



第五章 匹配与网络流II

计算机系网络所：张小平





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



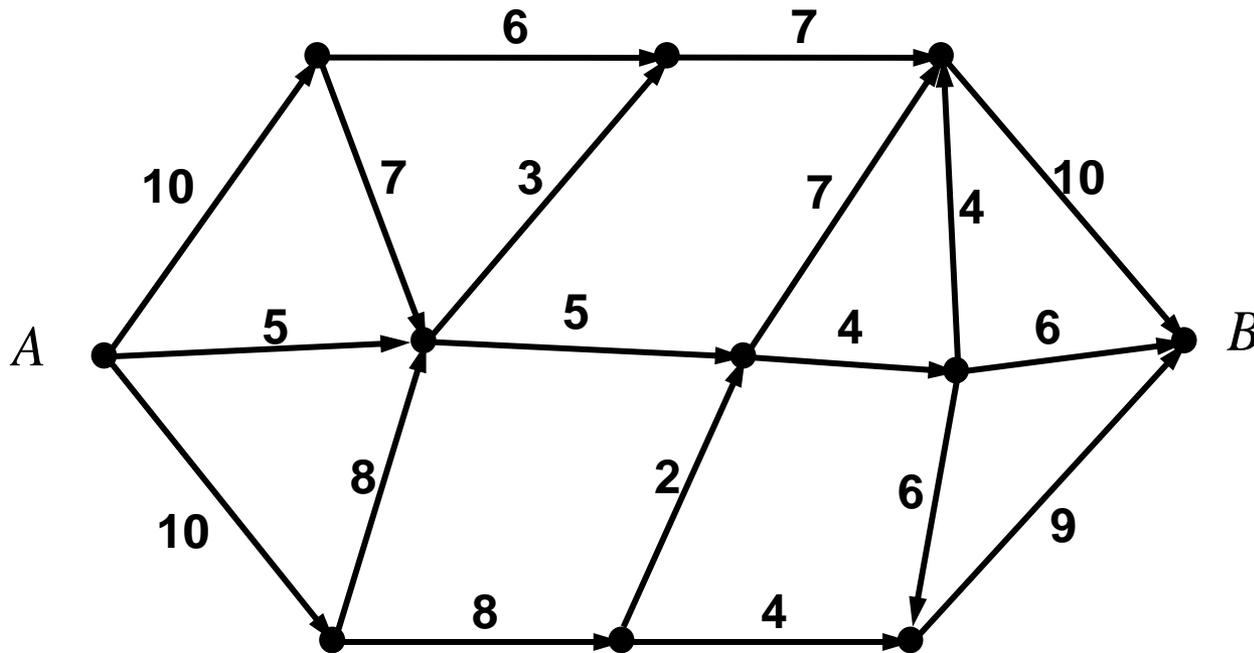
网络流图

- 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

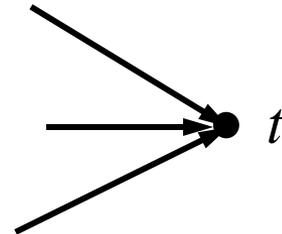
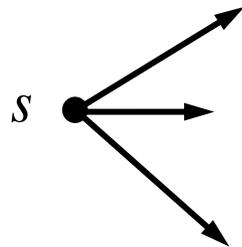
例：城市间公路网建设





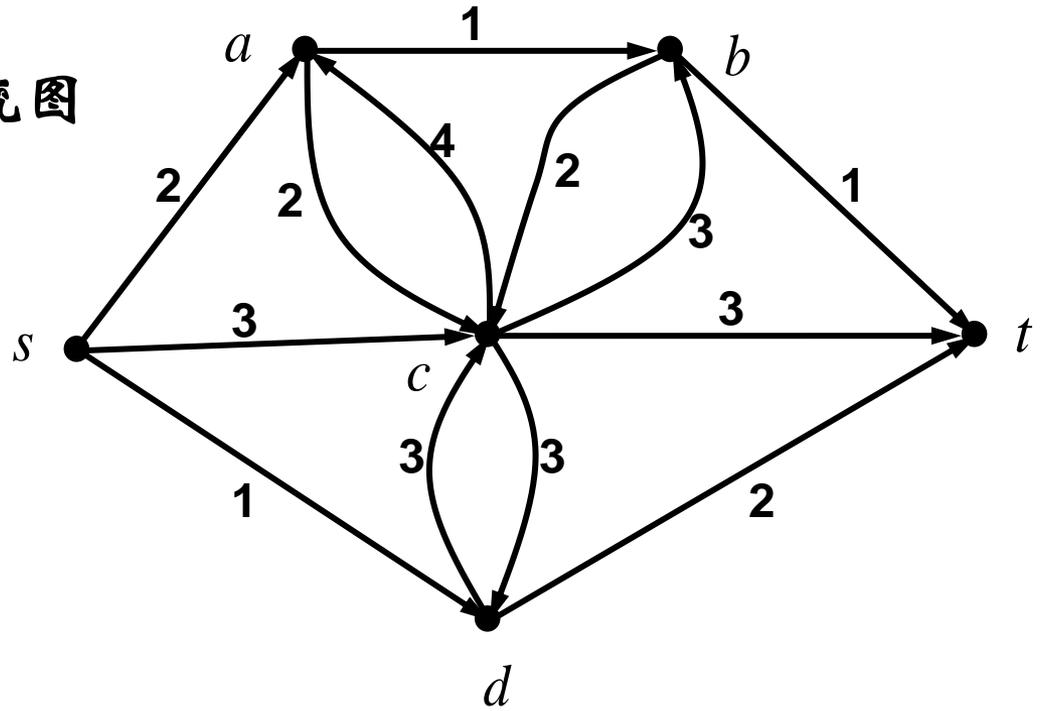
网络流图

- 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

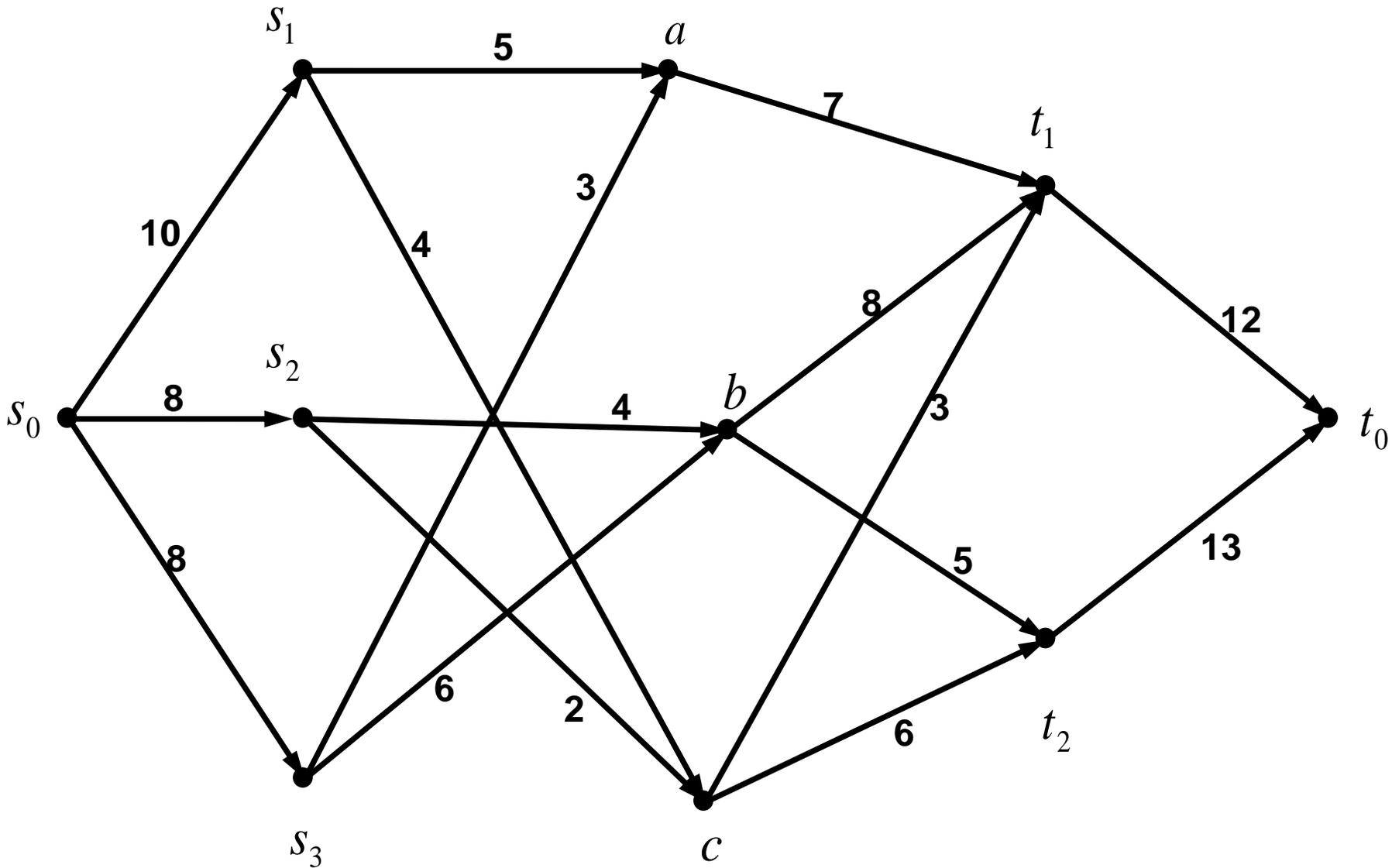


此类网络可以看作是某种产品

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





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. $f_{ij} \leq c_{ij}$, $e_{ij} \in N$ \longleftrightarrow 每条边的流量不超过边容量

