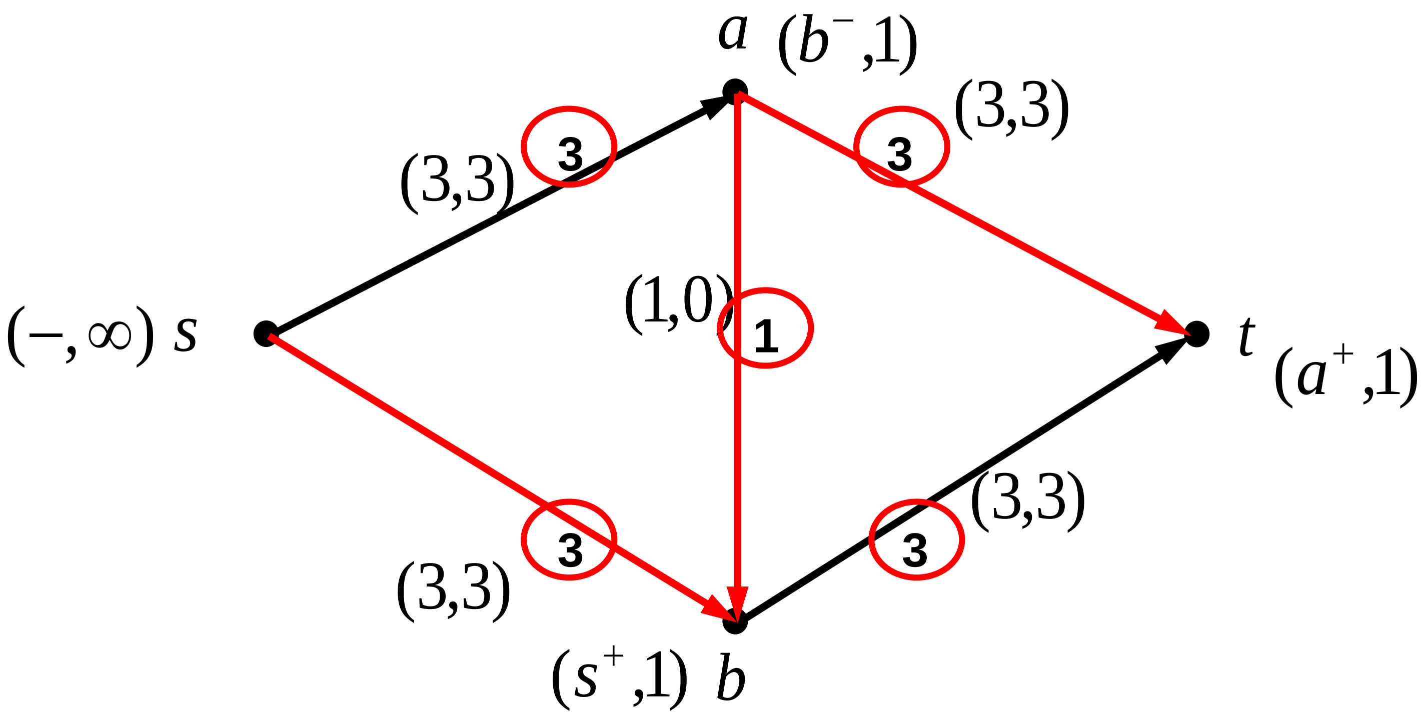
第4次增流！

例：



第5次增流！

例：



第6次增流！完成！

Ford-Fulkerson最大流标号算法

- Ford-Fulkerson算法复杂度
  - 和网络规模关系不确定
  - 和边容量参数有一定关系
  - 在特定情况下，边容量参数为无理数时，算法需要进行无数次增流过程

主要内容

- 5.1 二分图的最大匹配
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- **5.7 最大流的Edmonds-Karp算法**
- 5.8 最小费用流

最大流的Edmonds-Karp算法

- Edmonds-Karp算法思想：
  - 增流与标号思想基本等同于F-F算法
  - 每次沿最短路增流
  - 最短路径：广探法（宽度优先搜索）

Ford-Fulkerson最大流标号算法描述：

1. 任选一容许流分布f（一般为零流量）

标号过程：

2. 给 s 标号为 $(-,\infty)$
3. 若存在一个未标结点v，可通过正、反向标号得到标号，标之并转4，否则转7（算法结束）
4. 若$v=t$，转5，否则转3

