离散数学II
—图论第十一讲

周旻
清华大学软件学院
软件工程与系统研究所

2024年6月14日
Friday

第六章 网络流

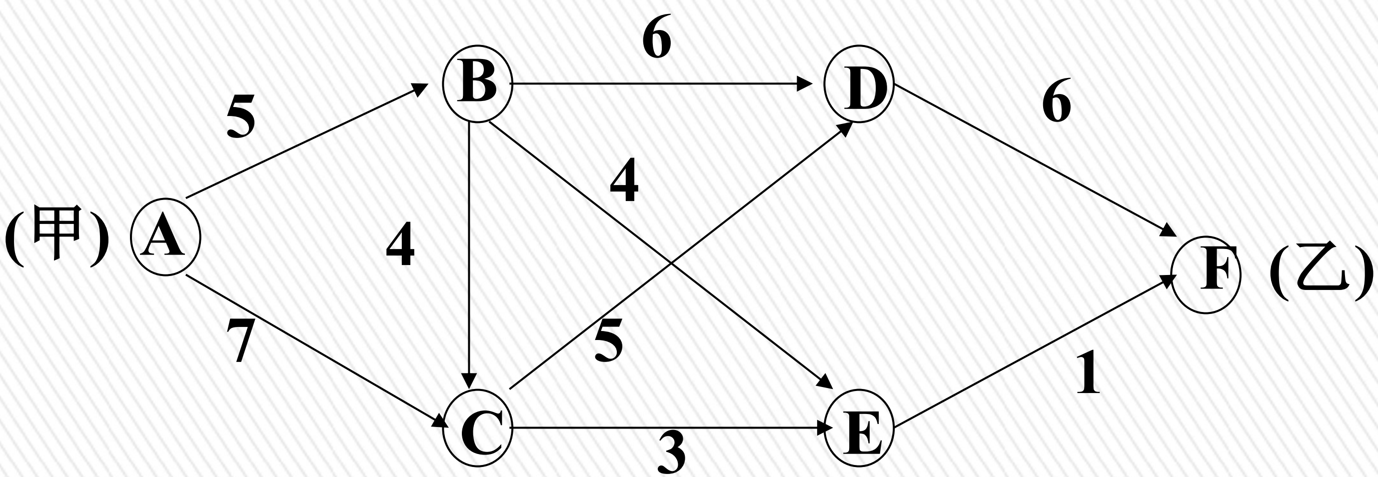
- 网络流图
- Ford-Fulkerson最大流标号算法
- 最大流的Edmonds-Karp算法
- 最小费用流

网络流图

网络流问题

(1) 应用背景

从甲地到乙地的公路网纵横交错，每天每条路上的通车量有上限。从甲地到乙地的每天最多能通车多少辆?



考虑每条路上的通行成本，如何确定某个车队的具体行车路线, 使总成本最小？

网络流问题

(1) 应用背景

- 网络流问题是一类应用极为广泛的问题，例如在交通运输网络中有人流、车流、货物流，供水网络中有水流，金融系统中有现金流，通讯系统中有信息流，等等。

50年代福特（Ford）、富克逊（Fulkerson）建立的“网络流理论”，是网络应用的重要组成部分。

是图论与组合最优化中内容丰富、应用广泛的一个问题。

例如：

产销网络流图可以看作某种产品从产地*s*通过不同的道路可达销地，边的容量表示沿这条边最多通过的量。

网络流问题

(2) 基本概念

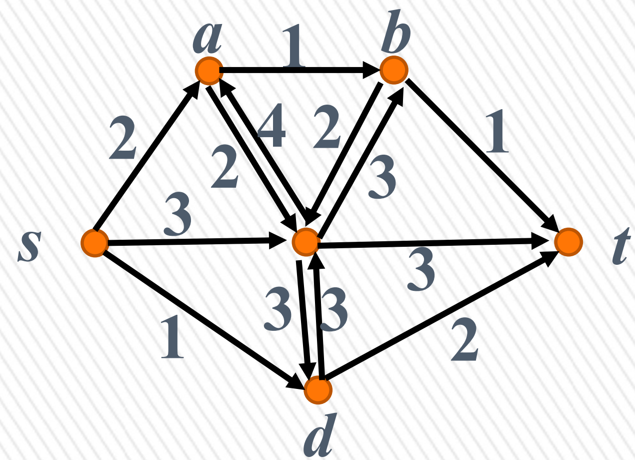
■ **定义6.1.1**

一个网络N是一个无自环的有向连通图，满足

① 只有一个入度为0的点s，称为**源**。

② 只有一个出度为0的点t，称为**汇**。

③ 每条边（或弧）(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j没有边，则$c_{ij} = 0$。

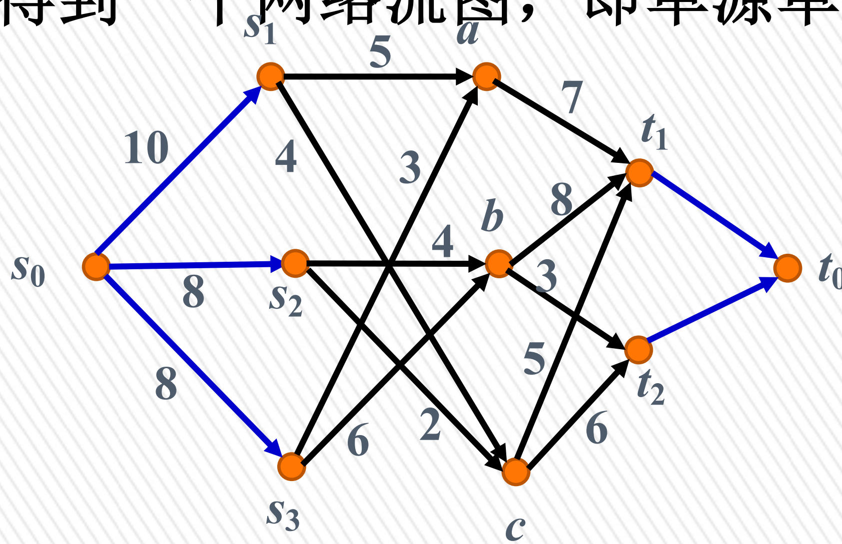
(既非源，又非汇的顶点。称为**中间点**，一般地$c_{ij} \neq c_{ji}$)

网络流问题

(2) 基本概念

源、汇不唯一世：可以增加一个**超发点**s_0，一个**超收点**t_0，增加若干边(s_0, s_i)和(t_j, t_0),，其中s_i, t_j分别是每个产地和销地

同时边(s_0, s_i)的容量是s_i的生产能力，(t_j, t_0)容量是t_j的销售能力，这样就得到一个网络流图，即单源单汇的图。



网络流问题

(2) 基本概念

流的定义：在网络N中，如果每条边e_{ij}都给定一个非负实数$f(e_{ij})$，满足

① $0 \le f(e_{ij}) \le c_{ij}, e_{ij} \in N$

② $\Sigma_j f(e_{ij}) = \Sigma_k f(e_{ki}), i \ne s, t$

则称f为该网络的**流**，又称为**可行流**

① 称为**容量约束**：一条有向边的流量不能够超过这条有向边的容量。

② 称为**守恒条件**：对于任何中间点v，物资输入v的流量等于输出v的流量。

注：可行流总是存在的，例如所有$f_{ij} = 0$ (称为零流)

网络流问题

(2) 基本概念

在网络N的一个容许流分布f里，满足$f_{ij} = c_{ij}$的边称为饱和边，否则为非饱和边

对收、发点u_t, u_s有$\Sigma_i f_{si} = \Sigma_j f_{jt} = W$

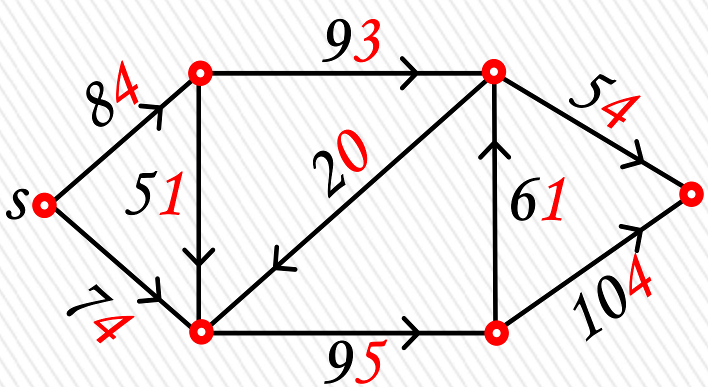
（即从u_s点发出的物资总量等于u_t点输入量）

W为网络流的总流量

如果一个容许流分布使得网络的流量w_0为最大，

即$w_0 = \max \Sigma_j f_{sj}$,

就说w_0是网络的最大流



网络流问题

(3) 最大流问题

等价于线性规划问题：

求同时满足以下条件的max w

$$\Sigma f_{ij} - \Sigma f_{ji} = w \quad i = s$$
$$\Sigma f_{ij} - \Sigma f_{ji} = -w \quad i = t$$
$$\Sigma f_{ij} - \Sigma f_{ji} = 0 \quad i \neq s, t$$
$$0 \leq f_{ij} \leq C_{ij} \qquad \forall (i, j) \in V$$

