最短路径问题的算法

一般网络的最短路径问题可以看成是一个线性规划模型（事实上是一个更特殊的运输模型），可依据对偶性构造其求解算法。

暴力穷举法：由于可能的路径为指数多个，不是一个好方法。
如果基于动态规划的思想，可给出最短路径问题的强多项式时间算法。
以下仅说明具有代表性的算法之一：

Dijkstra's algorithm. 最短路径的性质：若s到v的一条最短路径经过某个顶点c，即 $s \to c \to v$，那么这条路径上子路径 $s \to c$是s到c的最短路径。

算法 DIJKSTRA

输入 有向图$G=(V,E)$, 各边长度$c: E \to \mathbb{R}_+$, 始点$s \in V$.

输出 从始点到所有节点$v \in V$的最短路径及其长度$c^*(v)$.

步一 初始化：令$d(s):=0, d(v):=\infty (v \in V-\{s\})$, 以及$X:=\emptyset$.

步二 迭代：选取一个满足$d(v^*)=\min\{d(v) \mid v \in V-X\}$的节点$v^* \in V-X$.

步三 更新： $X:=X \cup \{v^*\}$.
进一步对$w \in V-X$的各边$e=(v^*,w) \in E$作如下更新：

$$d(w):=\min\{d(w), d(v^*)+d(v^*,w)\}.$$

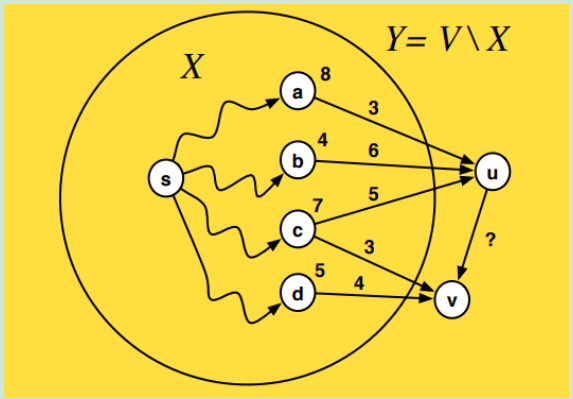
步四 结束判定：如果$P=V$则结束计算；否则回到步二。

最短路径问题的算法

- 步一：把图分为已访问节点 X 和未访问节点 $V - X$, 并记录s到所有点的（临时）最短路径为 $d(v)$. 对于已访问节点$x \in X$， $d(x)$是s到x的最短路径；未访问节点 $y \in V - X$, $d(y)$叫做临时最短路径。
- 步二详解：对于已访问节点集合X，我们寻找X的所有邻居节点中，到X距离最小的顶点。即 $v^* \in V - X$是问题 $\min_{x \in X,\ v \in V-X} (d(x) + w(x, v))$的解。

例

若我们有如下的图，已知s到所有X中所有顶点最短距离。对于 $u, v \in V - X$, 我们得到 $p(v) = d(d) + w(d, v) = 9$.



- 步三：更新已访问集合 $X = X \cup \{v^*\}$, 更新临时距离 $d(w), w \in V - X$.
- 步四：当所有节点均访问过，即 $X = V$时，结束算法；否则继续更新集合X。