增流过程：

5. 设v的标号是(d_v,δ_v) ，
   1. 若 $d_v=u^+$ ，则令 $f(u,v)=f(u,v)+\delta_t$
   2. 若 $d_v=u^-$ ，则令 $f(v,u)=f(v,u)-\delta_t$
6. 若$u=s$，删去全部标号并转2，否则令$v=u$，转5
7. 结束

Edmonds-Karp最大流标号算法描述：

1. 任选一容许流分布f（一般为零流量）

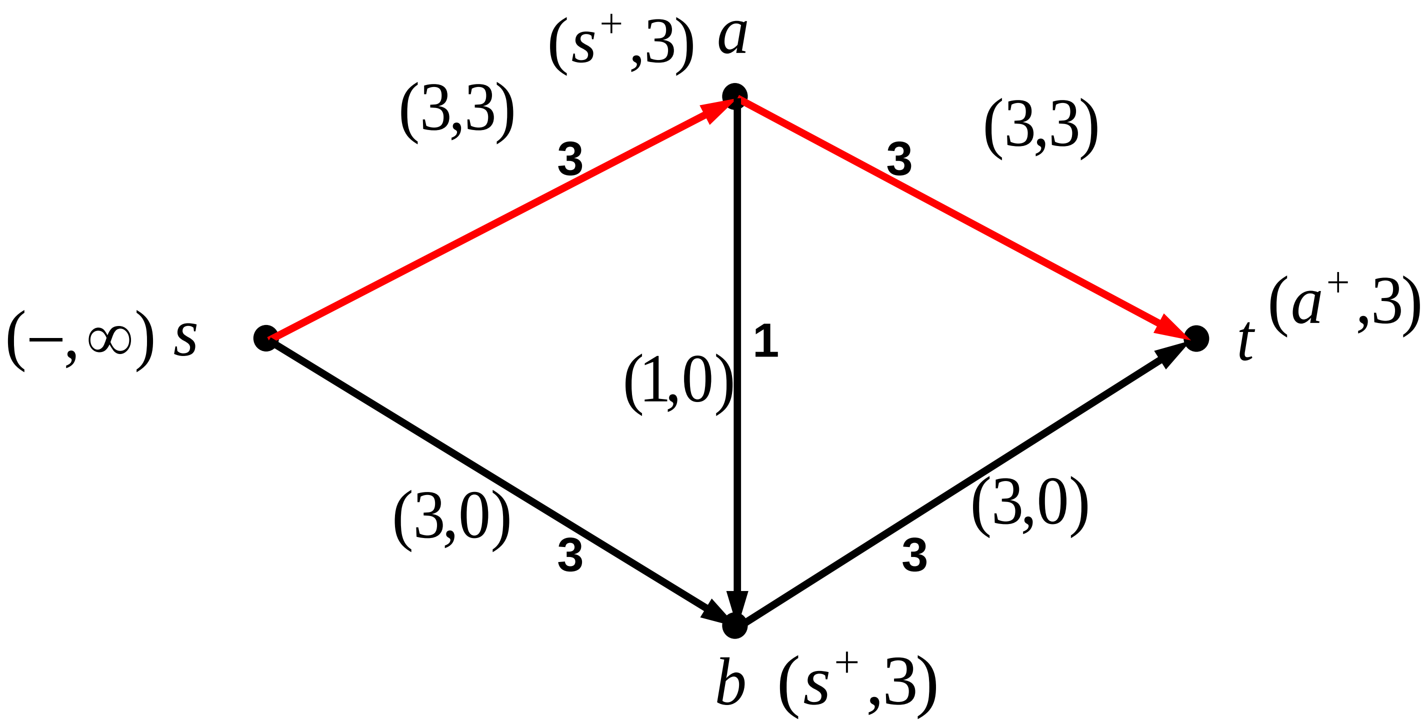
标号过程：

2. 给 s 标号为 $(-,\infty)$
3. 按照先标号先检查的原则，选择最早标号但尚未检查的结点u，检查其所有未标号的邻结点v，标之并转4，如果所有结点都已经检查，则转7（算法结束）
4. 若t得到了标号，则令$v=t$，转5，否则转3