因此最大流问题可以通过单纯形法或其他线性规划的方法解决，但我们可以利用图论得到更简单的方法。

网络流问题

(3) 最大流问题

定义6.1.2 设S是网络流图$N=(V,E)$中的一个结点集，满足

(1) $s\in S$

(2) $t\in\bar{S}, \bar{S}=V-S$

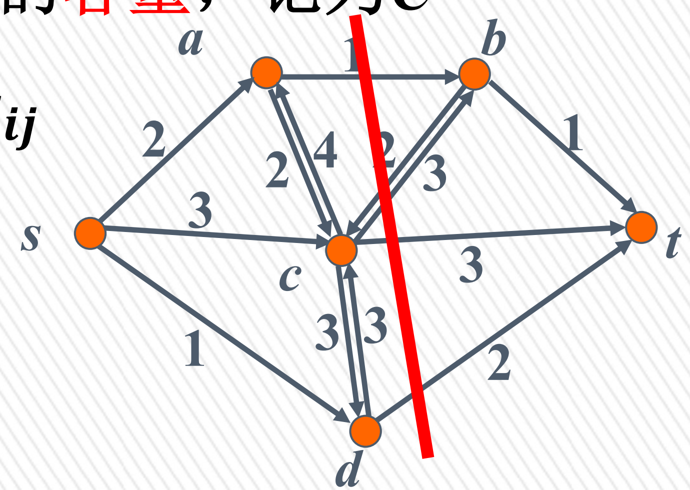
针对单一发点和收点的网络

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$(S,\bar{S})$

把割切的全部边集去掉后，由s到t无任何有向路。

$(S,\bar{S})$中的各边的容量之和称为该割切的容量，记为C

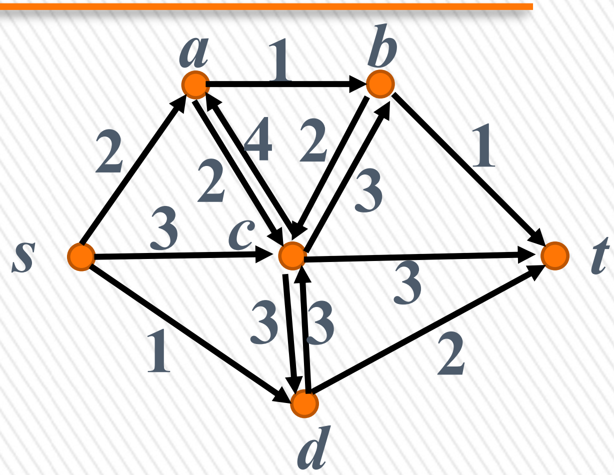
$$C(S,\bar{S})=\sum_{(i,j)\in(S,\bar{S})} c_{ij}$$



注意：计算时应是正向边的容量之和

网络流问题

(3) 最大流问题



例6.1.1

图中令$S = \{s\}$

则$(S, \bar{S}) = \{(s, a), (s, c), (s, d)\}$, $C(S, \bar{S}) = 6$

令$S = \{s, a, c\}$

则$(S, \bar{S}) = \{(a, b), (c, b), (c, t), (c, d), (s, d)\}$, $C(S, \bar{S}) = 11$

定理6.1.1

网络的最大流量小于等于最小的切割容量，即$\max w \le \min C$

网络流问题

(4) 增流路径

增流路径

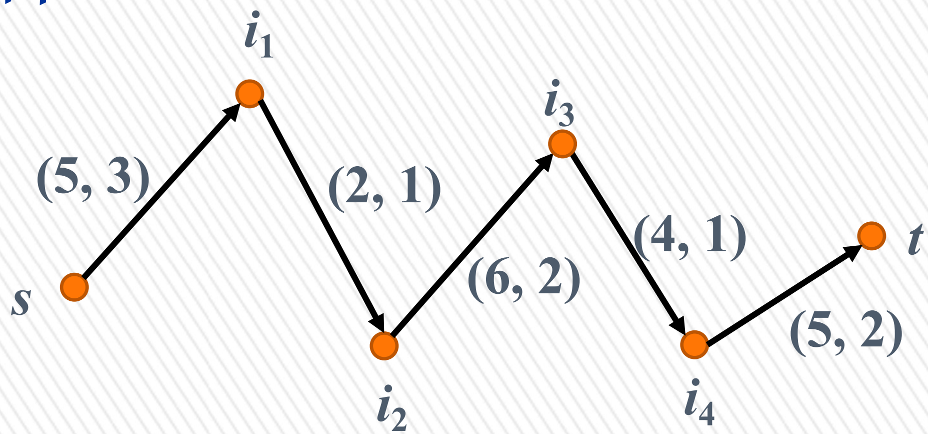
- 如果网络的容许流并不是最大流，就一定存在着从s到t的增流路径
- 令$s, i_1, i_2, \ldots, i_k, t$是一条从$s$到$t$的初级路径$P_{st}$

a. 前向边的情况： 每条边的方向都是从i_j到i_{j+1}

- 如果这条路径上每条边e_{ij}都有$f_{ij} < c_{ij}$，那么令$\delta = \min_{e_{ij} \in P_{st}} (c_{ij} - f_{ij})$，这时令$P_{st}$每条边的流都增加$\delta$，结果仍然是网络的容许流分布，但流量比先前增加了$\delta$

网络流问题

(4) 增流路径



该道路上$\delta = 1$，即沿这条s-t道路网络的流量最多可增加1

有后向边怎么办？

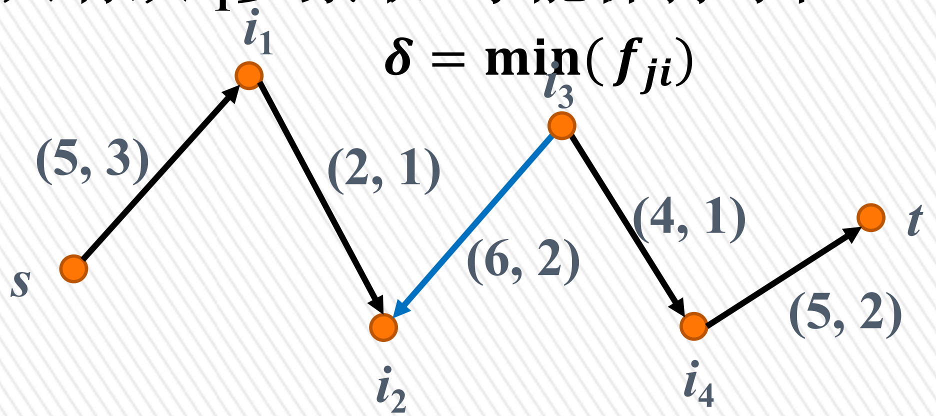
网络流问题

(4) 增流路径

b. 后向边的情况： 下图的$\delta = 1$，在这条道路上的增流过程是

- 汇点的流入量增加1是从i_4获得
- i_4要保持流的守恒，应使f_{34}增加1
- 而i_3的守恒是由i_3少供应1个单位流给i_2而得到保证
- 因此增流路径中的后向边e_{ji}一定要$f_{ji} > 0$，这时i_2由于i_3少供应1，只有从i_1多索取1才能保持守恒

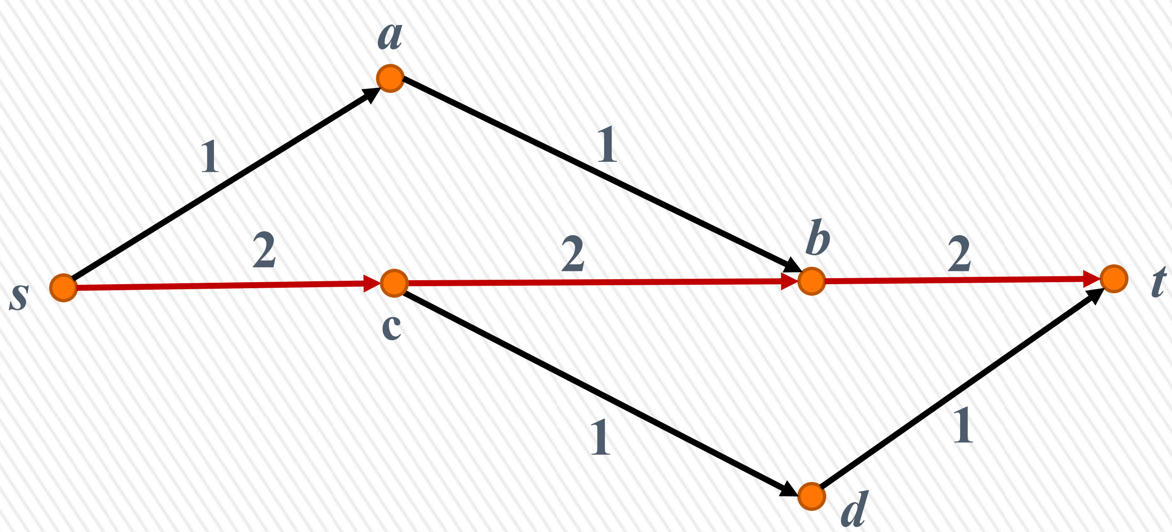
$$\delta = \min(f_{ji})$$



网络流问题

(4) 增流路径

例6.1.2 图中，如果最初流量$w = 0$,
第一条增流路径可以是(s, c, b, t)
它全部由前向边组成，$\delta = 2$，因此可增流2
这时边(s, c), (c, b), (b, t)的流都是2，其余边均为0,
这是一个容许流分布