最短路径问题实例

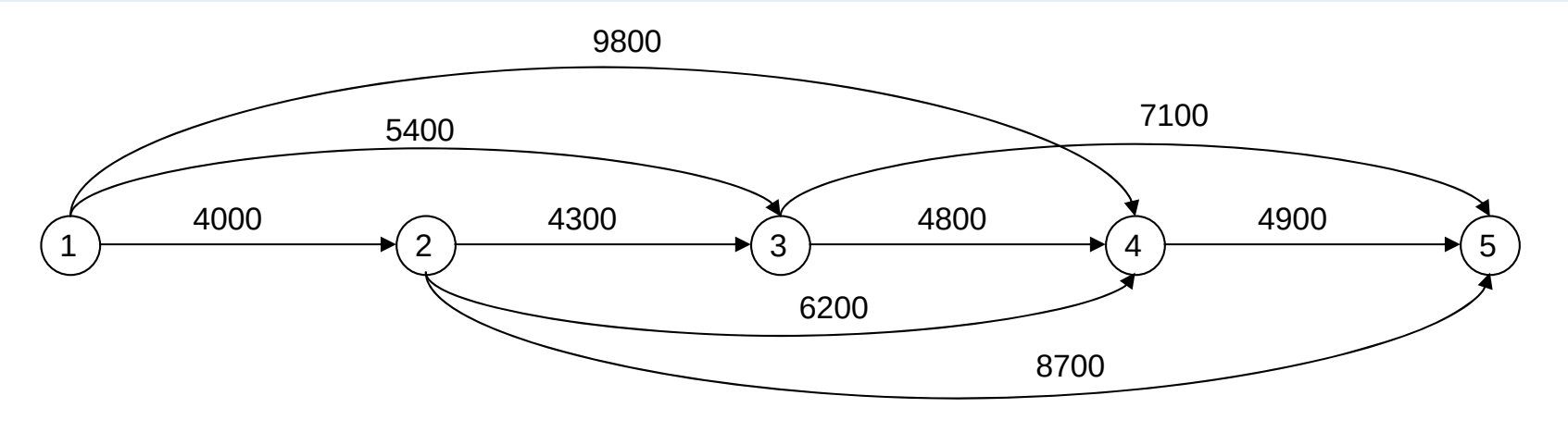
最短路径问题不仅可以应用于直接场景，如路径规划，也有其他的建模实例：某设备租赁公司制定一个未来5年的设备更新规划。租赁的设备可连续提供1-3 年服务，每年结束可以换一台新的，也可以继续运行。已知于各年初更换一台新机器的价格及不同年限机器更换价如表所示。试确定该机器的最优更新策略，使 5年内用于更换总费用为最省。

Table: Equipment replacement

	Replacement cost for given years in operation		
Equipment acquired at start of year	+1	+2	+3
1	4000	5400	9800
2	4300	6200	8700
3	4800	7100	–
4	4900	–	–

最短路径问题实例

最短路径问题的应用实例：设备更新



作业：使用Dijkstra算法计算该问题的最短路。

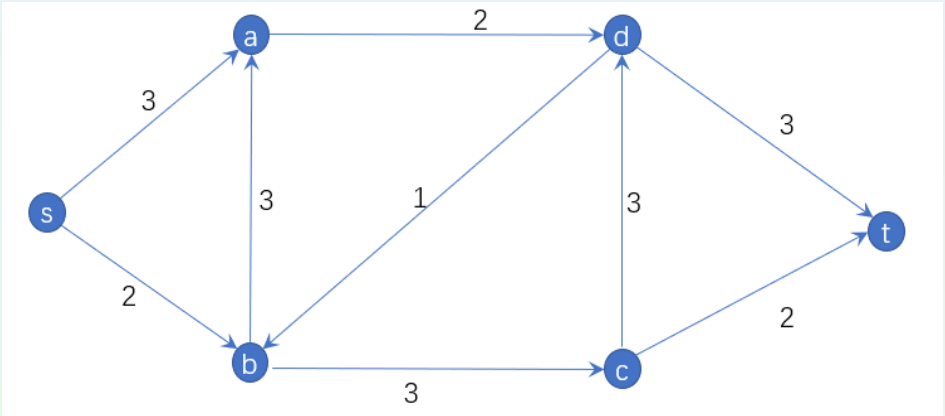
3.3 最大流问题

在有向图$G = (V, E)$上，若图存在边 (u, v), 则不存在(v, u)，每条边$e \in E$有最大容量限制$c(e) > 0$。此外，存在两个特殊的顶点: source 和sink，分别记作 s, t.