2. $\sum_j f_{ij} = \sum_k f_{ki}$, $i \neq s, t$ \longleftrightarrow 注入 i 结点的流量总和等于结点 i 发出的流量总和

3. $\sum_j f_{sj} = \sum_k f_{kt} = w$ \longleftrightarrow 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



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**

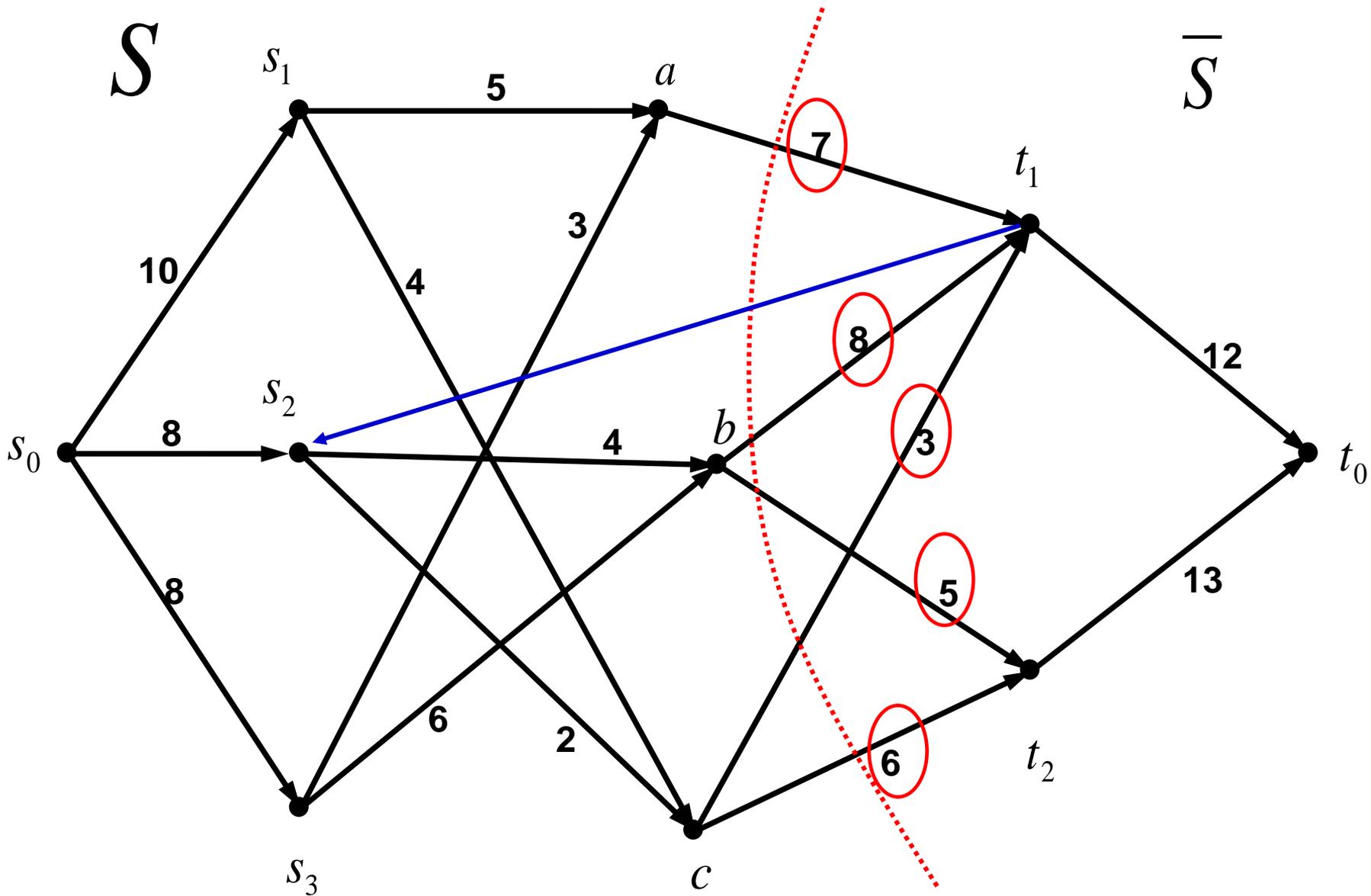


网络流图

- 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

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



割切和割集的关系?

网络最大流和割切的关系?



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

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

证明：

- 设 f 是给定网络的任一允许流分布，则满足

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

$$\sum_j (f_{ij} - f_{ji}) = 0 \quad i \neq s, t \quad j \in V \quad (2)$$



网络流图

— 任给一个割切 (S, \bar{S}) , 满足 $s \in S, t \in \bar{S}$, 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

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



网络流图

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

故
$$w = \sum_{\substack{i \in S \\ j \in \bar{S}}} (f_{ij} - f_{ji}) \leq \sum_{\substack{i \in S \\ j \in \bar{S}}} (f_{ij}) \leq \sum_{\substack{i \in S \\ j \in \bar{S}}} c_{ij} = C(S, \bar{S})$$

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

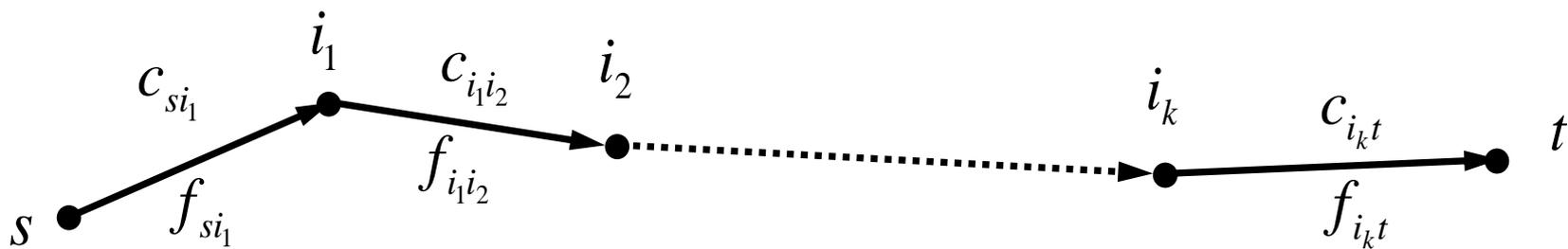
$$\max(w) \leq \min C(S, \bar{S})$$



网络流图

思考：

- 如果网络的允许流并不是最大流，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！

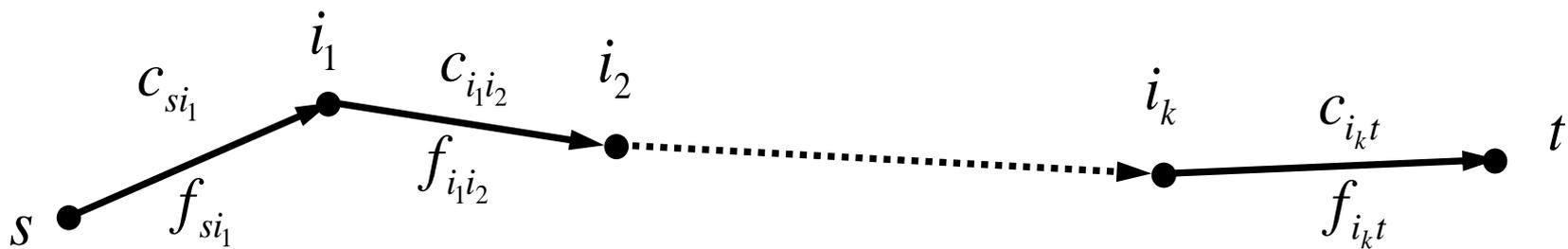


路径 P_{st} ，每条边的方向都是从 i_j 到 i_{j+1} ，称为**向前边**

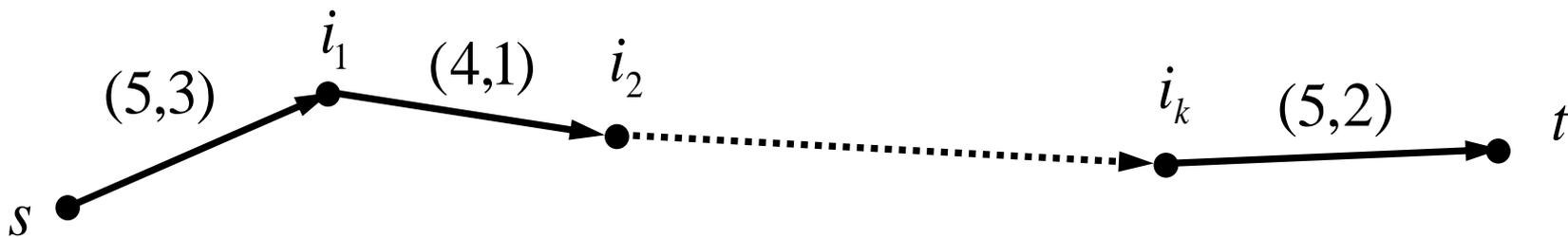
假如每条边上都有 $f_{ij} < c_{ij}$ ，则令 $\delta = \min_{e_{ij} \in P_{st}} (c_{ij} - f_{ij})$ ，

此时令每条边的流量都增加 δ ，

结果仍然是网络的允许流分布，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： $a = c_{ij}$ ， $b = f_{ij}$

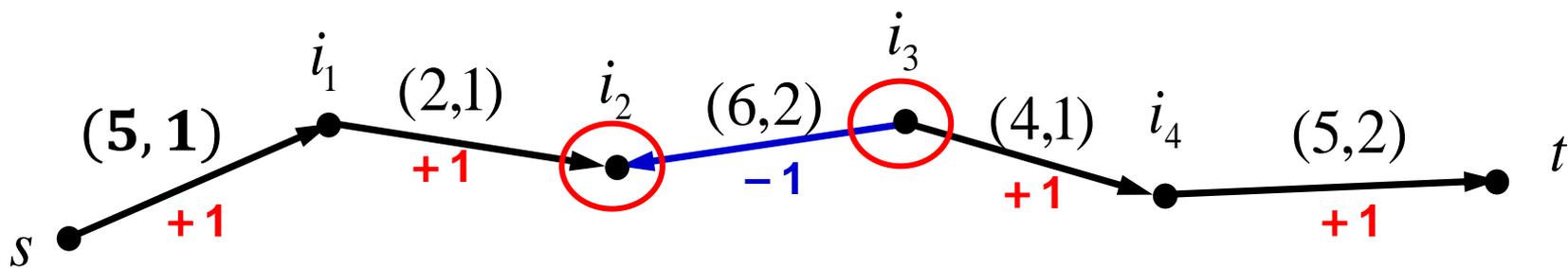




网络流图

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

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

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

结果仍然是网络的允许流分布, 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

直观表述：

- 如果从 s 出发，沿途的向前边都不饱和，直到 t 都是如此，那么该路径一定可增流。
- 如果从 s 出发，沿途向前边不饱和，向后边流量大于零，直到 t 都是如此，那么该路径一定可以增流



网络流图

- 思考:

- 网络流图中, 通过不断的增流, 可以达到最大流量。

- 那么, 最大流量究竟是多少?