增流过程：

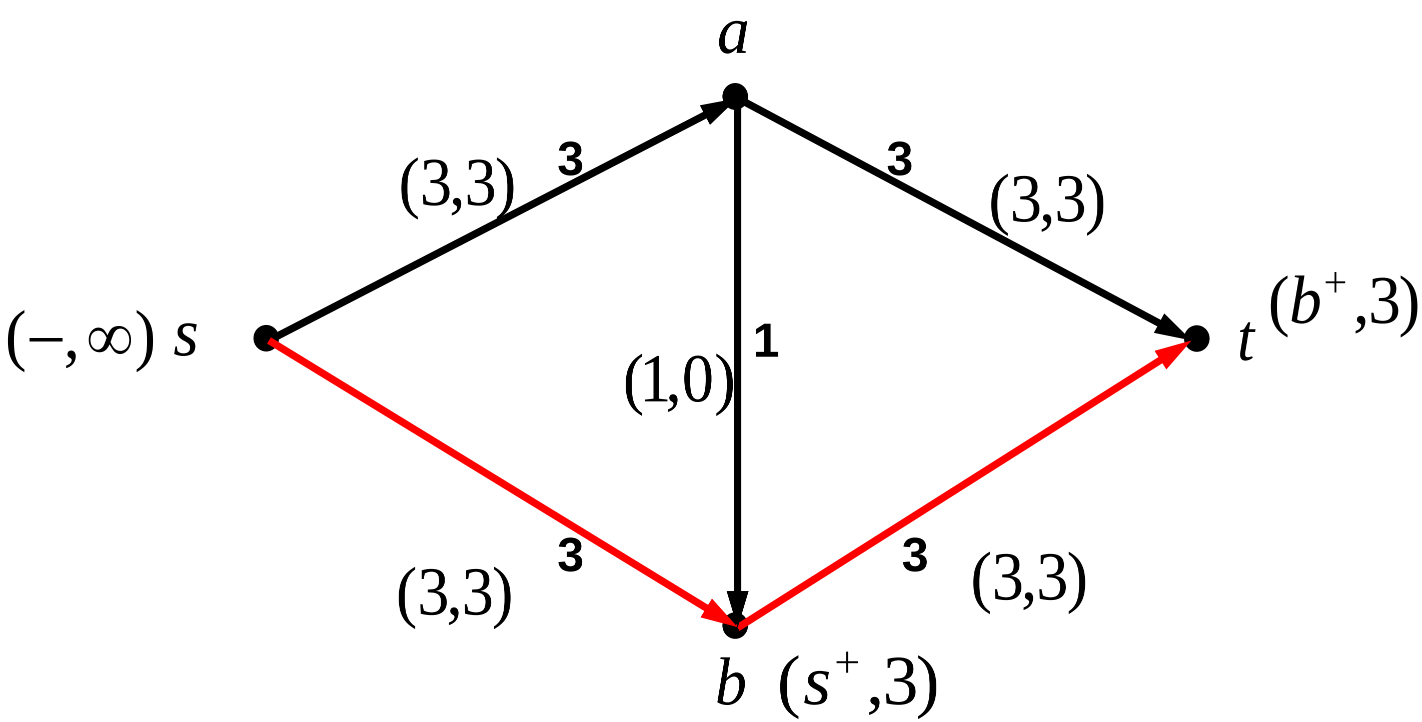
5. 设v的标号是(d_v,δ_v)，
   1. 若 $d_v=u^+$ ，则令 $f(u,v)=f(u,v)+\delta_t$
   2. 若 $d_v=u^-$ ，则令 $f(v,u)=f(v,u)-\delta_t$
6. 若$u=s$，删去全部标号并转2，否则令$v=u$，转5
7. 结束