定义 (Flow)

流量(flow)是有向图$G = (V, E)$上的一个函数$f : E \to R$, 满足如下条件：

1. **容量约束:** $0 \leq f(e) \leq c(e)$.
2. **流量守恒:** 对于任意$v \in V - \{s, t\}$, 有 $\sum_{e \in \mathrm{In}(v)} f(e) = \sum_{e \in \mathrm{Out}(v)} f(e)$, 这里$\mathrm{Out}(v)$和$\mathrm{In}(v)$分别表示$G$中流出和流入$v$节点的边集合。



f也可以写成顶点的函数，即$f : V \times V \to R$. 相应地,

1. **容量约束:** 若$(u, v) \in E$, 则$0 \leq f(u, v) \leq c(u, v)$. (若$u, v$不直接相连, $f(u, v) = 0$.)
2. **流量守恒:** 对于任意$u \in V - \{s, t\}$, 有 $\sum_{v \in V} f(u, v) = 0$.
3. **反对称：** $f(u, v) = -f(v, u)$.

最大流问题

定义

从s的纯流出量，定义为流f的值,记作 $|f|$,

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

引理

- $f(X, X) = 0$,
- $f(X, Y) = -f(Y, X)$
- $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$, *if* $X \cap Y = \emptyset$.
- $f(u, V) = 0$, $\forall u \in V - \{s, t\}$.

最大流问题

定理

流的值$|f|$等于流出t的流的和：

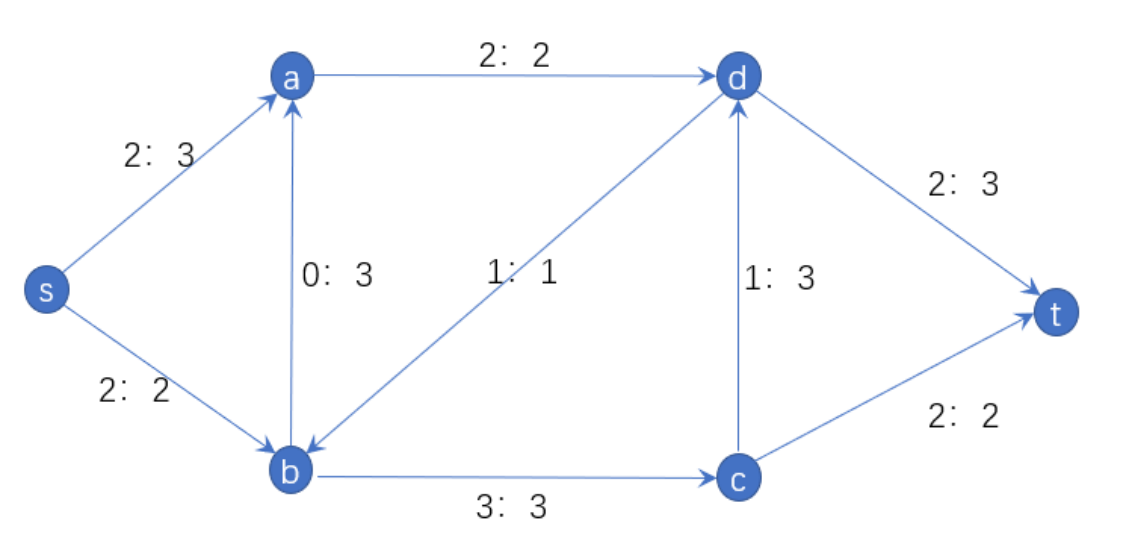
$$|f| = \sum_{v \in V} f(v, t) = f(V, t).$$

证明：

$$\begin{aligned} |f| &= f(s, V) \\ &= f(V, V) - f(V - s, V) \\ &= f(V, V - s) \\ &= f(V, t) + f(V, V - s - t) \\ &= f(V, t). \end{aligned} \tag{46}$$