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

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



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

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

证明：

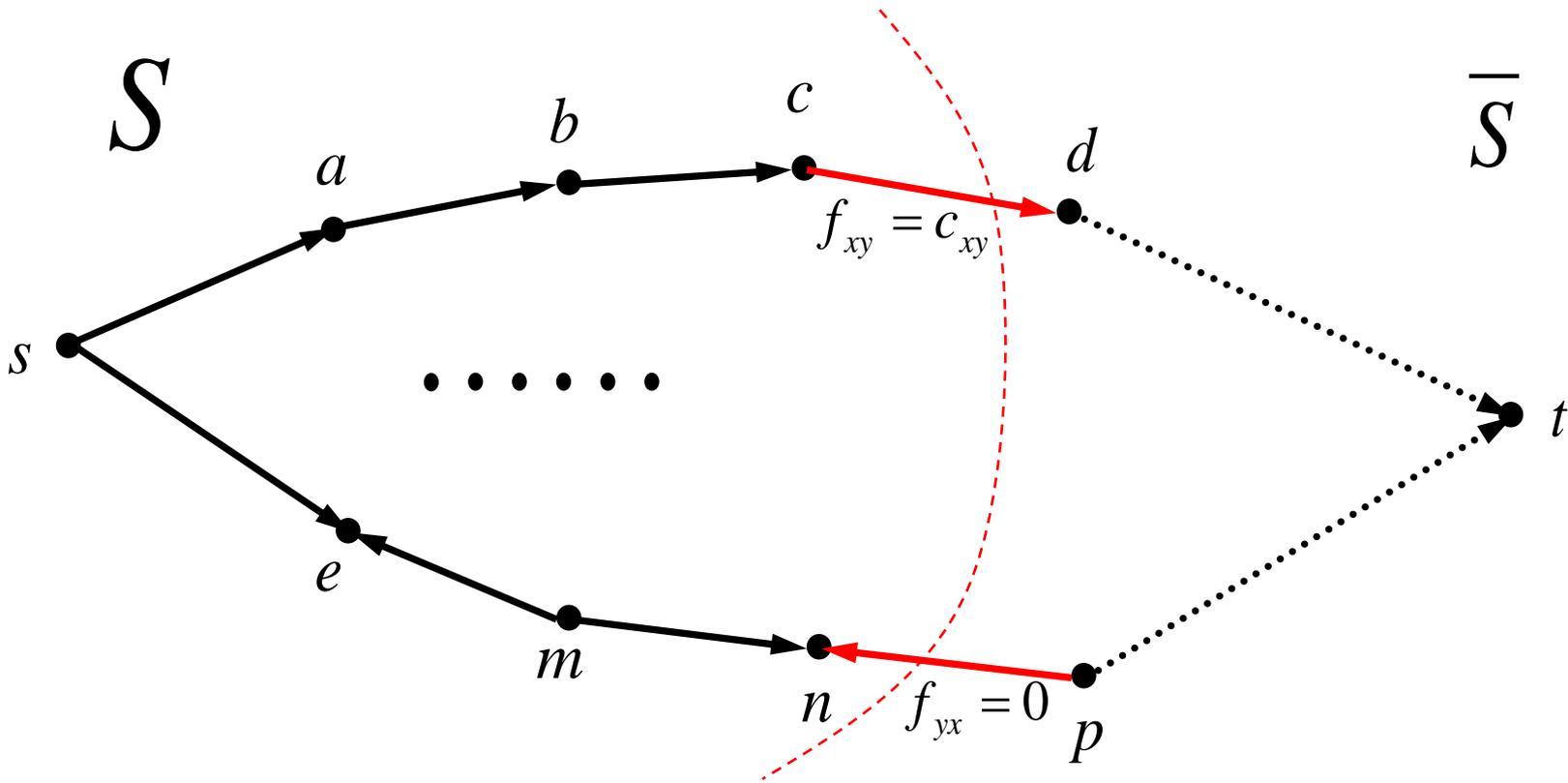
- 设网络中已经存在一个最大的允许流 f

- 定义一个割切 (S, \bar{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 $y \in S$



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

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

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

- 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 是最大允许流分布矛盾

- 故 $\bar{S} \neq \Phi$ 。

- 据该割切的定义方法，对于任意跨于 S 和 \bar{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

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

- 又根据定理 5.5.1 $\max(w) \leq \min C(S, \bar{S})$

- 故 $\max(w) = \min C(S, \bar{S})$

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

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



网络流图

思考：

- 如何确定一个网络流图的最大流？
- 不断寻找增流路径！
- 定理5.5.2的证明过程，实质上给出了一个寻找增流路径的方法



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, δ) ，其中， v 表示标号从哪个结点来， δ 表示可增流的量，“ \pm ”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 $\delta_v = \min(\delta_u, c(e) - f(e))$
 - 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 $\delta_v = \min(\delta_u, f(e))$