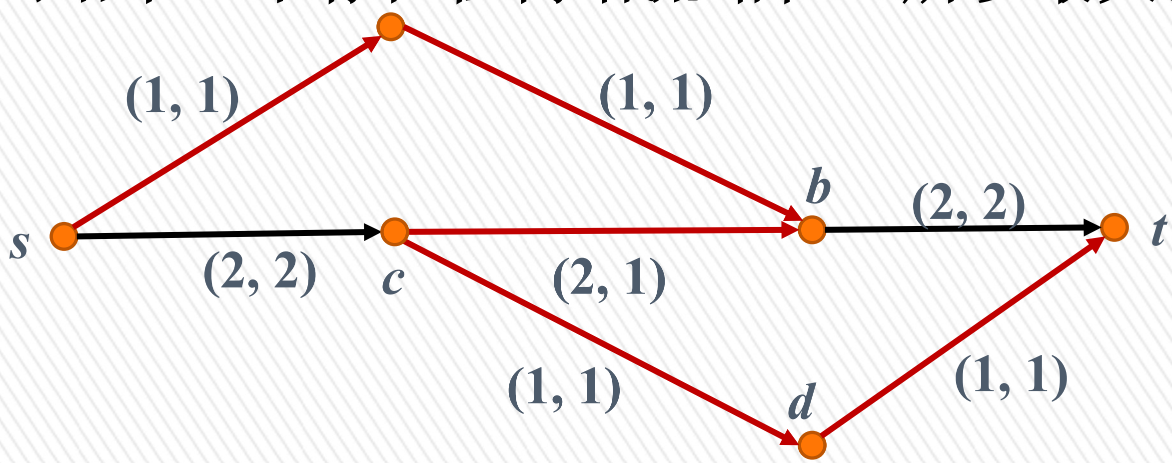


网络流问题

(4) 增流路径

例6.1.2（续）

- 此时还存在另一条增流路$(s, a, b, c, d, \mathrm{t})$
- 其中(c, b)是后向边，$f_{cb} = 2$，其余都是前向边，满足$f_{ij} > c_{ij}$，这条路上$\delta = 1$
- 因此增流之后得到下图，其中边(c, b)的流为1，这仍然是一个允许流分布
- 此时网络中已不存在任何增流路径。所以最大流量是$w_0 = 3$



网络流问题

(4) 增流路径

- 对于前向边，如果$f_{ij} = c_{ij}$，或者对于后退边$f_{ji} = 0$，则为饱和边
- 对于一条路径P_{st}上的所有边，如果前向边都有$f_{ij} < c_{ij}$，后向边都有$f_{ji} > 0$，则称这条道路为可增流路径，令

$$\delta_{ij} = \begin{cases} c_{ij} - f_{ij}, & \text{当}(i,j)\text{为前向边} \\ f_{ij}, & \text{当}(i,j)\text{为后向边} \end{cases}$$

$$\delta = \min\{\delta_{ij}\}$$

- 只要可增流路径存在,便可使网络流量得到相应增加

网络流问题

(5) 最大流—最小割定理

定理6.1.2（最大流—最小割定理）

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f是一个最大流，流量为w，用下面的方法定义点集

(1) 令$s \in S$;

(2) 对N中的所有点,

若$x \in S$，(x, y)是前向边且$f_{xy} < c_{xy}$，则令$y \in S$;

若$x \in S$，(y, x)是后向边且$f_{yx} > 0$，则令$y \in S$;

则必有$t \notin S$，否则存在s到t的一条增流路径，与f是最大流矛盾。因此$t \in \bar{S}$

网络流问题

(5) 最大流一最小割定理

证明(续)：根据前面S的生成定义，任意满足$x \in S, y \in \bar{S}$的边

若(x, y)为前向边，只能是$f_{xy} = c_{xy}$;

若(y, x)为后向边，只能是$f_{yx} = 0$;

代入w的计算公式得：

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

由定理6.1.1，$\max w \leq \min C\,(S, \bar{S})$

故 $\max w = \min C\,(S, \bar{S})$

网络流问题

(5) 最大流—最小割定理

最小割的物理意义

网络从发点到收点的各通路中，由容量决定其通过能力，最小割则是此路中的咽喉部分，或者叫瓶口，其容积最小，它决定了整个网络的最大通过能力。要提高整个网络的运输能力，必须首先改造这个咽喉部份的通过能力。

Ford-Fulkerson
最大流标号算法

Ford-Fulkerson最大流标号算法

增流路径算法

从一个可行流开始，寻求关于这个可行流的可增流路径，若存在，则可以经过调整，得到一个新的可行流，其流量比原来的可行流要大，重复这个过程，直到不存在关于该流的可增增流路径时就得到了最大流。

如何寻找可增流路径？

Ford-Fulkerson最大流标号算法

Ford-Fulkerson算法（1957）

以网络最大流等于最小割切容量定理为基础，包含两个过程:

(1) 标号过程

- 检查网络中是否存在关于f的增流路径
- 如果不存在，则由定理，此时的f是最大流分布，其流量为最大流
- 否则在标号过程中最后能标到结点t，即存在s到t的增流路径，转过程（2）

(2) 增流过程

- 确定一条从s到t的增流路径并修正这条路上的流，得到新的容许流分布f'，再转（1）

Ford-Fulkerson最大流标号算法

Ford-Fulkerson算法

Step0. 令f是任意一个流(例如$f = 0$)。给s一个永久标号(-, ∞)。

Step1. 标号过程：若v_i已标号，如果可找到一个未标号结点v_j，则继续执行标记结点v_j，否则无法再找到可增流路径，结束

a. 若存在$(v_i, v_j) = a$且$f(a) < c(a)$，则v_j标号$(v_i^+,\ \delta_{v_j})$，$\delta_{v_j} = \min\{\delta_{v_i},\ c(a) - f(a)\}$

b. 若存在边$(v_j, v_i) = a$且$f(a) > 0$，则给v_j标号$(v_i^-,\ \delta_{v_j})$，$\delta_{v_j} = \min\{f(a),\ \delta_{v_i}\}$

Step2. 若t已被标号，则找到了一条增流路径，转Step3，否则迭代执行step1和2。

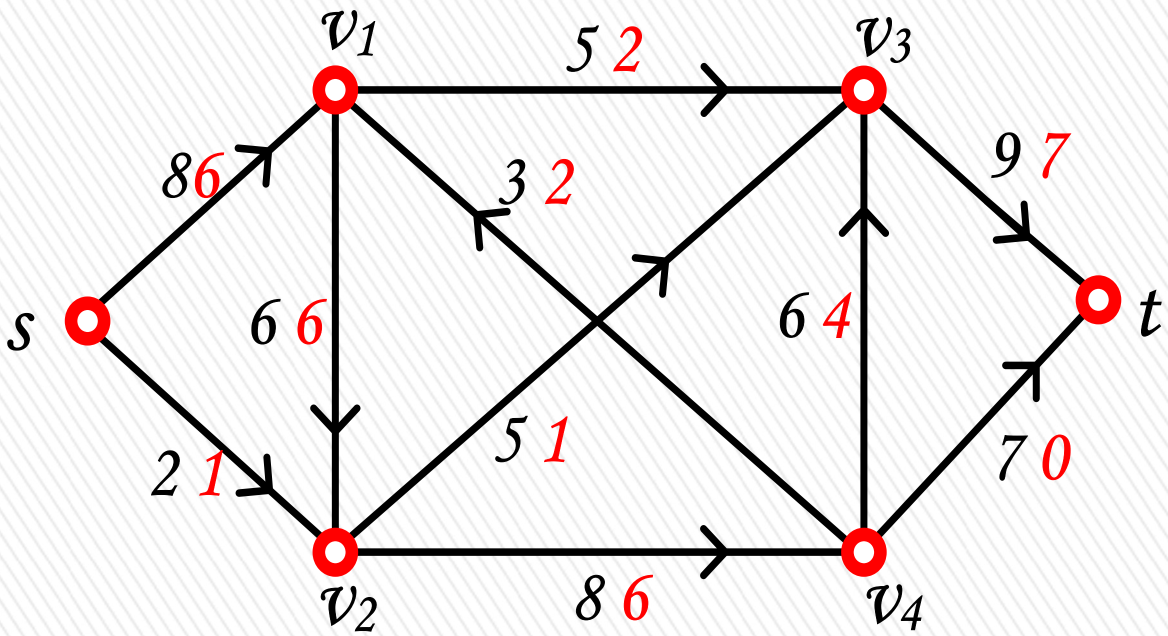
Step3. 由点t开始，使用标号的第一个元素构造一条f增流路p，修改f得到新的流f'，以f'代替f，掉除s外的所有点的f标号。返回Step1。

这里

$$f' = \begin{cases} f(a) + \delta_t & \text{若是前向边} \\ f(a) - \delta_t & \text{若是后向边} \\ f(a) & \text{其它.} \end{cases}$$