最大流问题

所谓最大流问题，就是求出使流值达最大的可行流问题，



线性规划可描述如下：

$$\begin{aligned} \max \quad & f(s,V) = \textstyle\sum_{v:(s,v)\in E, v\in V} f(s,v) \\ \text{s.t.} \quad & \sum_{v\in V} f(v,u) = \textstyle\sum_{w\in V} f(u,w), \forall u \in V - s - t \\ & 0 \le f(u,v) \le c(u,v), (u,v) \in E. \end{aligned} \tag{47}$$

最小割问题

定义

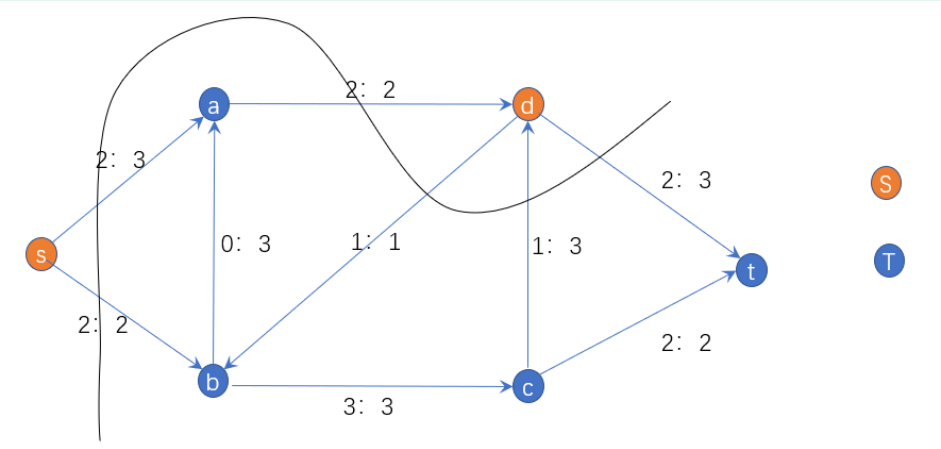
流网络$G = (V, E)$的一个割(S, T)是指，顶点集合V的一个分割，使得 $s \in S, t \in T$. 如果f是G的流量，那么一个割上的流量为$f(S, T)$.

也就是说，割把一个流网络的顶点集划分成两个集合S和T, 使得源点$s \in S$且汇点$t \in T$.

定义

分割(S, T)的容量为

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



如图，分割的容量为$c(S, T) = 3 + 2 + 1 + 3 = 9$. 割流量为$f(S, T) = 2 + 2 + (-2) + 1 + (-1) + 2) = 4$.

最大流最小割定理

所谓最小割问题，即找到一种割法，使割的容量最小。也就是说，找到通过能力最弱的断面。

引理

对于任意割，我们有$|f| = f(S, T)$. 另外$|f| \leq c(S, T)$.

Proof.

$$\begin{aligned} f(S, T) &= f(S, V) - f(S, S) \\ &= f(S, V) \\ &= f(s, V) + f(S - s, V) \\ &= f(s, V) \\ &= |f|. \end{aligned} \tag{48}$$

$$\begin{aligned} |f| &= f(S, T) \\ &= \sum_{u \in S} \sum_{v \in T} f(u, v) \end{aligned} \tag{49}$$