一次标号过程以标到 t 为终结条件

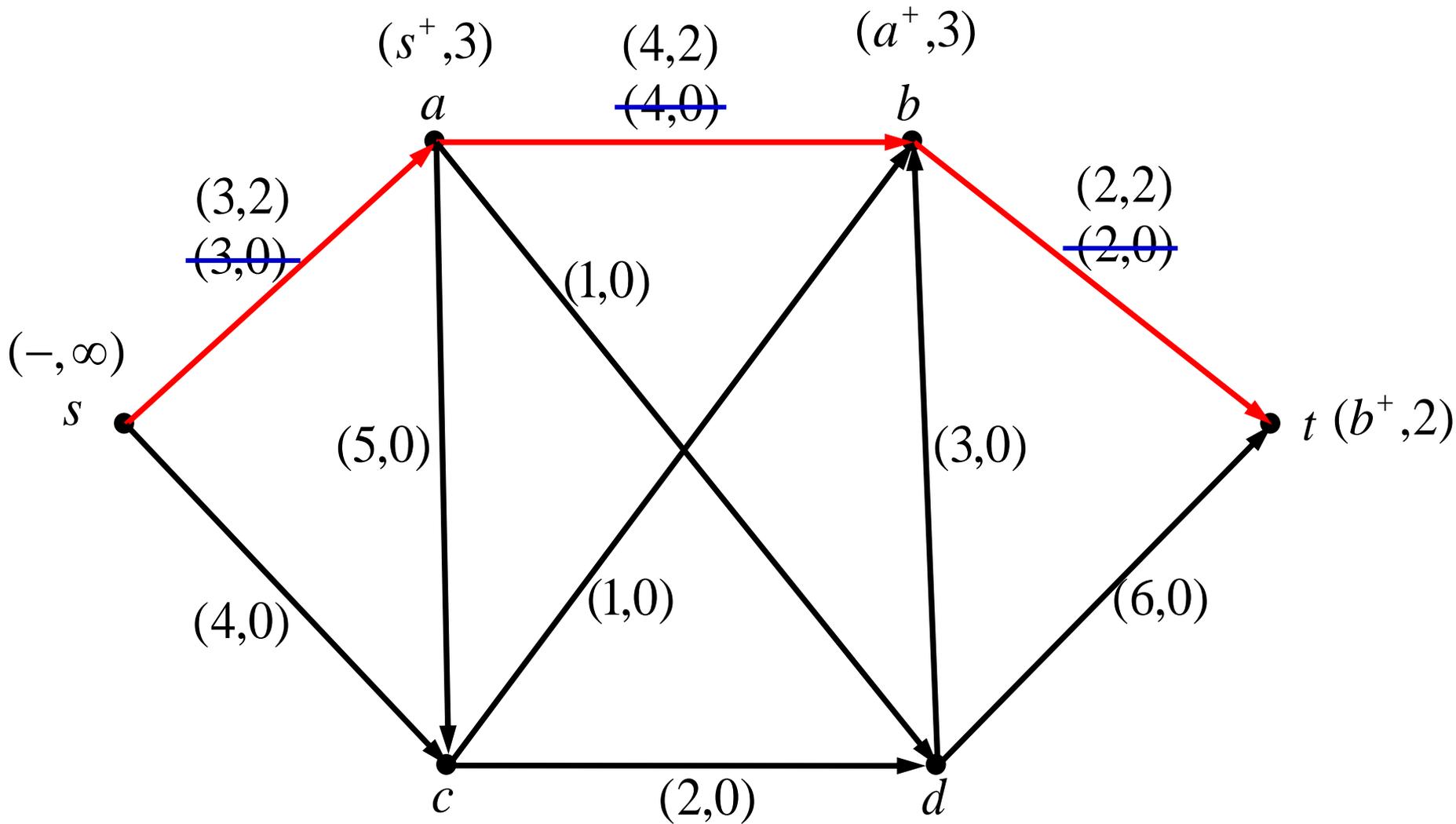




Ford-Fulkerson最大流标号算法

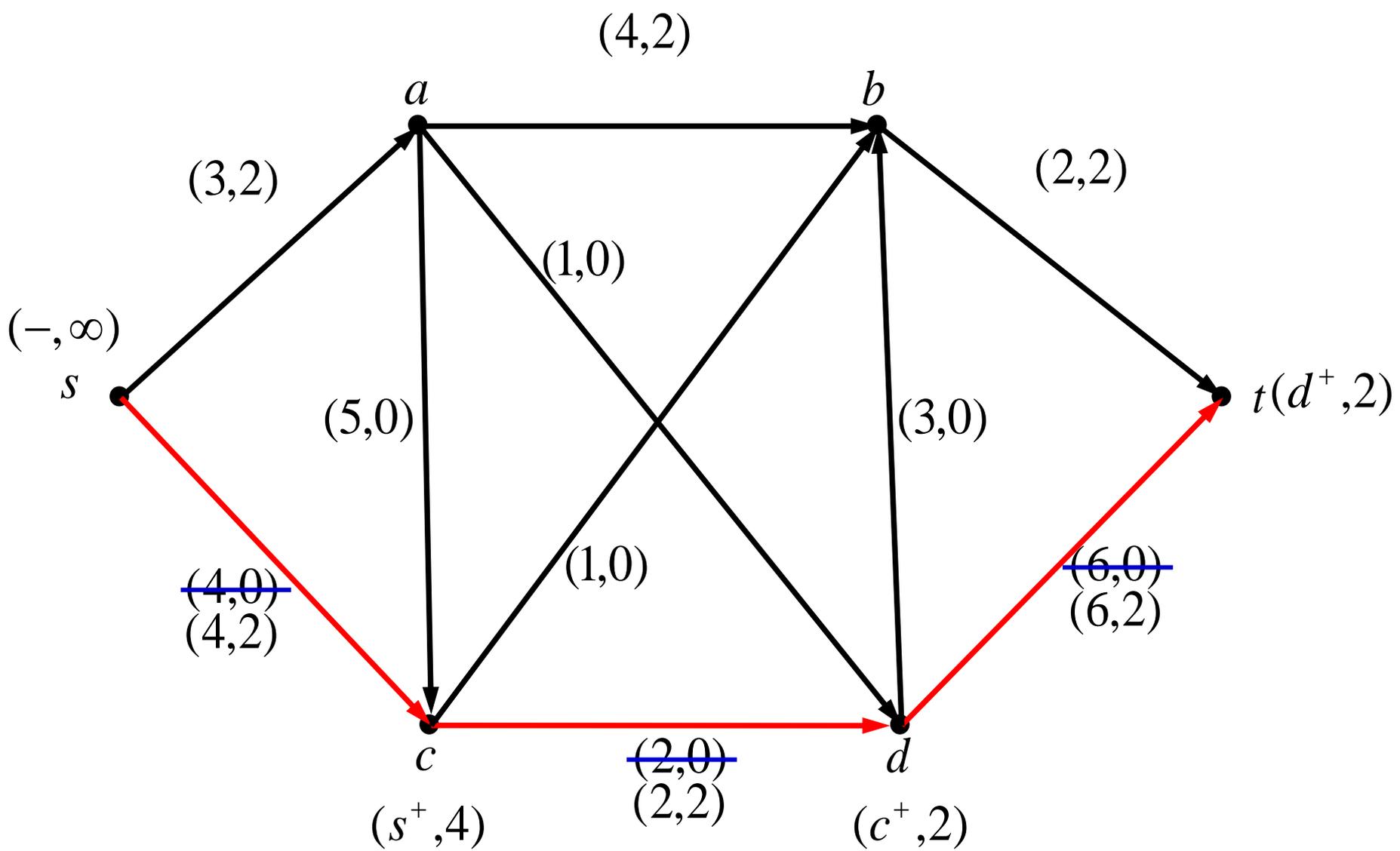
- 增流过程：
 - 对每条边都用二元组 (c_{ij}, f_{ij}) 标记，分别表示该边的容量和实际流量
 - 在标号过程中，如果 t 得到了标号，则意味着图中存在增流路径，并可增流 δ_t 。
 - 将增流路径上的边标记进行修改，得到新的允许流分布 f' ，完成增流