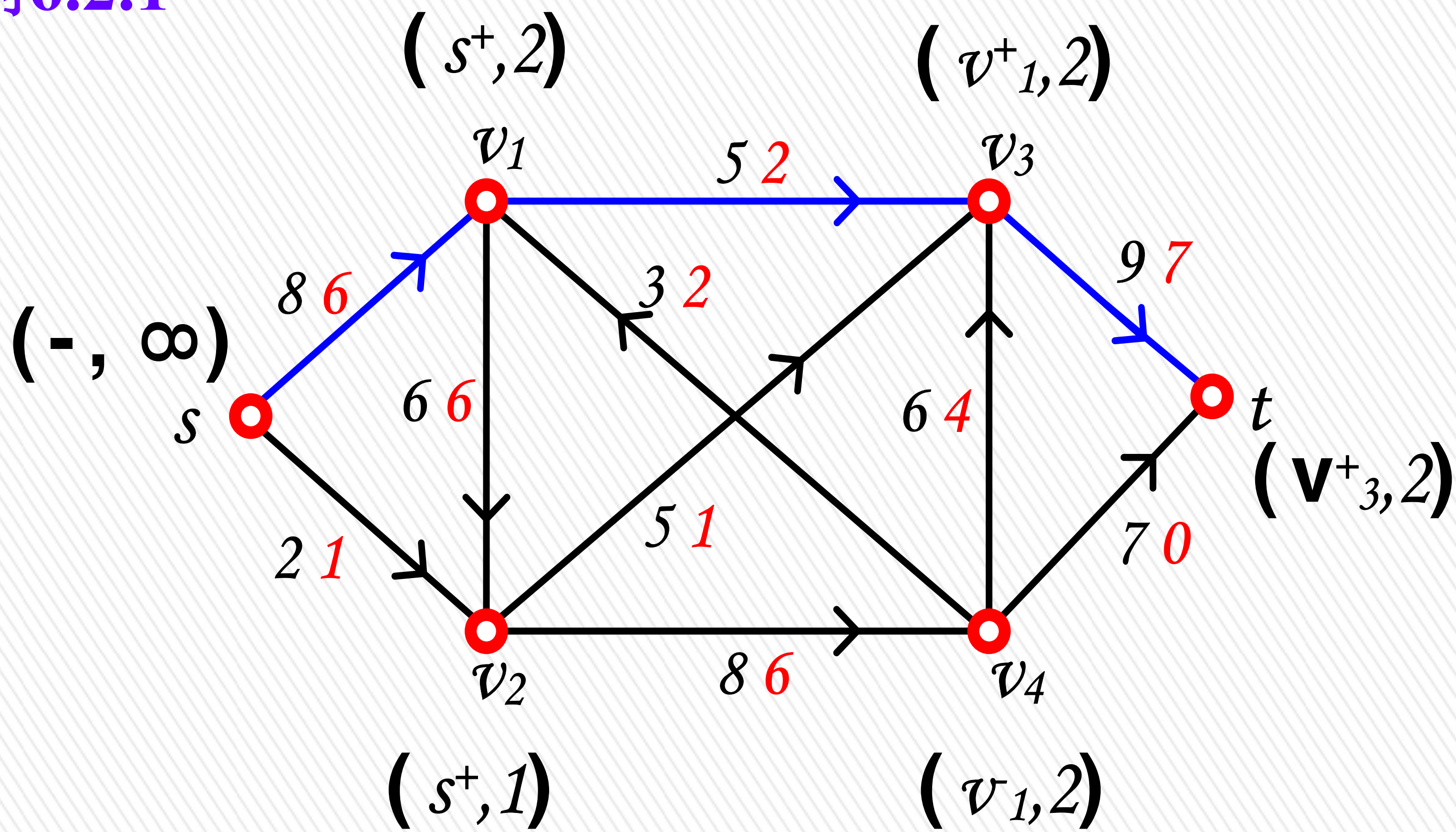
Ford-Fulkerson最大流标号算法

例6.2.1



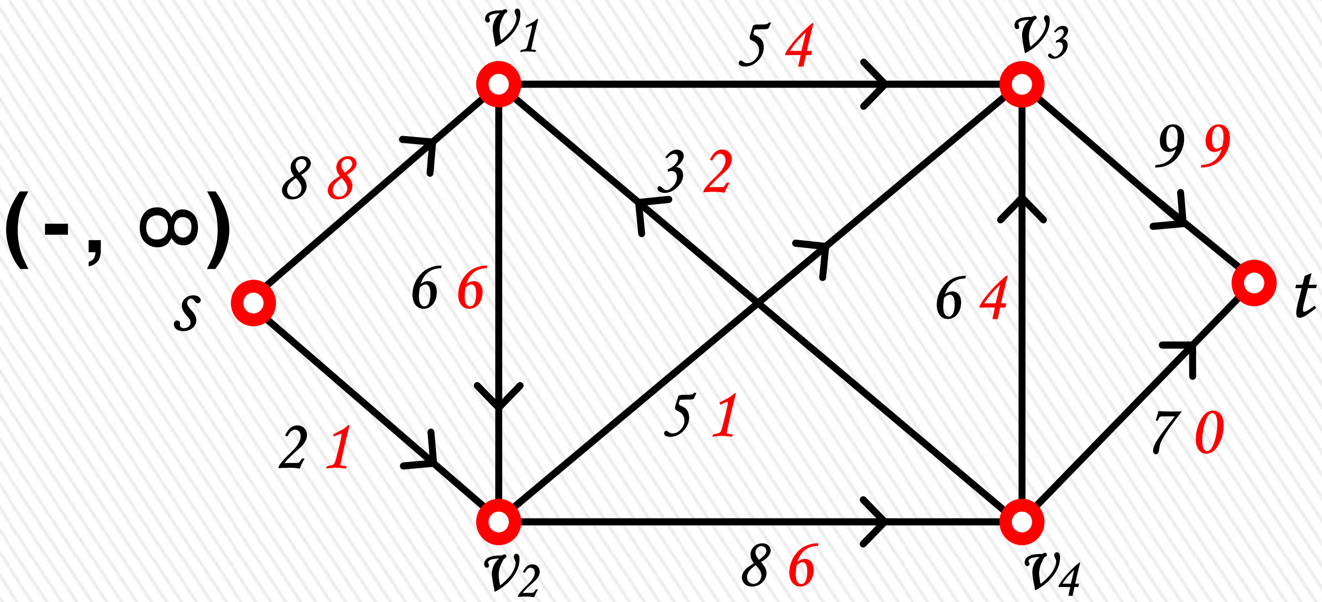
Ford-Fulkerson最大流标号算法

例6.2.1



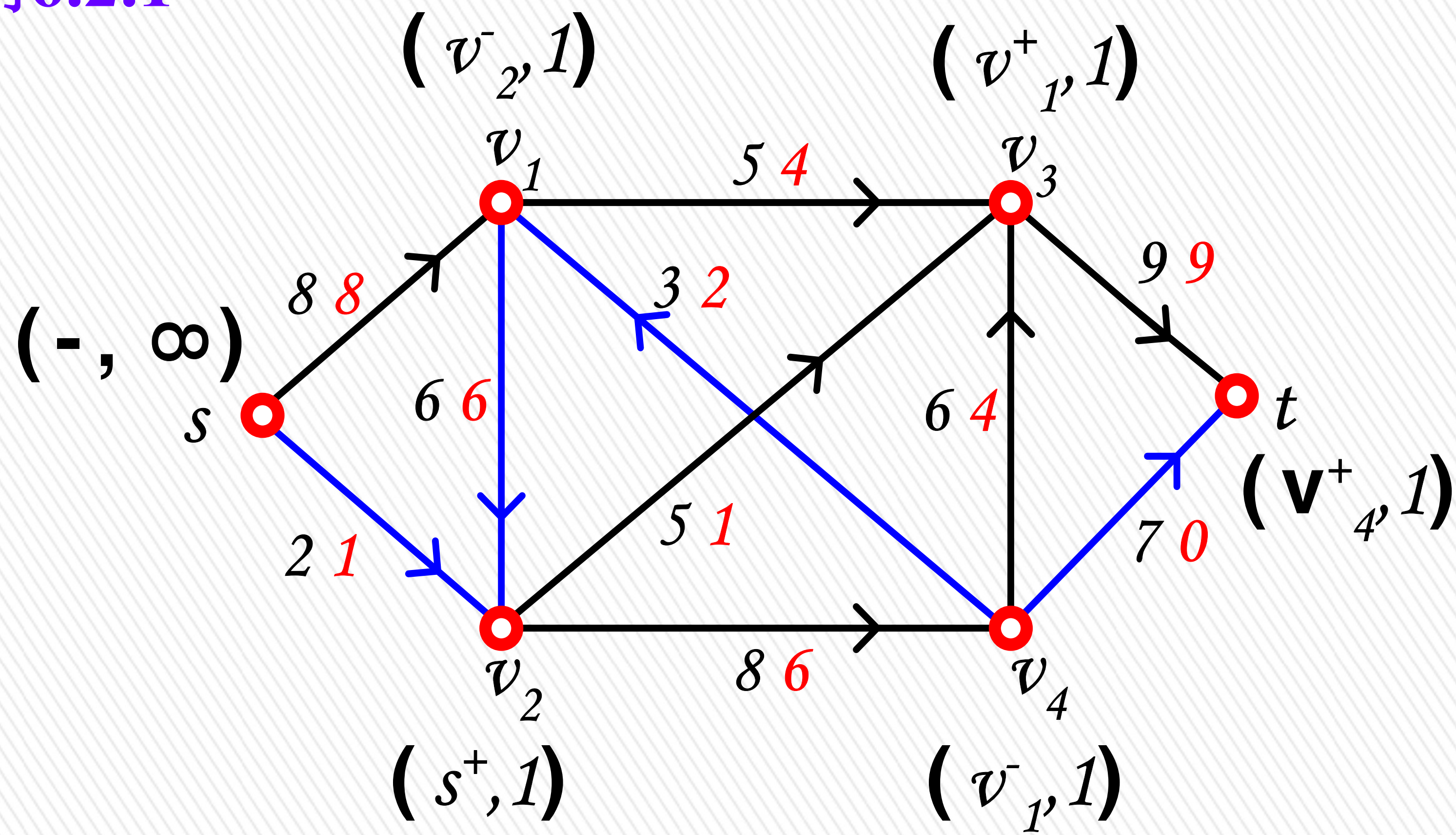
Ford-Fulkerson最大流标号算法

例6.2.1



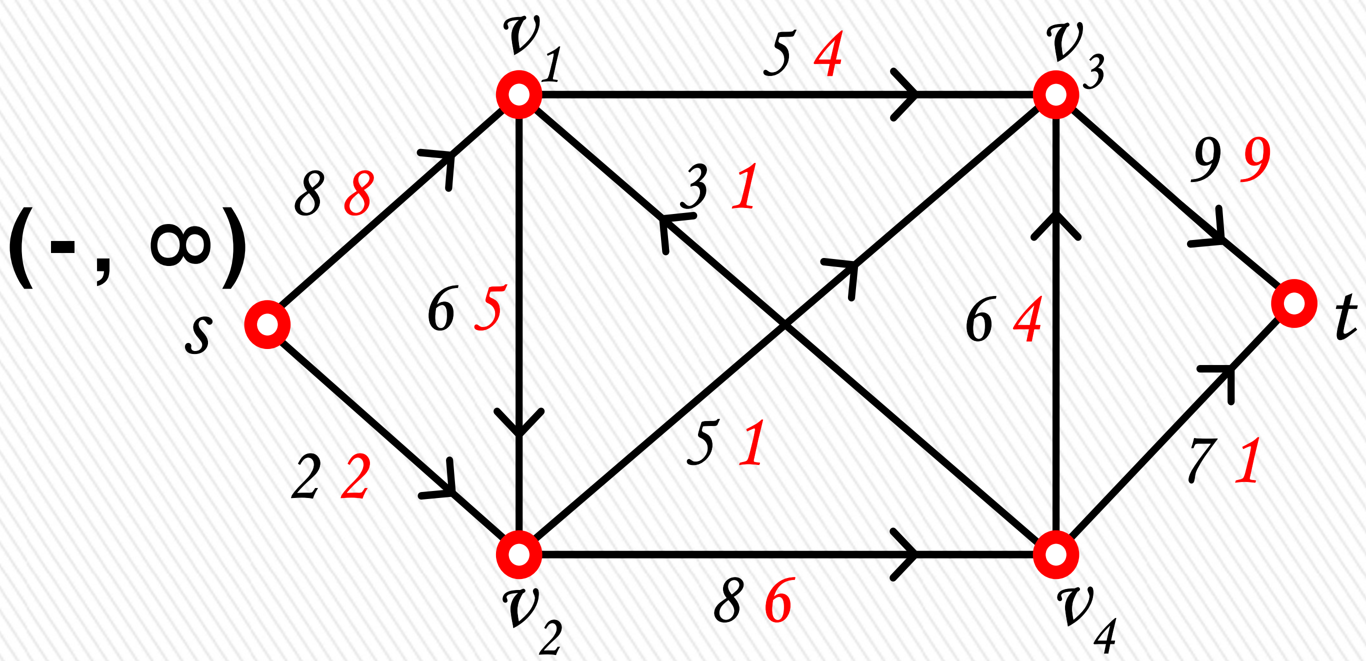
Ford-Fulkerson最大流标号算法

例6.2.1



Ford-Fulkerson最大流标号算法

例6.2.1



Ford-Fulkerson最大流标号算法

存在问题

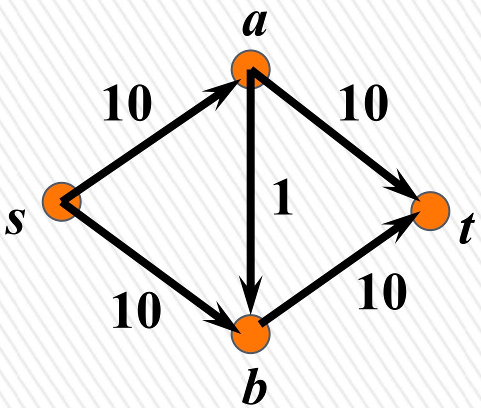
在算法中，对结点的标号顺序是任意的

即可以任选一条s到t的增流路径

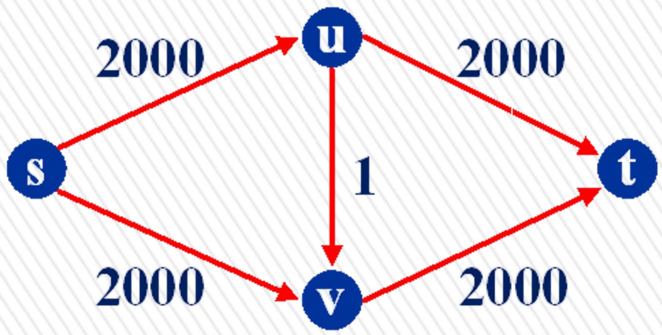
每次所选的增流路径并不一定是最好的

算法复杂性可能会依赖于任选的参数

深度优先搜索可增流路径，导致算法复杂度不确定



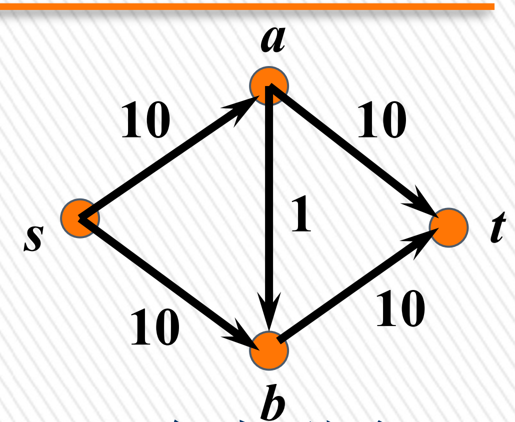
Ford-Fulkerson最大流标号算法



- 最多迭代多少次（即增广的次数）就很难估计，在最坏情况下，与边的容量有关；如上图：先增广$s \to u \to v \to t$，然后增广$s \to v \to u \to t$，每次只能增广 1 个单位，故要增广4000次才能结束
- 克服这种缺点的方法：
  - 尽量先用路径长度最短（段数少）的增广链（SAP）
  - 尽量不重复前面出现过的增广链

最大流的 Edmonds-Karp算法

最大流的Edmonds-Karp算法



Edmonds-Karp算法

严密的标号算法