最大流最小割定理

定理

最大流最小割定理： 任意一个流网络的最大流量等于该网络的最小割的容量。

可以写出最大流线性规划问题(47)的对偶问题，并且其是最小割问题的对偶问题。

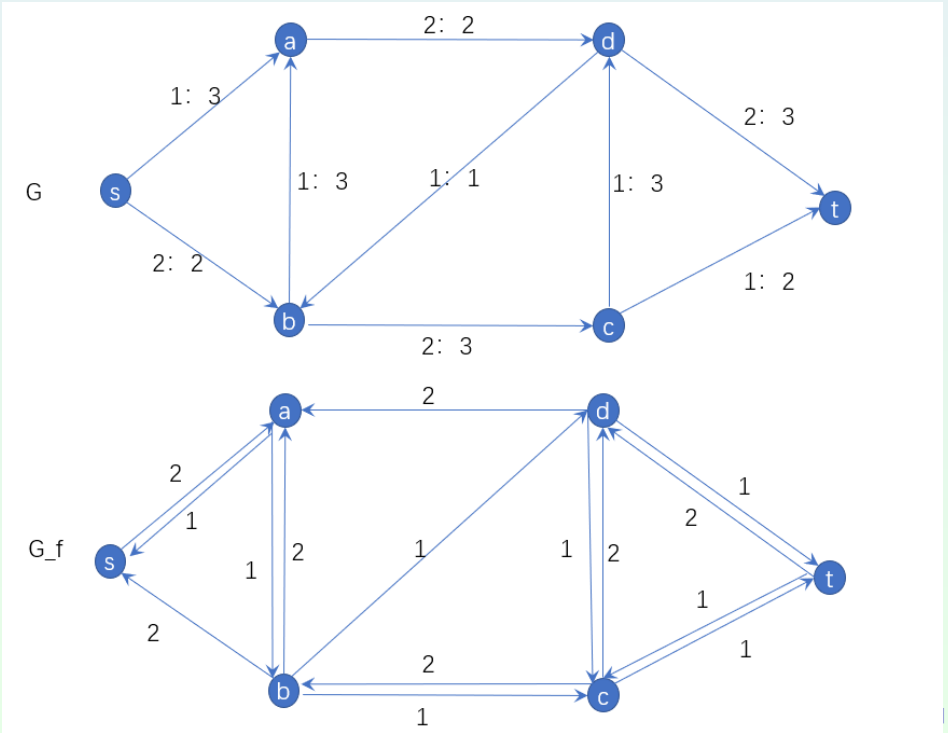
余网络（residual network)和流扩充路（augmenting path）

网络$\mathcal{N}=(G,s,t,\mathbf{c})$中给定一个可行流$f$， **余网络** 构造方式如下：

对于$\mathcal{N}$的各边$e=(u,v)\in E$，按照以下规则生成边(u,v)或(v,u)，得到的有向边集合记为E_f，同时确定其容量c_f.

R-1 如果$c(e)-f(e)>0$，则生成$(u,v)\in E_f$，并令其容量$c_f(u,v)=c(u,v)-f(u,v)$.
R-2 如果$f(u,v)>0$，则生成$(v,u)\in E_f$，并令其容量$c_f(v,u)=f(u,v)$.

所得到的有向网中所有容量$\bar{c}>0$的边构成余网络$\mathcal{N}_f=(G_f,s,t,c_f)$，其中有向图$G_f=(V,E_f)$，容量$c_f=\{c_f(e)>0\mid\forall e\in E_f\}$. $\mathcal{N}_f$中从s到t的路称为**流扩充路**。