例：



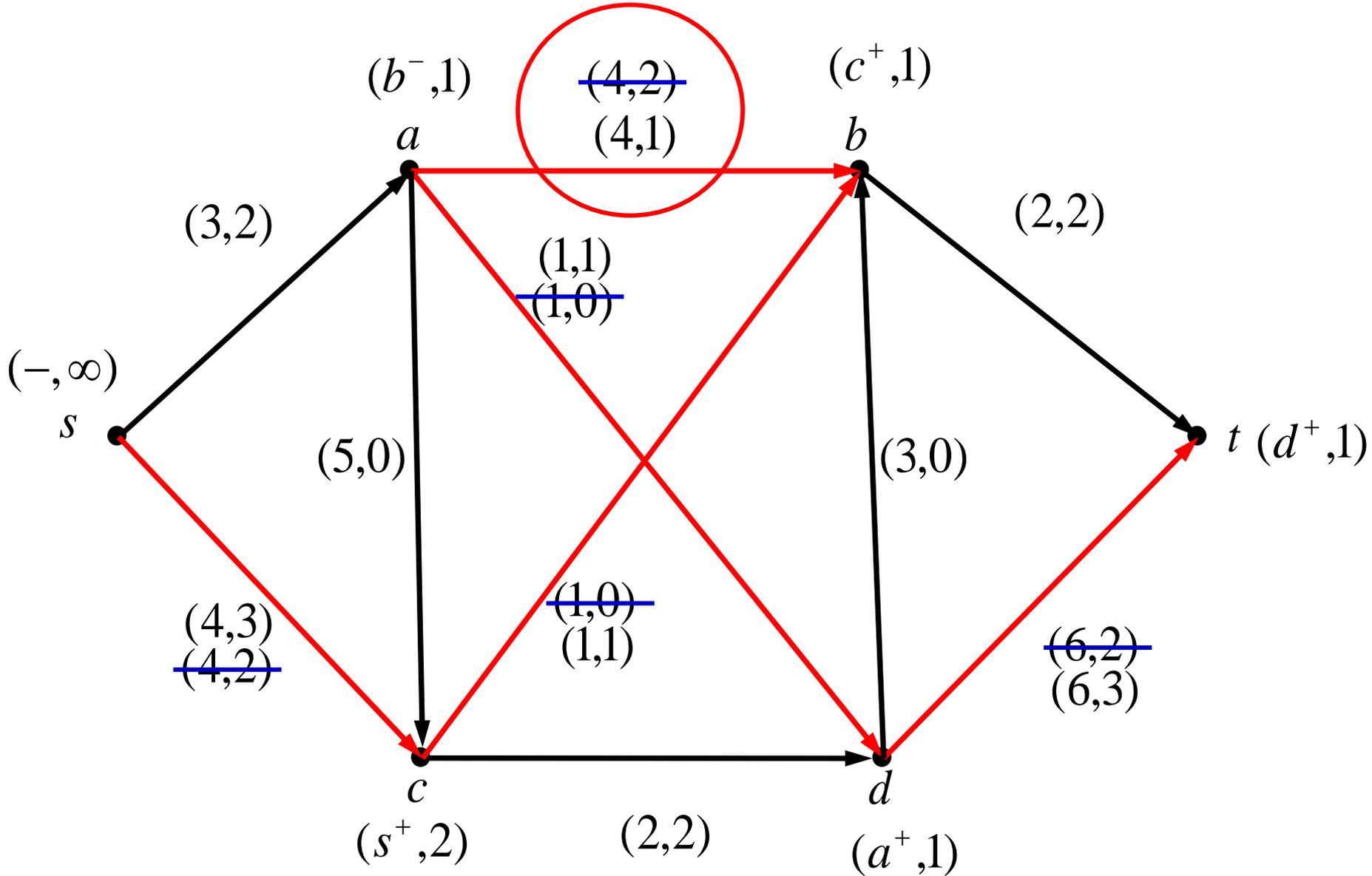
第一轮标号及增流过程结束，此时 $w = 2$

例：



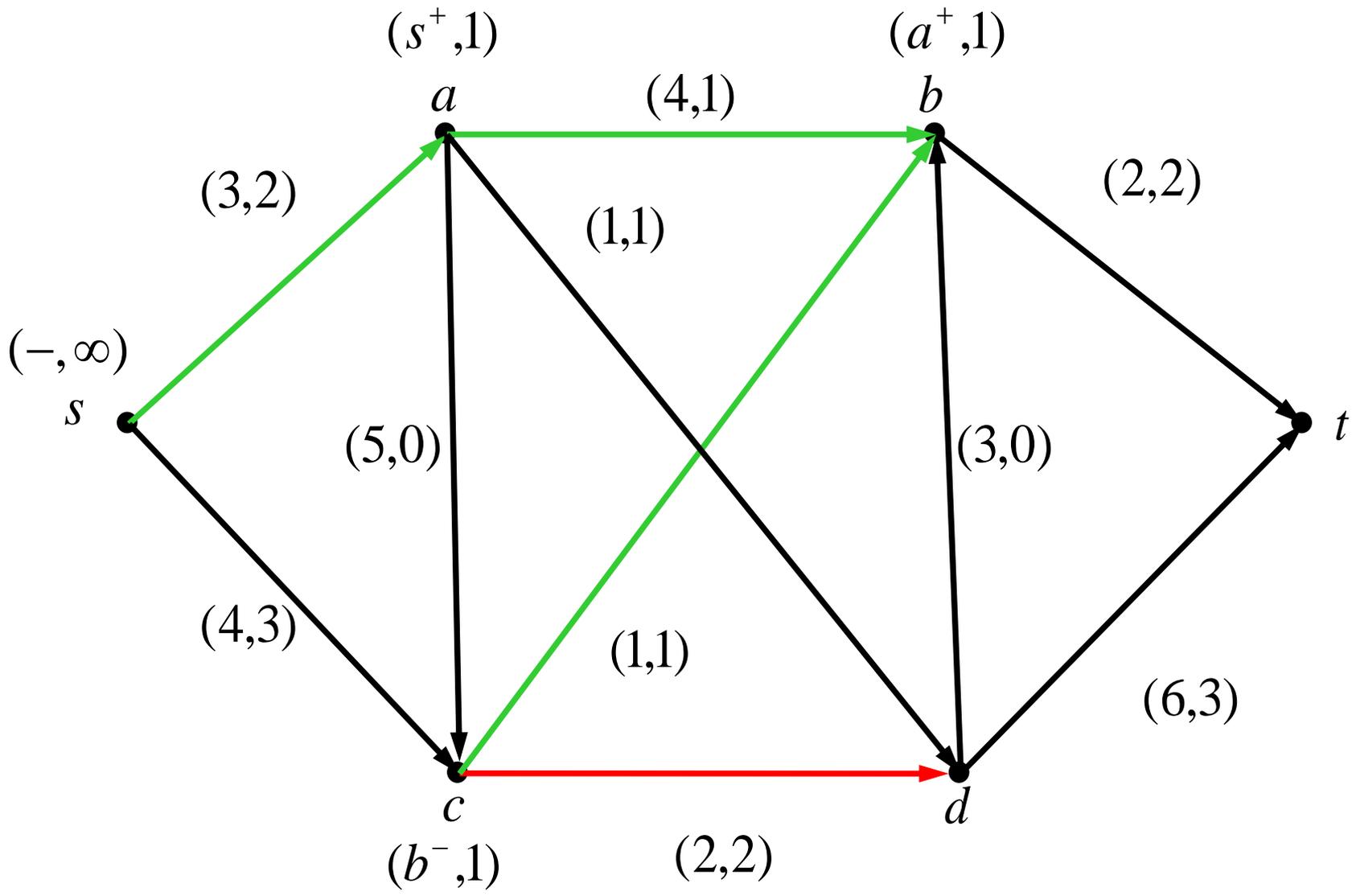
第二轮标号及增流过程结束，此时 $w = 4$

例:

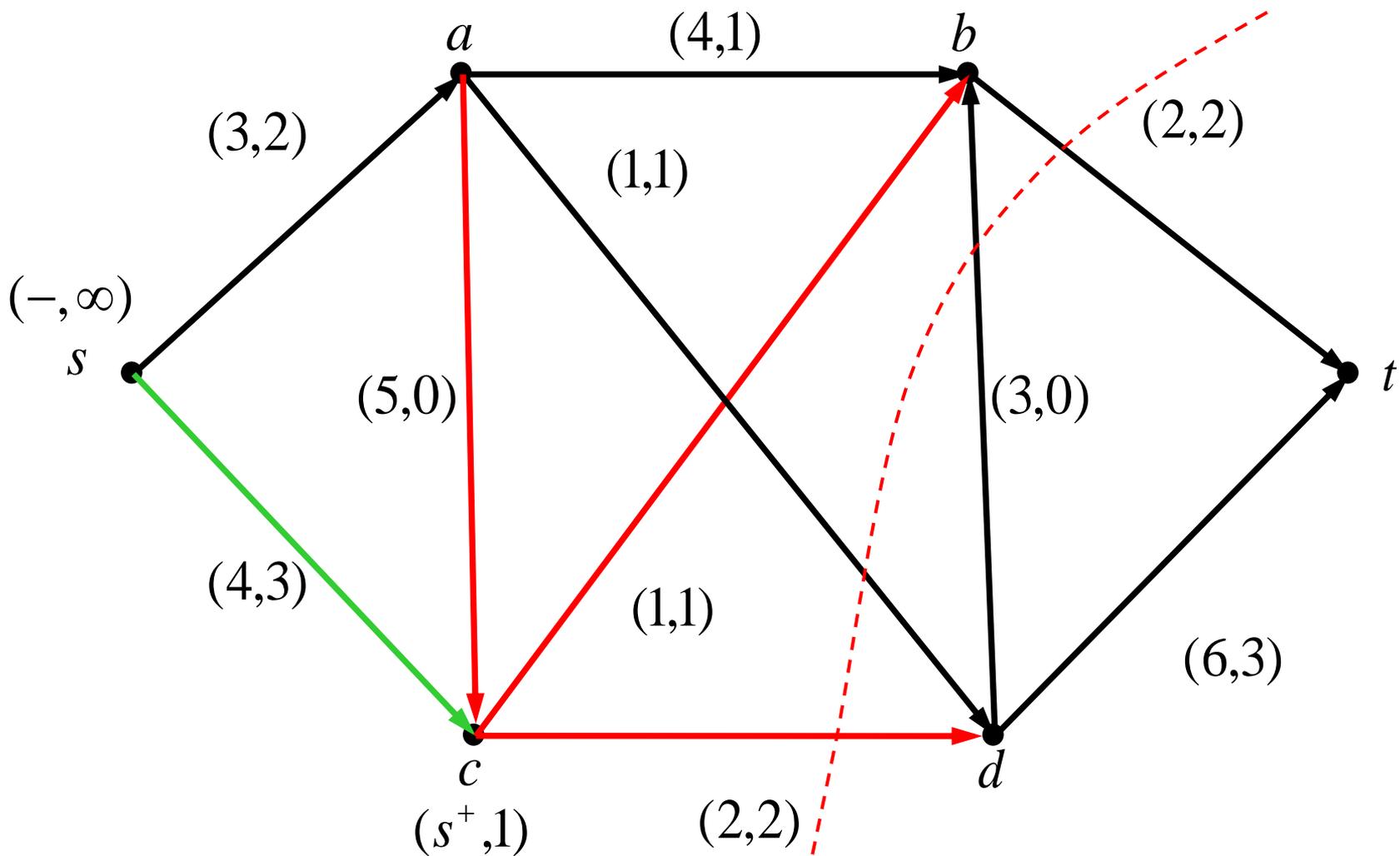


第三轮标号及增流过程结束，此时 $w = 5$

例：



例：



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

算法结束，此时 $\max(w) = 5$

Ford-Fulkerson最大流标号算法描述:

1. 任选一允许流分布 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

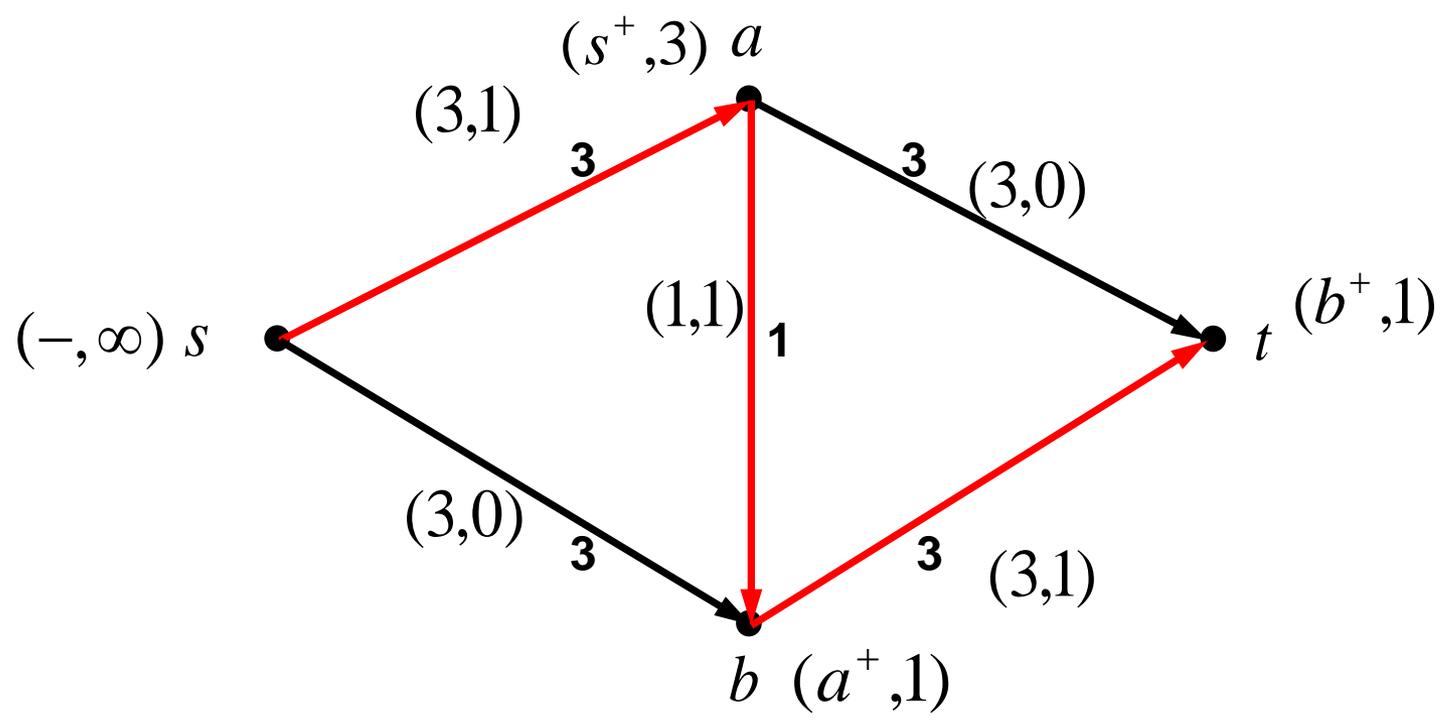
- 修改流量分布

- 最多 m 次流量值修改

- 循环往复, 直至无法找到增流路径。

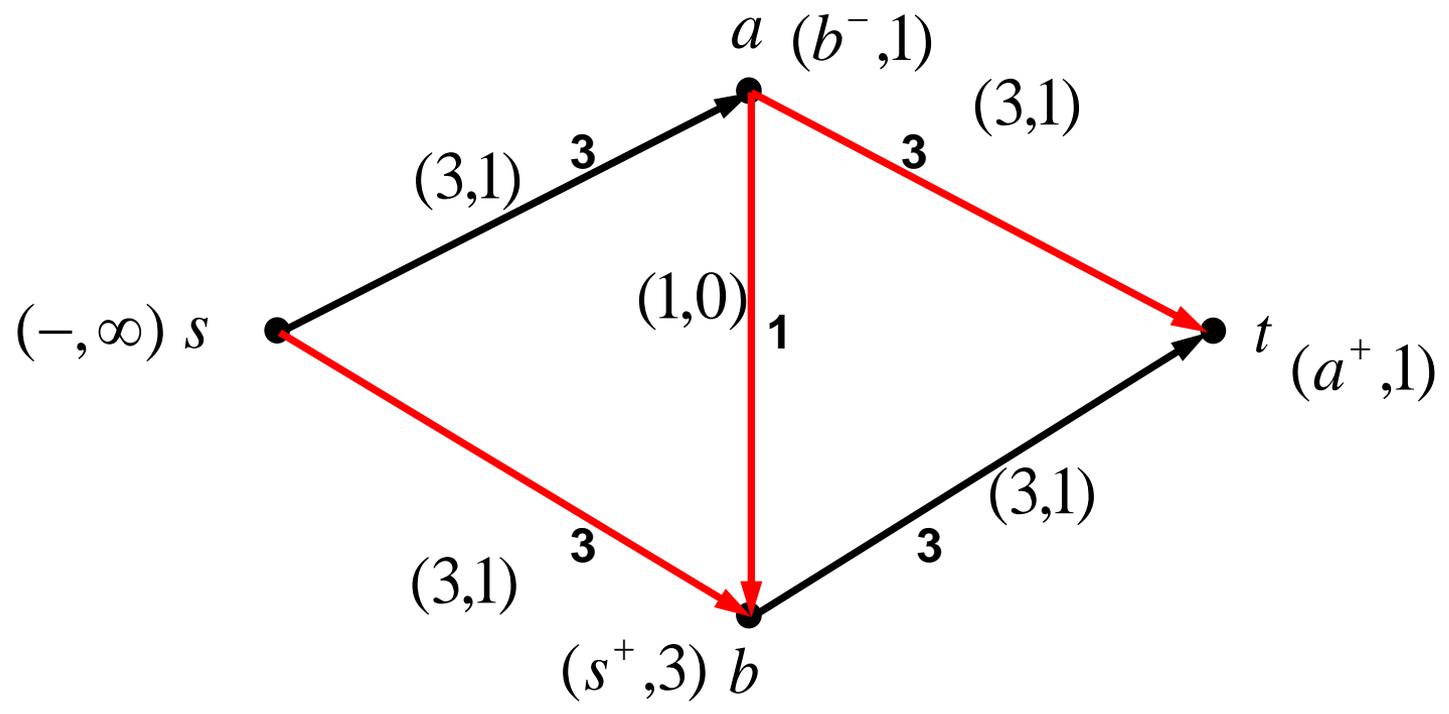
- 循环次数?