例：



第1次增流！

例：



第2次增流！结束！

最大流的Edmonds-Karp算法

- 思考：
  - Edmonds-Karp算法的复杂度如何？
    - 寻找增流路径

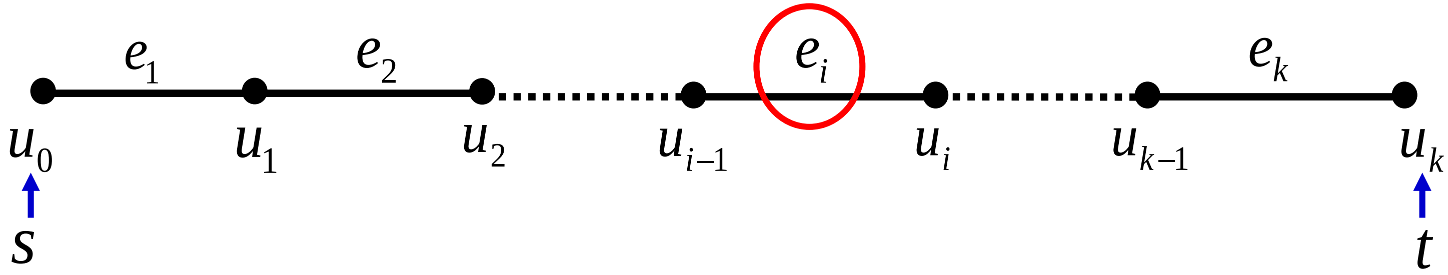
      $s \rightarrow t$ 标号过程（广探法）

      最多经过n-1次标号标到 t 结点
    - 修改流量分布

      最多m次流量值修改
    - 循环往复，直至无法找到增流路径。
    - 循环次数？ $\rightarrow$ 增流路径条数

最大流的Edmonds-Karp算法

- 定理5.7.1 如果在Edmonds-Karp标号算法中，每条增流路径都是当前最短的增流路，则网络中的增流路不超过 $m(n+2)/2$ 条



设 f 是网络N中的容许流分布

路径 P_{st} 是一条增流路径，其流量增幅为 δ，此时必定有

边 e_i 满足 $\begin{cases} \delta = c(e_i) - f(e_i) & e_i \text{ 为向前边} \\ \text{或 } \delta = f(e_i) & e_i \text{ 为向后边} \end{cases}$

此时，称边 e_i 为该路径的瓶颈。

最大流的Edmonds-Karp算法

- 设标号法从网络N的初始流分布f_0开始，依据Edmonds-Karp算法依次构造网络N的容许流f_1，f_2，…

- 令$\lambda^i(u,v)$表示在容许流f_i分布下，从结点u到结点v的最短非饱和路径长度

Tsinghua University

最大流的Edmonds-Karp算法

- 引理5.7.1 对每个结点v及每个k =0,1,2,...

$$\lambda^k(s,v) \le \lambda^{k+1}(s,v) \qquad (1)$$

$$\lambda^k(v,t) \le \lambda^{k+1}(v,t) \qquad (2)$$

最大流的Edmonds-Karp算法

- 引理5.7.2 若$k<p$，且向前(后)边e是从f_k变为f_{k+1}的瓶颈，同时也是从f_p变为f_{p+1}的瓶颈，则一定存在l，满足$k<l<p$，并且边e为从f_l到f_{l+1}时增流路中的边，且在增流路中一定是向后(前)边。

最大流的Edmonds-Karp算法

- 引理5.7.3 在采用先标号先检查原则求网络的最大流时，如果边e是从 f_k 到 f_{k+1} 时增流路中的一条向前(后)边，后来又是从 f_l 到 f_{l+1} 时增流路的一条向后(前)边 $(k<l)$，则有

$$\lambda^l(s,t) \geq \lambda^k(s,t)+2$$

清華大学 Tsinghua University

最大流的Edmonds-Karp算法

- 定理5.7.1 如果在Edmonds-Karp标号算法中，每条增流路径都是当前最短的增流路，则网络中的增流路不超过 $m(n+2)/2$ 条

最大流的Edmonds-Karp算法

- 定理5.7.1 如果在Edmonds-Karp标号算法中，每条增流路径都是当前最短的增流路，则网络中的增流路不超过 $m(n+2)/2$ 条

证明：

- 设边e的方向都是从u到v，一个容许流分布的序列为 $f_{k1}, f_{k2}, \ldots$，其中$k1<k2<\cdots$，而且边e在 f_{ki} 中是向前边瓶颈。
- 由引理5.7.2可以断定，存在另一个序列 $l_1, l_2, \ldots$，满足 $k1<l1<k2<l2\cdots$，且e在 f_{li} 中为向后边。

最大流的Edmonds-Karp算法

- 由引理5.7.3　$\lambda^{ki}(s,t)+2\leq\lambda^{li}(s,t)$

  同时　$\lambda^{li}(s,t)+2\leq\lambda^{k(i+1)}(s,t)$

- 因此　$\lambda^{k1}(s,t)+4(j-1)\leq\lambda^{kj}(s,t)$
- 由于　$\lambda^{kj}(s,t)\leq n-1$，　$\lambda^{k1}(s,t)\geq 1$
- 所以　$j\leq\dfrac{n+2}{4}$
- 即e作为向前边最多能够充当瓶颈(n+2)/4次，类似可以作为向后边最多能够充当瓶颈(n+2)/4次。
- 由于网络中有m条边，故得证！

Tsinghua University

主要内容－总结

- 5.1 二分图的最大匹配
- 5.2 完全匹配
- 5.3 最佳匹配及其算法
- 5.4 最大基数匹配
- 5.5 网络流图
- 5.6 Ford-Fulkerson最大流标号算法
- 5.7 最大流的Edmonds-Karp算法
- 5.8 最小费用流

作业

- 课后 10 、 12
- 第二次实验，会布置在网络学堂
- 下次课：习题课