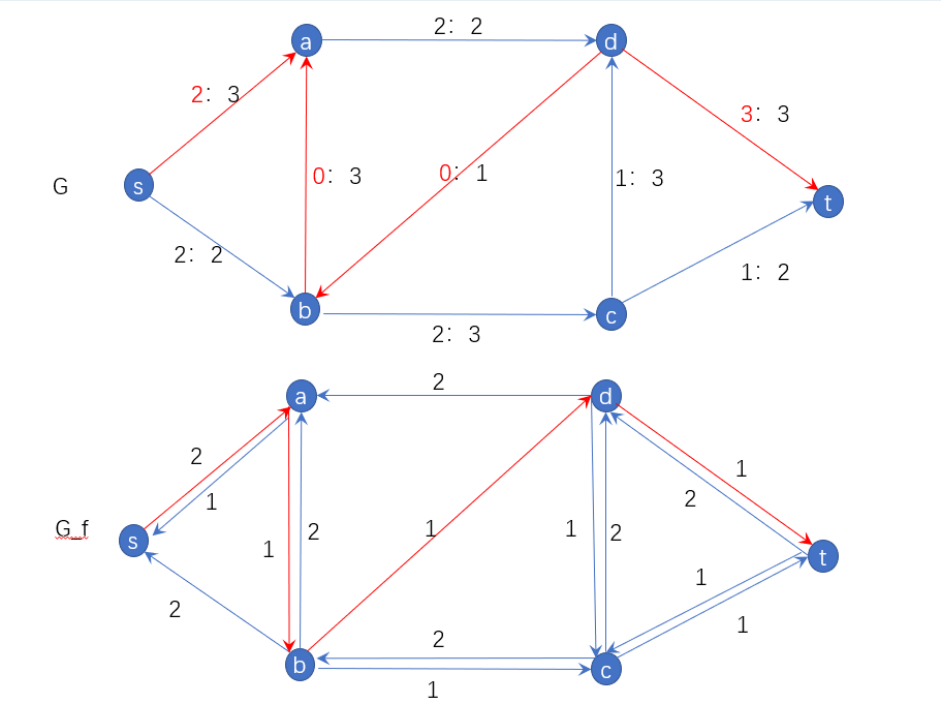


最大流问题的算法

引理

（流扩充路）给定网络$\mathcal{N} = (G, s, t, c)$, 其可行流$f$为最大流的充要条件是，余网络$\mathcal{N}_f$中不存在有流扩充路。

如果存在有流扩充路p，则f可修正为流值更大的流,通过沿着p对f增加$c_f(p) = \min_{(u,v)\in p}\{c_f(u, v)\}$. (若$G_f$中的边与$G$中的有向边相反，则减去相应的值。)



最大流问题的算法

算法 Ford-Fulkerson Algorithm:

输入 网络$\mathcal{N}=(G,s,t,c)$, 其中有向图$G=(V,E)$.

输出 从s到t的最大流值$f_{\max}$.

步一 初始化：令$f(e):=0(e\in E)$以及$f_{\max}:=0$.

步二 余网络：构造当前可行流f的余网络$\mathcal{N}_f=(G_f,s,t,c_f)$.

步三 流的扩充：如果$\mathcal{N}_f$中没有流的扩充路则结束计算。相反，如果存在有流的扩充路，则选取其一p, 通过沿着p对f增加$c_f(p)=\min_{(u,v)\in p}\{c_f(u,v)\}$. (若$G_f$中的边与$G$中的有向边相反，则减去相应的值。) 令$f_{\max}=f+c_f(p)$. 回到步二。

3.4 最小成本流问题

考虑网络$\mathcal{N} = (G, s, t, \mathbf{c}, w)$上的流$f$, 其中$w(u, v)$是边$(u, v)$的费用，其中流值固定为

$$|f| = f^*.$$

在流值$|f| = f^*$不超过网络的最大流的条件下，求出使成本达最小的流就是所谓最小成本流问题。

例子：

有足够多辆卡车要将数量无限的某种物品从一个地点运输到另外一个地点，现在有有限条单向行驶道路直接或者间接地连接了这两地。但是每一条道路都有运输通过总数量的限制，称为容量，同时携带物品通过该路段时，都会按照携带物品数量多少被收取一定的费用。如何合理地安排每辆车的行驶路线，使得在完成一定运输量的情况下，交付的总费用尽可能少？

注意，在此问题中总费用仅包括携带物品通过路段时被收取的费用，车辆和路线安排上没有限制，但通过某一路段的物品数量总和不得超过它的容量，收取的费用与携带物品的多少成正比。