例：



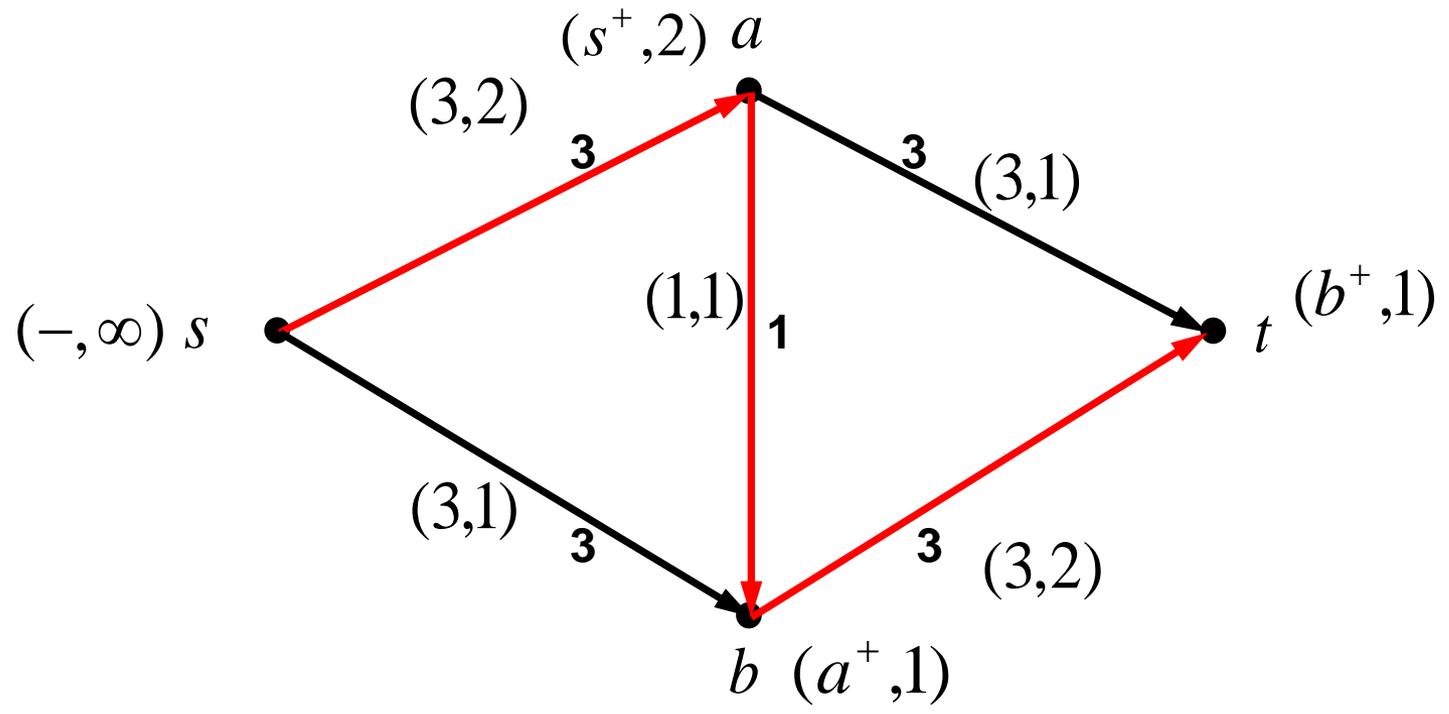
第1次增流！

例：



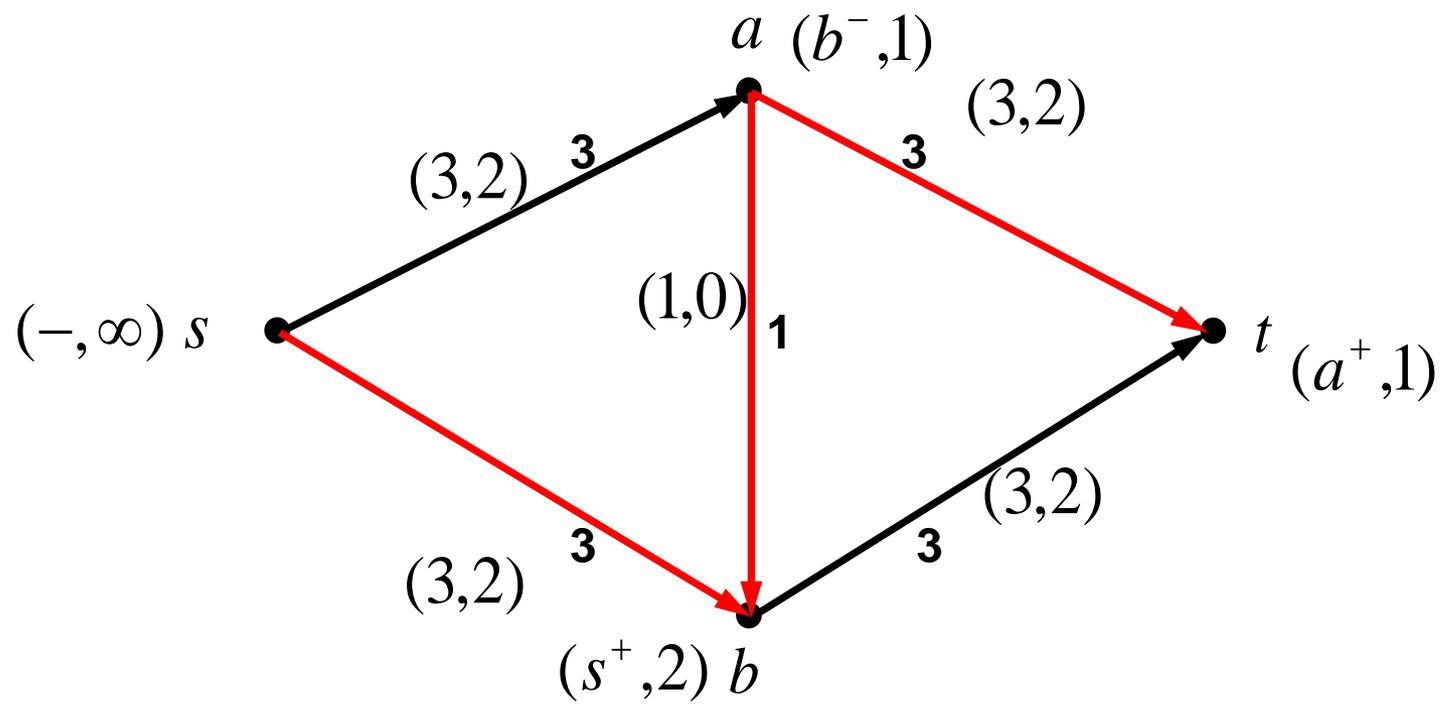
第2次增流！

例：



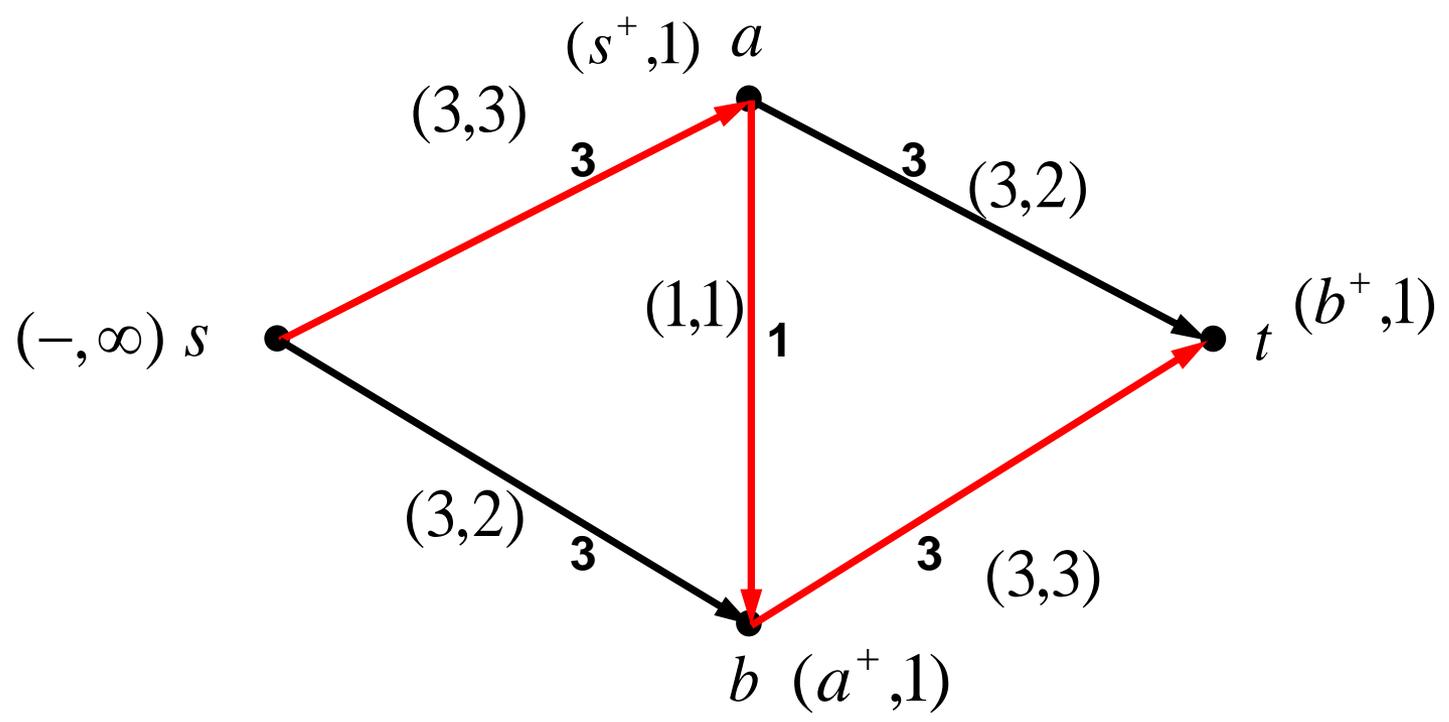
第3次增流！

例：



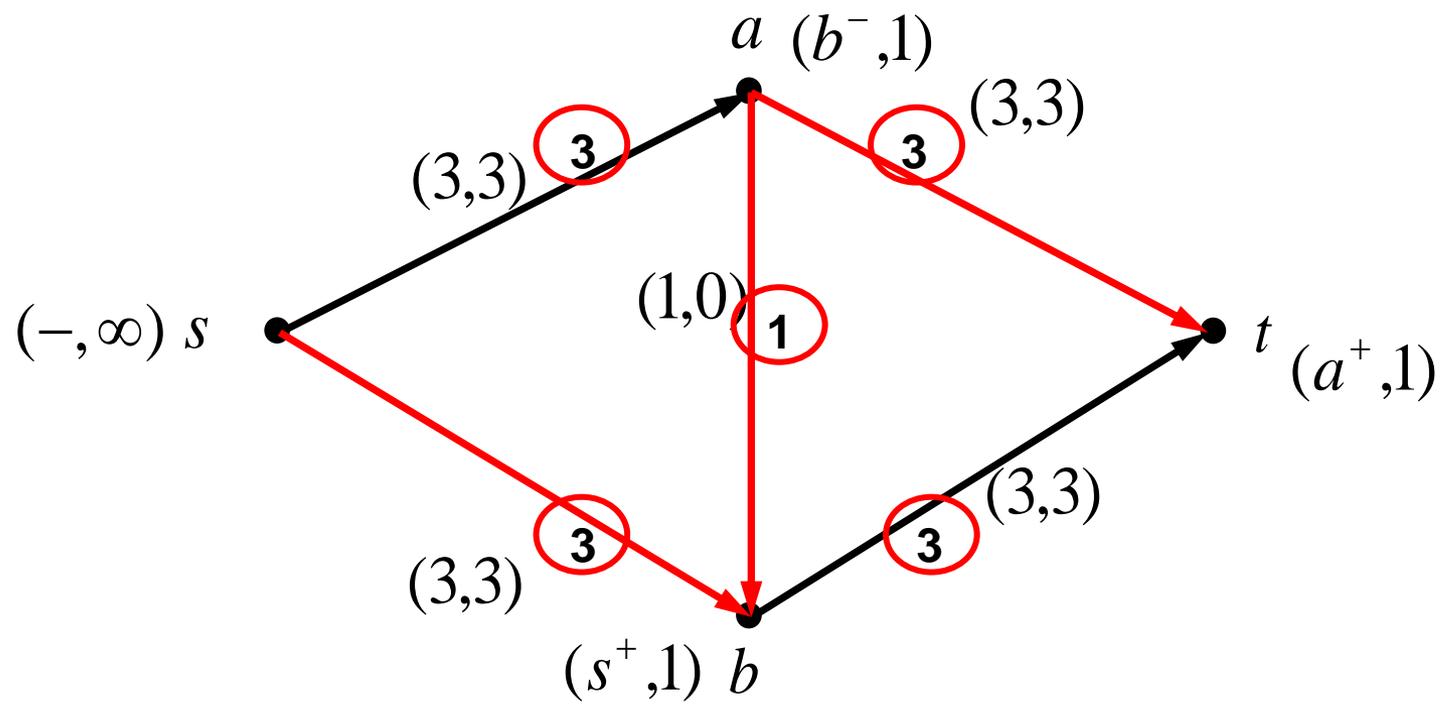
第4次增流！

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. 任选一允许流分布 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 $f(u, v) = f(u, v) + \delta_v$

2. 若 $d_v = u^-$, 则令 $f(v, u) = f(v, u) - \delta_v$

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. 任选一允许流分布 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) ,

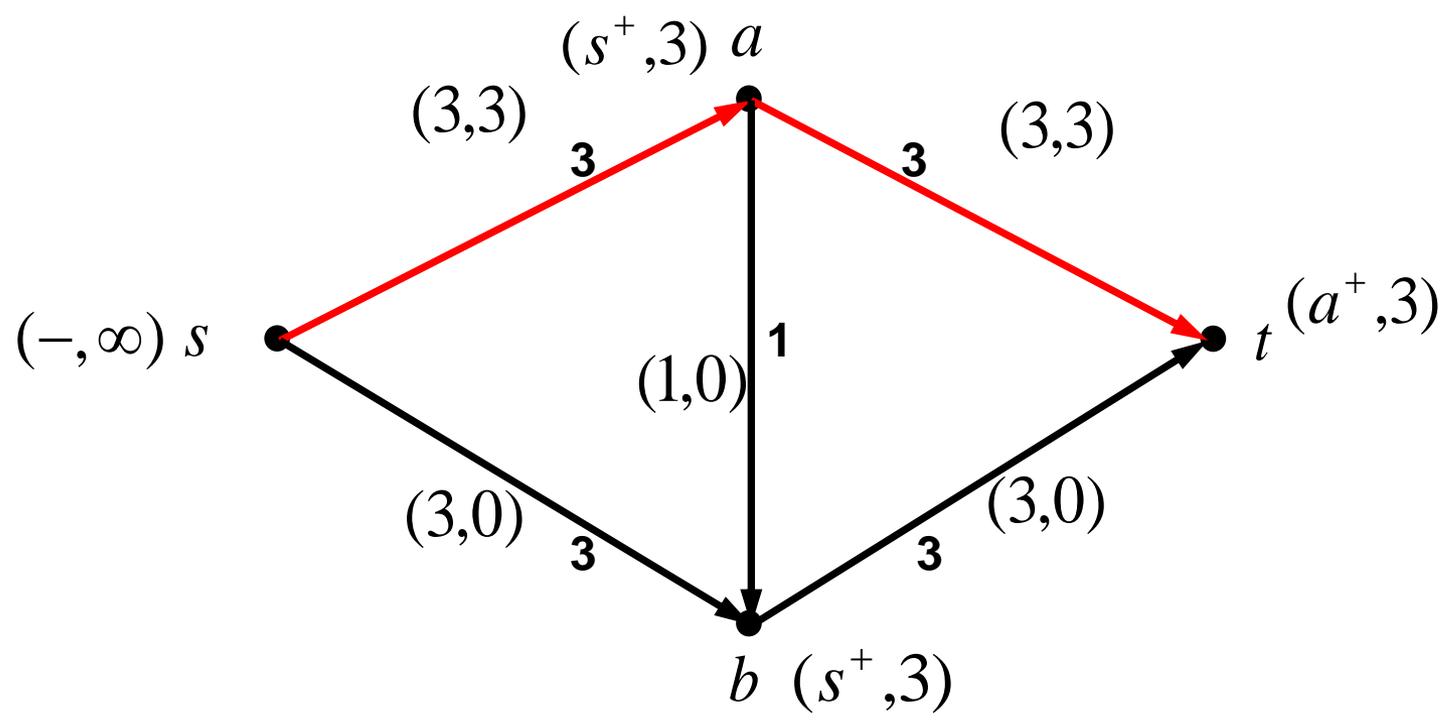
1. 若 $d_v = u^+$, 则令 $f(u, v) = f(u, v) + \delta_t$

