第 9 章 网络流理论

赵功名

中国科学技术大学计算机科学与技术学院

目录

例 9.1 假设将某个商品从产地通过公路或铁路运到消费市场，每段公路或铁路的运载能力有限，如何安排每段路程的运输方案，使得该商品能够尽快运给客户？

例 9.2 给定一个计算机网络，每段网络或每个路由器的带宽有限，要将一个文件从某个服务器传给一些用户，如何进行路由，选择恰当的数据传输方案，使得该文件能够尽快传给用户？

以上问题可以抽象为“网络”，用网络流理论来解决.

定义 9.1 **一个网络可以定义为一个四元组** $N = (D, s, t, c)$，**其中**:

(1) D 是一个弱连通的有向图.

(2) $s, t \in V(D)$，**分别称为源与汇**.

(3) $c : E(D) \to \mathbf{R}$ **为容量函数**. 任给 $e \in E(D)$，$c(e) \geq 0$ **为边** e **的容量**.

定义 9.2 网络 $N=(D,s,t,c)$ 上的流函数为 $f\colon E(D)\to \mathbf{R}$，要求满足：

(1) 任给 $e\in E(D)$，都有 $c(e)\ge f(e)\ge 0$， (9.1)

(2) 任给 $v\in V(D)-\{s,t\}$，都有 $\sum_{e\in\alpha(v)} f(e)-\sum_{e\in\beta(v)} f(e)=0.$ (9.2)

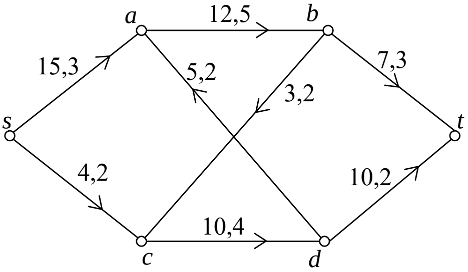
其中，$\alpha(v)$ 是所有以 v 为头的边集，而 $\beta(v)$ 则是所有以 v 为尾的边集. f 的流量定义为

$$
\begin{aligned}
Val(f) &= \sum_{e\in\alpha(t)} f(e)-\sum_{e\in\beta(t)} f(e)\\
&= \sum_{e\in\beta(s)} f(e)-\sum_{e\in\alpha(s)} f(e). \qquad (9.3)
\end{aligned}
$$

我们的目标是找到流量最大的流函数 f^*，即

$$Val(f^*)=max_{f\text{是}N\text{的流函数}} Val(f).$$

例 9.3 如图为一个非常简单的网络，其中每条边上标的第一个数值为该边的容量，第二个参数就是流函数在该边上的流量，流函数的流量为 $Val(f) = 5$



定义 9.3 给定网络 $N=(D,s,t,c)$，$S\subset V(D)$，满足 $s\in S$，$t\in\overline{S}=V(D)-S$，则称

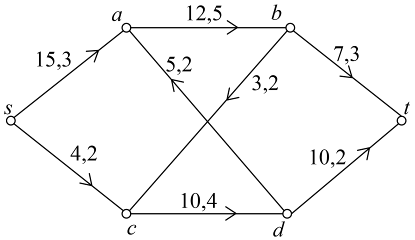
$$(S,\overline{S})=\{e=(u,v)|e\in E(D),u\in S,v\in\overline{S}\}$$

为网络 N 的一个**截**，而称

$$C(S,\overline{S})=\sum_{e\in(S,\overline{S})}c(e) \tag{9.4}$$

为 $(S,\overline{S})$ 的**截量**. 截量最小的截称为**最小截**.

例 如图，取 $S_1=\{s,a\}$，则 $\bar{S}_1=\{b,c,d,t\}$，$(S_1,\bar{S}_1)=\{(s,c),(a,b)\}$ 就是一个截，其截量为 $C(S_1,\bar{S}_1)=c((s,c))+c((a,b))=4+12=16$.
取 $S_2=\{s,a,b\}$，则 $\bar{S}_2=\{c,d,t\}$，$(S_2,\bar{S}_2)=\{(s,c),(b,c),(b,t)\}$ 就是一个截，其截量为 $C(S_2,\bar{S}_2)=c((s,c))+c((b,c))+c((b,t))=4+3+7=14$.
$C(S_2,\bar{S}_2)$ 是该网络的最小截. 截的含义在于，若将一个截中所有的边从网络中移除，则网络中不存在从 s 到 t 的有向轨道，流函数的流量自然为 0. 所以说，一个截的截量揭示了网络流量的上界.



定理 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，则有

$$Val(f)=\sum_{e\in(S,\overline{S})} f(e)-\sum_{e\in(\overline{S},S)} f(e).$$

证明 由流函数的定义知，

$$\sum\nolimits_{e\in\alpha(t)} f(e)-\sum\nolimits_{e\in\beta(t)} f(e)=Val(f) \qquad (9.5)$$

$$\sum\nolimits_{e\in\alpha(v)} f(e)-\sum\nolimits_{e\in\beta(v)} f(e)=0, \qquad v\in V(D)-\{s,t\}. \qquad (9.6)$$

对于 $\overline{S}$ 中所有顶点来说，t 满足公式 (9.5)，而其余所有的顶点都满足公式 (9.6). 将上述公式 (9.5) 与 (9.6) 对 $\overline{S}$ 中所有顶点求和，得到

$$\sum\nolimits_{v\in\overline{S}}(\sum\nolimits_{e\in\alpha(v)} f(e)-\sum\nolimits_{e\in\beta(v)} f(e))=Val(f). \qquad (9.7)$$

下面我们对图 D 中所有的边进行分析，看看不同类型的边对公式 (9.7) 的贡献如何. 对于图 D 中有向边 $e=(x,y)$ 来说，共有如下四种情况：

(1) $x,y\in\bar{S}$. 由于 x 与 y 都在 $\bar{S}$，公式 (9.7) 中出现两次 $f((x,y))$. 一次是对顶点 x 求和，x 是边 (x,y) 的尾，$f((x,y))$ 以负项出现；另一次是是对顶点 y 求和，y 是边 (x,y) 的头，$f((x,y))$ 以正项出现. 两者抵消，对 (9.7) 式的贡献为 0.