每次沿一条最短的增流路径增流

Edmonds and Karp在1972年，以及Dinic在1970年都独立的证明了如果每步增广路径都是最短的话，那么整个算法将会执行$O(n*m)$步

广度优先搜索时最坏情况下需$O(m)$次

书上有证明，不做要求

使用广探法 (先标号先检查）$O(n*m*m)$

进一步改进，结合启发式，可提高到$O(n*n*m)$

最大流的Edmonds-Karp算法

Edmonds-Karp算法

Step0. 令f是任意一个流(例如$f = 0$)。给s一个永久标号$(-, \infty)$。

Step1. 标号过程：按先标号先检查的顺序，选择标号最早但尚未检查的点v_i，若所有的点都已检查，说明找不到增流路径，结束。否则对v_i的所有未标号邻点v_j，如果能通过正向或反向标号给以标号，则依次标号v_j

a. 若存在$(v_i, v_j) = a$且$f(a) < c(a)$，则v_j标号(v_i^+, δ_{v_j})，$\delta_{v_j} = \min\{\delta_{v_i}, c(a) - f(a)\}$.

b. 若存在边$(v_j, v_i) = a$且$f(a) > 0$，则给v_j标号(v_i^-, δ_{v_j})，$\delta_{v_j} = \min\{f(a), \delta_{v_i}\}$.

Step2. 若t已被标号，则找到了一条增流路径，转Step3，否则迭代执行step1和2。

Step3. 由点t开始，使用标号的第一个元素构造一条f增流路p。修改f得到新的流f'，以f'代替f，去掉除s外的所有点的f标号。返回Step1。

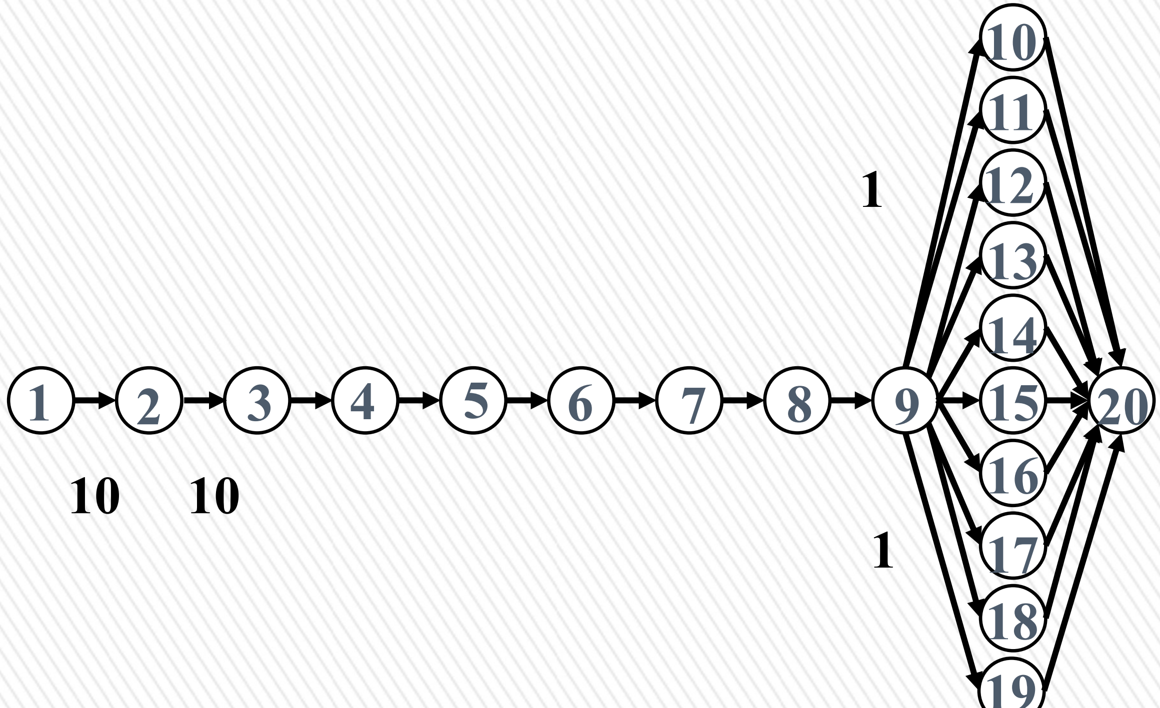
这里

$$f' = \begin{cases} f(a) + \delta_t & \text{若是前向边} \\ f(a) - \delta_t & \text{若是后向边} \end{cases}$$