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

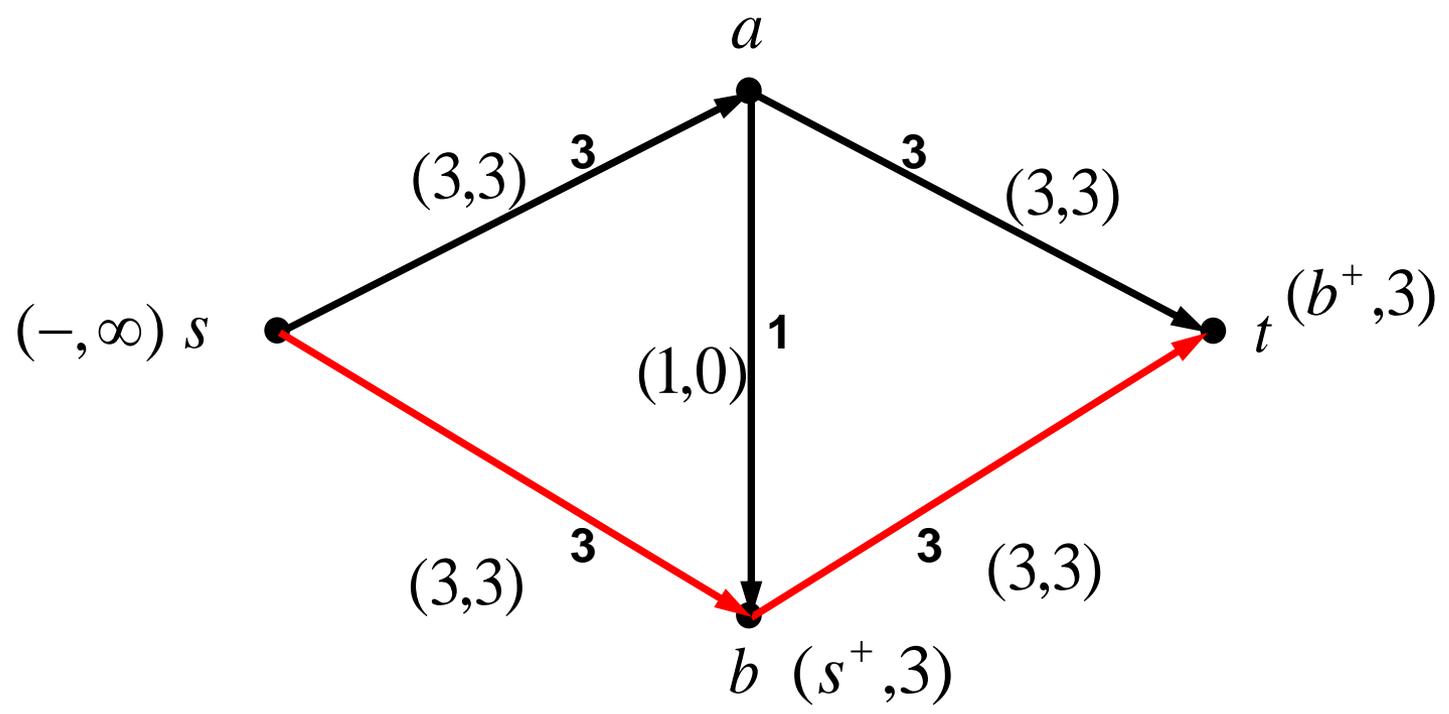
7. 结束

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

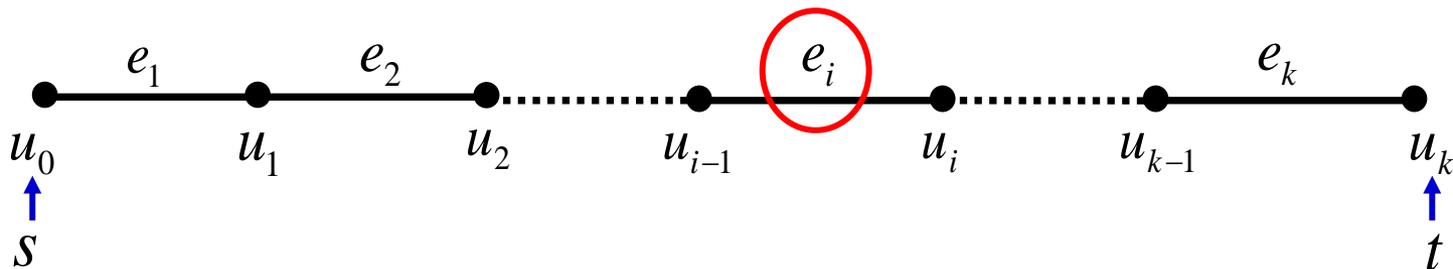
- 循环往复, 直至无法找到增流路径。

- 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 中的允许流分布

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 的允许流 f_1 , f_2 , ...
- 令 $\lambda^i(u, v)$ 表示在允许流 f_i 分布下，从结点 u 到结点 v 的最短非饱和路径长度



最大流的Edmonds-Karp算法

- 引理5.7.1 对每个结点 v 及每个 $k = 0, 1, 2, \dots$

$$\lambda^k(s, v) \leq \lambda^{k+1}(s, v) \quad (1)$$

$$\lambda^k(v, t) \leq \lambda^{k+1}(v, t) \quad (2)$$



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

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



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 ，一个允许流分布的序列为 f_{k_1}, f_{k_2}, \dots ，其中 $k_1 < k_2 < \dots$ ，而且边 e 在 f_{k_i} 中是向前边瓶颈。



最大流的Edmonds-Karp算法

— 由引理5.7.2可以断定, 存在另一个序列 l_1, l_2, \dots , 满足 $k_1 < l_1 < k_2 < l_2 \dots$, 且 e 在 f_{l_i} 中为向后边

— 由引理5.7.3 $\lambda^{k_i}(s, t) + 2 \leq \lambda^{l_i}(s, t)$

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



最大流的Edmonds-Karp算法

- 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
$$j \leq \frac{n+2}{4}$$
- 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!



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



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



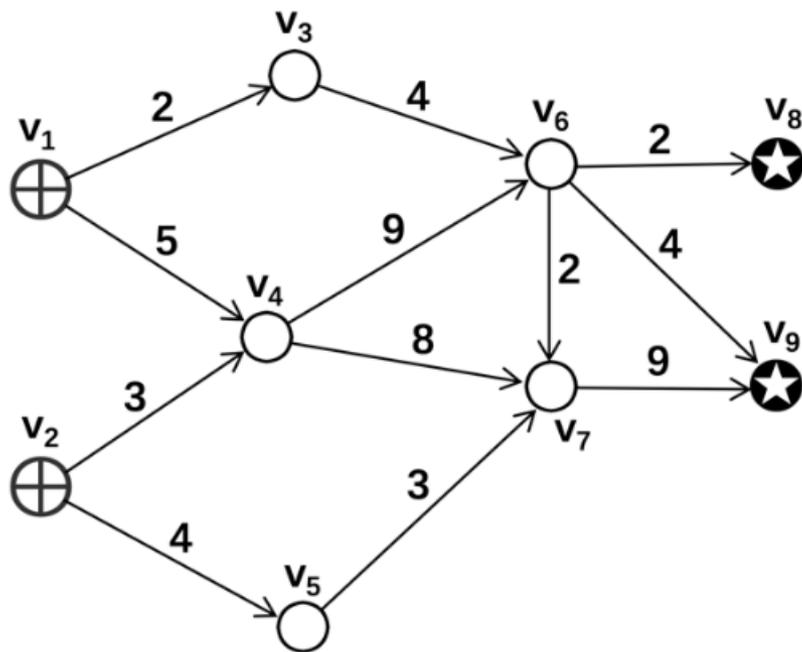
思考

- 最大匹配问题和最大流问题有什么关系？



练习题

某地区的道路分布如图所示，其中节点 $v_1 \dots v_9$ 表示各个区域，边表示单向道路。现在A市的 v_1 和 v_2 地区爆发了一种传染性极强的疫情。为了阻止疫情传播到A市以外的节点 v_8 和 v_9 ，造成更大影响，可以考虑实行交通管制。如下图，如果对道路进行交通管制（代价为对应边的权值），来阻断病毒沿这条道路传播。请设计一种阻断疫情传播的最小代价方案。请给出管制的道路集合以及总代价。





作业

- 课后 1、3
- 选做：4，8
- 下次课（5月12日）：习题课