最小成本流问题

最小成本流问题的数学模型：

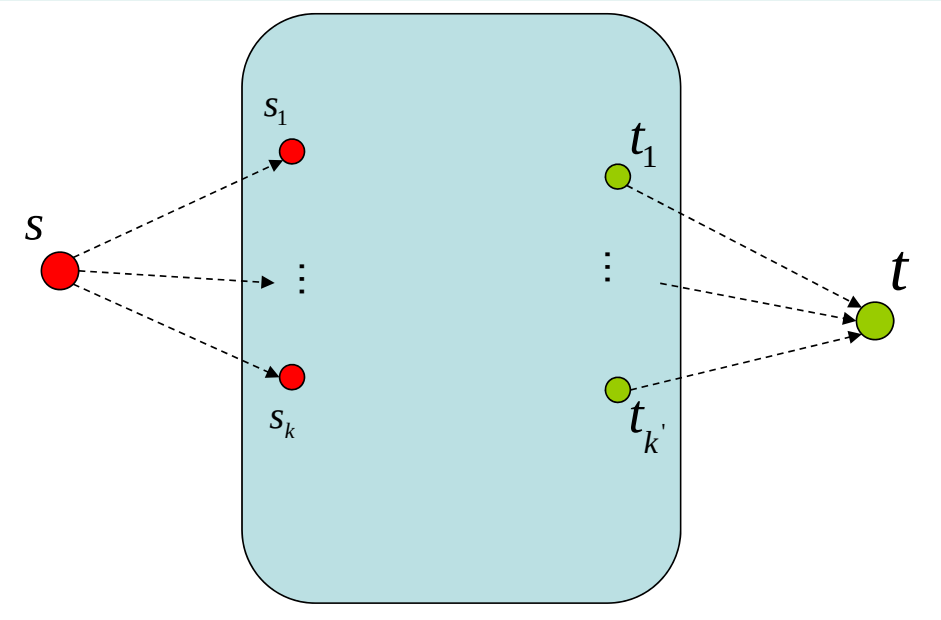
$$
\begin{aligned}
\min \quad & \sum_{e\in E} w(e)f(e) \\
\text{s.t.} \quad & \sum_{e\in \mathrm{Out}(v)} f(e) - \sum_{e\in \mathrm{In}(v)} f(e) = 0, v \in V - \{s,t\} \\
& \sum_{e\in \mathrm{Out}(s)} f(e) - \sum_{e\in \mathrm{In}(s)} f(e) = f^* \\
& 0 \le f(e) \le c(e), e \in E.
\end{aligned} \tag{50}
$$

最小成本流问题

在最小成本流问题中，可以有多个源点和汇点。如果最小成本流问题具有k个源点$s_i, i = 1, \cdots, k$以及各自的流出量$f(s_i) = f^*(s_i)$, 还有k'个汇点$t_j, j = 1, \cdots, k'$以及各自的流出量$f(t_j) = f^*(t_j)$, 则可以引入哑元源点s和哑元汇点t, 并添加$(k + k')$条边

$$(s, s_i): \ w(s, s_i) = 0, c(s, s_i) = f^*(s_i), i = 1, \cdots, k;$$

$$(t_j, t): \ w(t_j, t) = 0, c(t_j, t) = f^*(t_j), j = 1, \cdots, k'.$$



最小成本流问题

最小成本流问题具有很好的模型化能力：

（一）通过引入哑元终点t', 加入从$v \in V$出发的边(v, t')且满足

$$w(v, t') = 0, \quad c(v, t') = 1.$$

那么，最短路径问题成为求解从s到t'的具有流值$f^* = n$的最小成本流问题。这里，$n = |V|$, 并假定原网络各边的容量全为∞。

最小成本流问题

最小成本流问题具有很好的模型化能力：

（二）通过引入哑元始点s'，并构造如下两条边：

$$(s', s); \quad w(s', s) = 0, \quad c(s', s) = \infty$$

$$(s', t); \quad w(s', t) = 1, \quad c(s', t) = \infty$$

那么，最大流问题成为求解从s'到t的最小成本流问题。这里，设定原网络里$w(e) = 0(e \in E)$，并取最大流值的适当上界（比如 $\sum_{e \in E} c(e)$）为从s'出发的流值f^*.