最大流的Edmonds-Karp算法

存在问题

找到增流路径后，立即沿增流路径对网络流进行增流。
每一次增流可能需要对最多$n-1$条边进行操作。
最坏情况下，每一次增流需要$O(n)$计算时间。
有些情况下，这个代价是很高的。



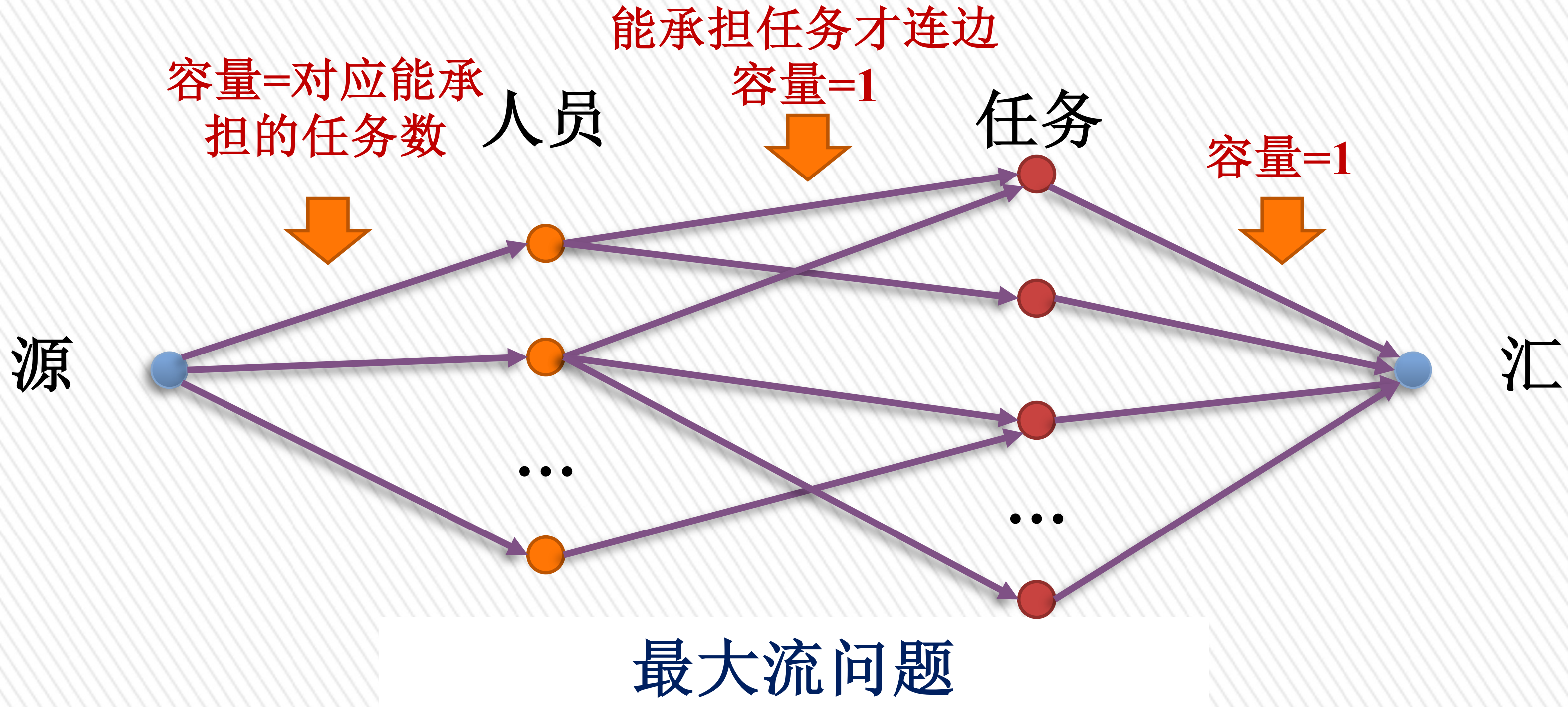
最大流算法

- 增广路算法
  - **Ford-Fulkerson标号算法 (1956)**
  - 最大容量增广路算法（结合**Dijkstra,** 梯度修正）
  - 容量变尺度算法（**1985, Gabow**）
  - 最短增广路算法： $O(n^2m)$ （**Edmonds-Karp**、**Dinic**（分层）、改进的最短增广路方法（距离标号））
- 预流推进算法
  - 推进与重标号算法 **(Push-relabel)** $O(n^2m)$
  - **FIFO**顶点选择策略 $O(n^3)$
  - **Dinic**分层图以及动态树 $O(nm\log n)$
  - 二分查找
  - 。。。

最大流应用问题

例：工作任务分配

- n个人，m项任务；
- 每项任务只能由指定的若干人处理；
- 每个人可处理的任务数量不同（给定常数）；
- 如何分配可以尽可能多的完成任务？

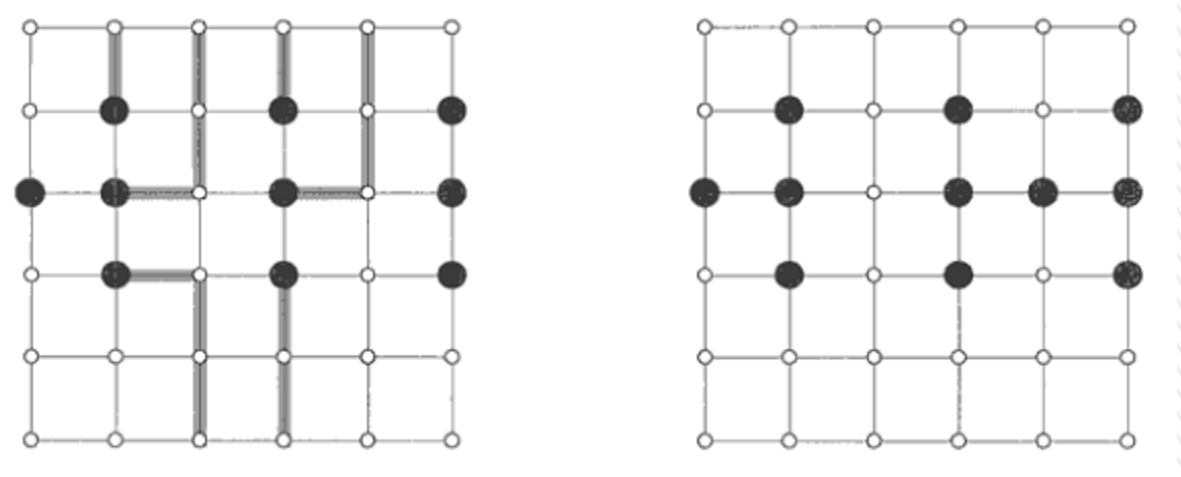


最大流应用问题

例：逃生路线问题

- $n*n$网格节点上有m个人，逃到边上节点就算逃生成功
- 每条边和节点容量均为1，
- 如何规划逃生路线, 使这些路线互不相交？

逃生成功

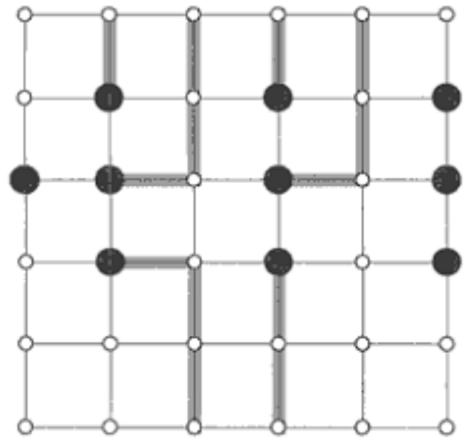


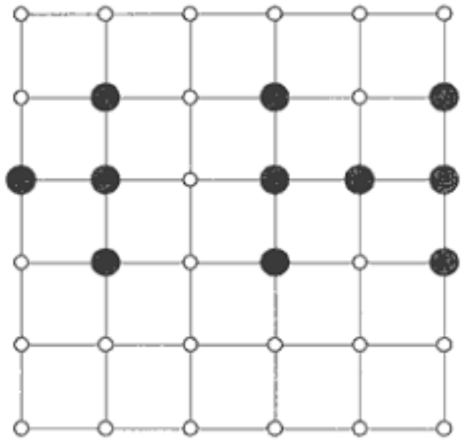
没有逃生路线

可以变成最大流问题

最大流应用问题

逃生成功





没有逃生路线

- *m*个人是供应节点（源，供应量为1）
- 只有边上节点可以是吸收节点（汇，吸收量为1）
- 多源多汇，容易变成单源单汇
- 每条边容量为1
- 每个节点容量为1（通过增加节点和边，变成边容量）

变成最大流问题

最大流应用问题

网络中可能会出现这样的情况：除了边有容量外，点也有容量。

解决的方法是将所有有容量的点分成两个点，如点v有容量C_v，将点v分成两个点v'和v''，令

$$C(v'v'') = C_v$$

最小费用流

（不要求掌握）

最小费用流

(1) 应用背景

不仅要使网上的流达到最大，或者达到要求的预定值，而且还要使运输流的费用是最小的，这就是最小费用流问题。

例：一批货物要从工厂运到车站，可以有多条路线选择，在不同的线路上每吨货的运费不同，而且每条线路的运货能力有限，这时怎样运输才能运费最省？

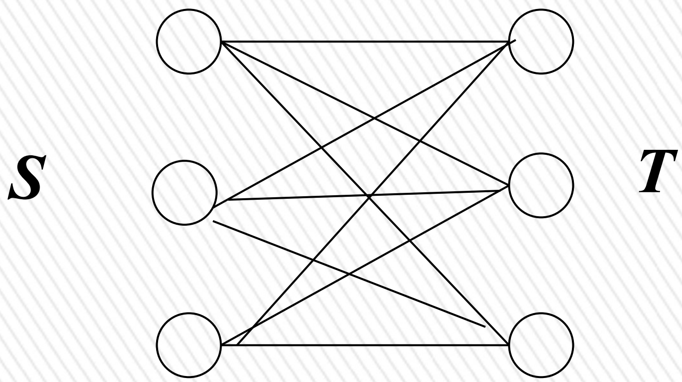
例：一个旅行社接待的一批客人第二天要从甲地飞往乙地，怎样安排才能旅费最省？

最小费用流

(1) 应用背景

例5.8.1： 最佳匹配问题

一家公司经理准备安排N名员工去完成N项任务，每人一项。由于各员工的特点不同，不同的员工去完成同一项任务时所获得的回报是不同的。如何分配工作方案可以使总回报最大？



特殊的最小费用流问题

（二分图，

$|S| = |T| = N$）

最小费用流

(2) 基本概念

- 图论语言

已知网络$G=(V, E, C)$，每条边$v_iv_j\in E$除了已给容量C_{ij}外，还给出了单位流量的费用$a_{ij}(\geq 0)$。

所谓最小费用流问题就是求一个总流量已知的可行流$f=\{f_{ij}\}$使得总费用

$$a(f)=\sum_{v_iv_j\in E} a_{ij} f_{ij}\text{最小。}$$

特别地，当要求f为最大流时，此问题即为最小费用最大流问题。

最小费用流

(3) 算法

- 最小费用流算法
- 原始-对偶算法

  Ford和Forkerson(1957, 1962)
- 瑕疵算法(**Out-Of-Kilter Algorithm**)
- 松弛(**Relaxation**)算法
- 网络单纯形算法

最小费用流

(4) 最短增流路径算法

基本思想：把费用看作边的长度，寻找从s到t的最短的增流路径，它的费用也就增长的最小，如果最后的流量达到w，这时的总费用一般应为最小。

步骤：

① 设网络$G = (V, E, C)$，取初始可行流f为零流，$w_0 = 0$

② 每条边均可看做一对方向相反的边

在当前的容许流分布下，修改各边(i, j)费用a^*_{ij},

当$0 \le f_{ij} < c_{ij}$时，$a^*_{ij} = a_{ij}$, 当$f_{ij} = c_{ij}$时，$a^*_{ij} = \infty$;

当$f_{ji} > 0$时，$a^*_{ij} = -a_{ji}$; 当$f_{ji} = 0$时，$a^*_{ij} = \infty$;

最小费用流

(4) 最短增流路径算法

③ 以a^*_{ij}为边权，找一条从s到t的最短增流路，计算δ。

$$\delta_{ij}=\begin{cases}c_{ij}-f_{ij}, v_iv_j\in\mu^+, v_iv_j\text{与}\mu\text{相同}\\ f_{ij}, \qquad v_iv_j\in\mu^-, v_iv_j\text{与}\mu\text{相反}\end{cases}$$

令$\delta=\min\{\delta_{ij}|v_iv_j\in\mu\}$

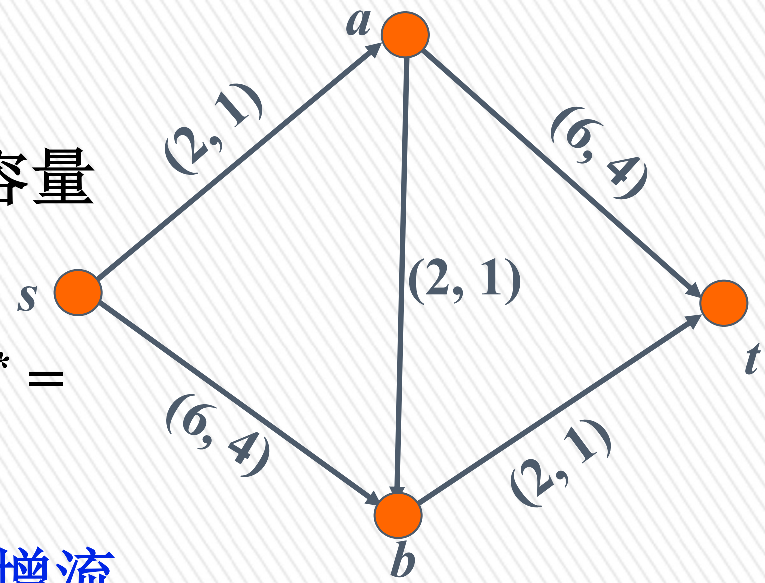
④ 如果$\delta+w_0>w$，则适当减少δ值，令$\delta=w-w_0$。执行5后结束；否则执行5后再回到2。

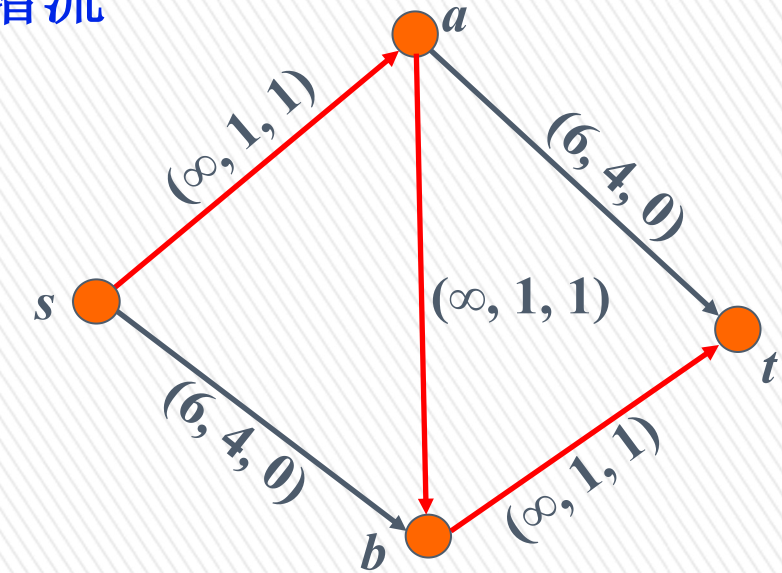
⑤ 增流。$f_{ij}=\begin{cases}f_{ij}+\delta, v_iv_j\in\mu^+,\\ f_{ij}-\delta, v_iv_j\in\mu^-.\end{cases}$

最小费用流

例6.5.3

- 运输网络的每条边都有运价和容量 (a_{ij}, c_{ij})
- 初始流量$w_0 = 0$，各边的费用$a_{ij}^* = a_{ij}$
- $P(s, a, b, t)$是当前的最短增流路增流量$\delta_t = 1$，即总流量$w_0 = 1$
- 增流后每边第3个权为当前流量
- 当前总费用为6（对照原图）
- 修改各边的费用a_{ij}^*

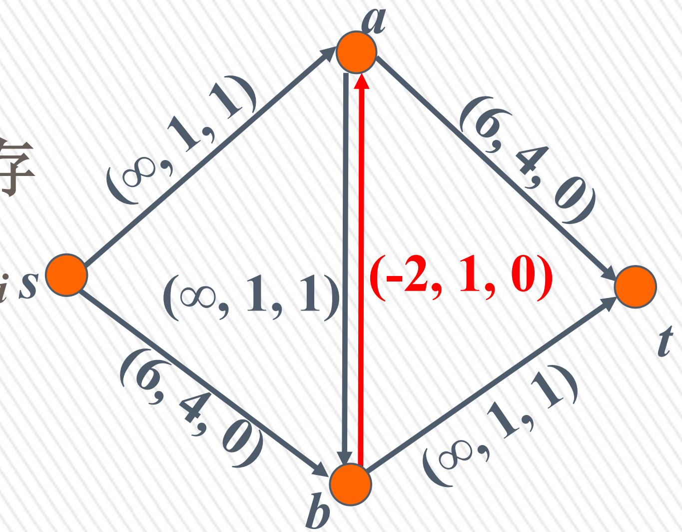




最小费用流

例6.5.3（续）

当边$(i, j)(i, j \neq s, t)$的$f_{ij} > 0$时，就对应存在一条边(j, i)，并且$a_{ji}^* = -a_{ij}$, $c_{ji} = f_{ij}$, $f_{ji} = 0$，再求s到t的最短增流路



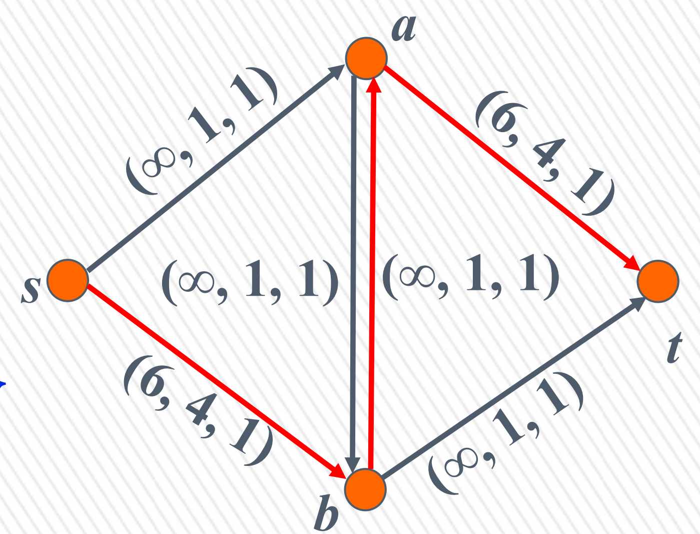
$P = (s, b, a, t)$是当前的最短增流路

$\delta_t = 1$，费用为10

总流量$w_0 = 2$

总费用$\sum a_{ij} f_{ij} = 6 + 10 = 16$

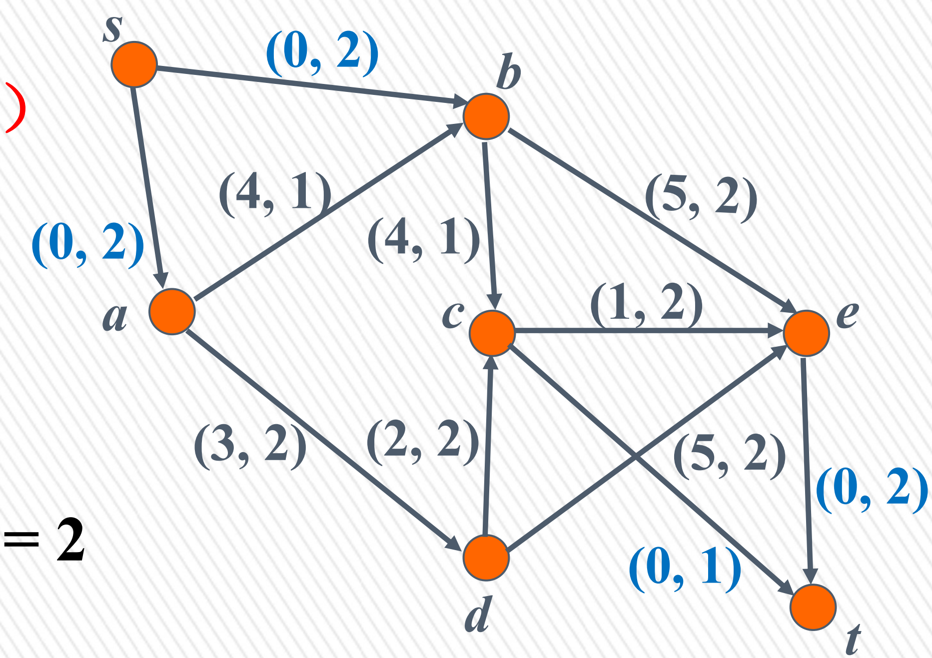
不存在增流路径，获得最小费用流分布



最小费用流

例6.5.4

- 多源多汇网络（花费, 容量）
  - 发点a, b均可供应2个单位
  - 收点c, e各接收1, 2个单位
- 增设一个超发点s
  - $a_{sa} = 0, c_{sa} = 2$；$a_{sb} = 0, c_{sb} = 2$
- 增设一个超收点t
  - $a_{ct} = 0, c_{ct} = 1$；$a_{et} = 0, c_{et} = 2$
- 转为单源单汇

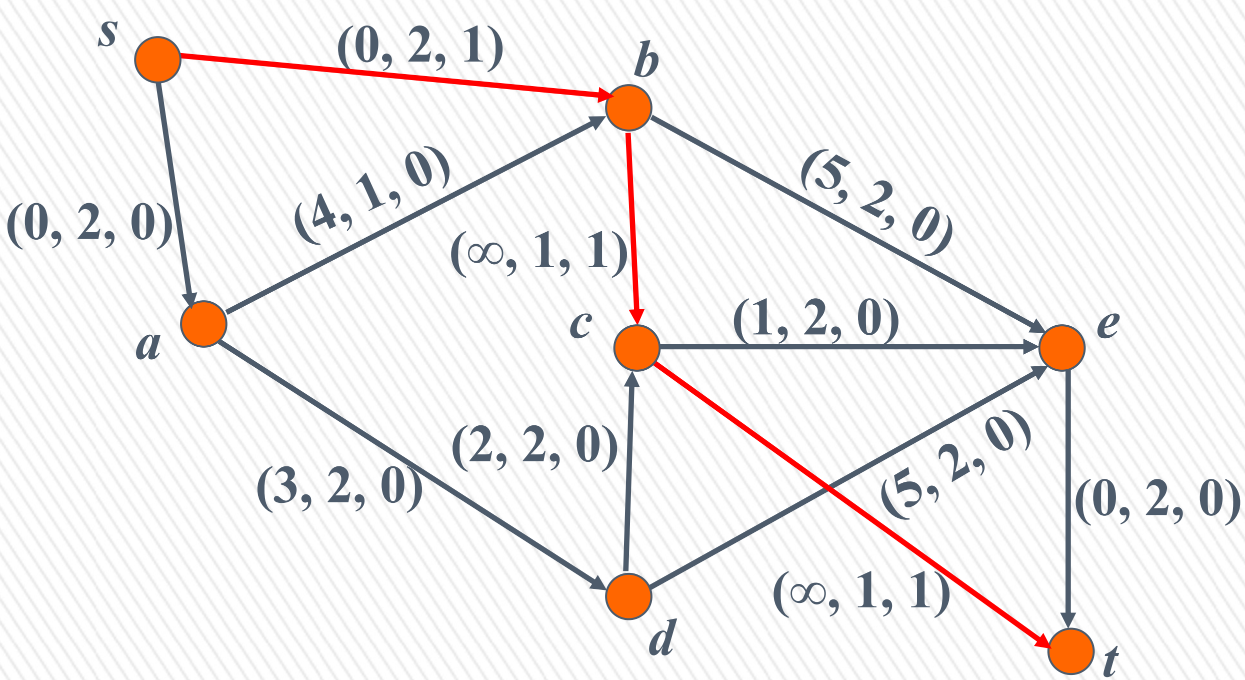


找<s, t>最小花费路径？

$w_0 = 0$，最短增流路径$P_1 = (s, b, c, t), \delta_t = 1$

最小费用流

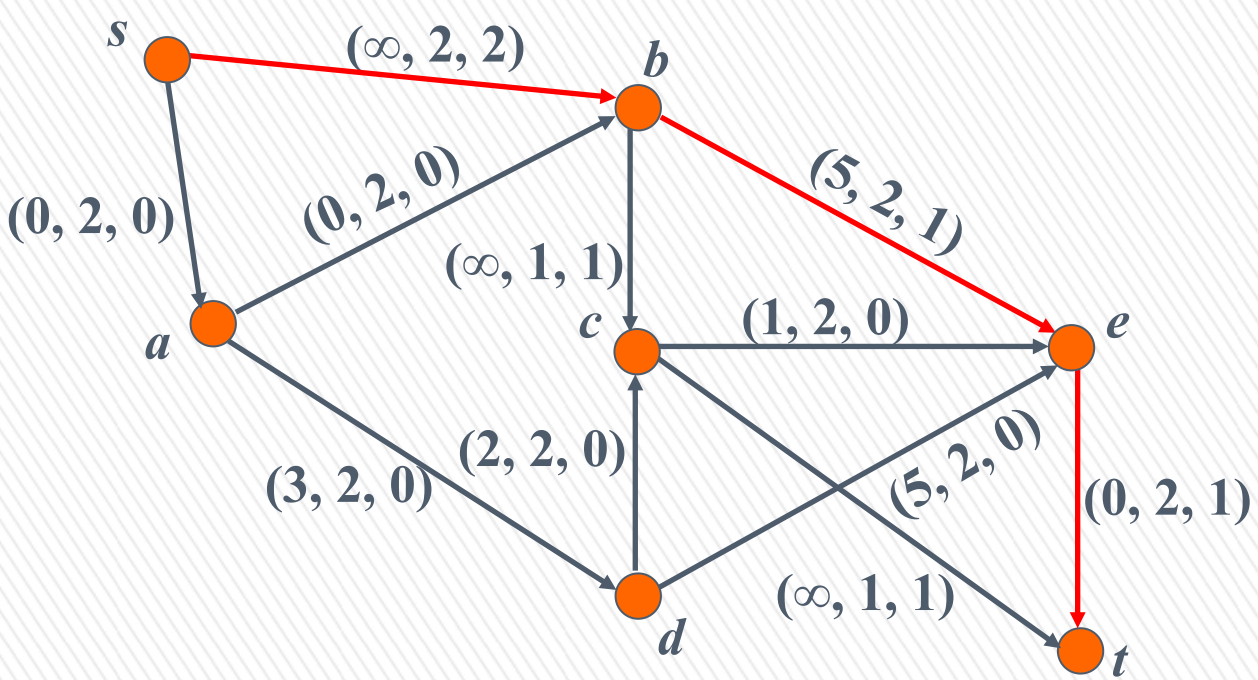
$w_0 = 0$，最短增流路径$P_1 = (s, b, c, t), \delta_t = 1$



$w_0 = 1$，找到最短增流路径$P_2 = (s, b, e, t), \delta_t = 1$

最小费用流

沿(s, b, e, t)增流，再找$<s, t>$可增流最小花费路径？



$w_0 = 2$，最短增流路径$P_3 = (s, a, d, c, e, t), \delta_t = 1$

最小费用流

沿(s, a, d, c, e, t)进行增流

$w_0 = 3$， 完成