(2) $x\in S$，$y\in\bar{S}$，则 $(x,y)\in(S,\bar{S})$. 由于仅有 $y\in\bar{S}$，y 是边 (x,y) 的头，在 (9.7) 式中，$f((x,y))$ 在对顶点 y 求和时以正项出现，对 (9.7) 式贡献为 $f((x,y))$.

(3) $y\in S$，$x\in\bar{S}$，则 $(x,y)\in(\bar{S},S)$. 由于仅有 $x\in\bar{S}$，x 是边 (x,y) 的尾，在 (9.7) 式中，$f((x,y))$ 在对顶点 x 求和时以负项出现，对 (9.7) 式贡献为 $-f((x,y))$.

(4) $x,y\in S$，$f((x,y))$ 在 (9.7) 式中不出现，贡献为 0.

综上可知，对于图 D 中有向边 e 来说，若 $e\in(S,\bar{S})$，则 e 对公式 (9.7) 贡献为 $f(e)$；若 $e\in(\bar{S},S)$，则 e 对公式 (9.7) 贡献为 $-f(e)$；而其余的边对公式 (9.7) 的贡献为 0. 证毕.

推论 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，则有

$$Val(f)\le C(S,\overline{S}).$$

证明 由定理 9.1 知，

$$\begin{aligned} Val(f) &= \sum_{e\in(S,\overline{S})} f(e) - \sum_{e\in(\overline{S},S)} f(e) \\ &\le \sum_{e\in(S,\overline{S})} c(e) - \sum_{e\in(\overline{S},S)} 0 \\ &= C(S,\overline{S}). \end{aligned}$$

证毕.

推论 9.2 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，若 $Val(f)=C(S,\overline{S})$，则 f 是最大流，$(S,\overline{S})$ 是最小截.

证明 假设 f' 是最大流，$(S',\overline{S'})$ 是最小截，由流量与截量的定义知，

$$Val(f)\le Val(f') \quad \text{且} \quad C(S',\overline{S'})\le C(S,\overline{S}),$$

而由推论 9.1 知，

$$Val(f')\le C(S',\overline{S'}),$$

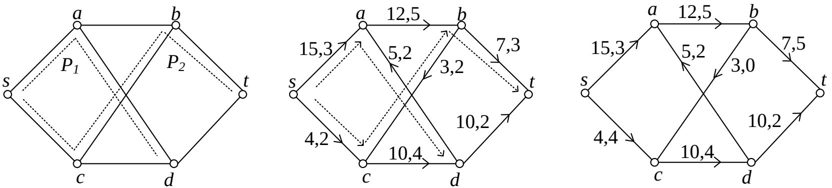
所以有

$$Val(f)\le Val(f')\le C(S',\overline{S'})\le C(S,\overline{S}),$$

在 $Val(f)=C(S,\overline{S})$ 的前提下，上述不等式必须全部为等号才能成立. 所以，f 是最大流，$(S,\overline{S})$ 是最小截. 证毕.

设有向图 D 对应的底图为 G，那么将 D 中所有的边方向略去，得到的无向图是 G. 设 $P(s,u)$ 为 G 中一条以 s 为起点、u 为终点的无向轨道. 我们在 G 中规定 $P(s,u)$ 的方向为从 s 到 u. 而 $P(s,u)$ 上每条无向边 $\bar{e}$ 都对应于 D 中的一条有向边 e. 若 e 的方向与 $P(s,u)$ 的方向相同，则称 e 为 P 的**正向边**；否则称为 P 的**反向边**.

例 9.4 如图为例 9.3 中的有向图对应的底图，$P_1(s,d) = sad$ 是底图中一条以 s 为起点的轨道，方向为从 s 到 d. 对应到原来的有向图中的边，(s,a) 为 P_1 的正向边，(a,d) 为 P_1 的反向边. 同理，$P_2(s,t) = scbt$ 也是底图中一条以 s 为起点的轨道，方向为从 s 到 t.(s,c) 与 (b,t) 均为 P_2 的为正向边，(c,b) 为 P_2 的反向边.



(a) (b) (c)

定义 9.4 给定网络 $N=(D,s,t,c)$，N 上的流函数 f，底图 G 中的无向轨道 $P(s,u)$.

(1) 若 e 是 $P(s,u)$ 的正向边，且 $f(e)<c(e)$，则称 e 为**未满载边**；

(2) 若 e 是 $P(s,u)$ 的正向边，且 $f(e)=c(e)$，则称 e 为**满载边**；

(3) 若 e 是 $P(s,u)$ 的反向边，且 $f(e)=0$，则称 e 为**零载边**；

(4) 若 e 是 $P(s,u)$ 的反向边，且 $f(e)>0$，则称 e 为**正载边**.

对于无向轨道 $P(s,u)$，我们定义 $P(s,u)$ 上每条边 e 的**可增载量** $l(e)$ 为：

$$l(e)=\begin{cases} c(e)-f(e); & e\text{是正向边} \\ f(e). & e\text{是反向边} \end{cases}$$

而 $P(s,u)$ 的可增载量 $l(P)$ 则定义为：

$$l(P)=min_{e\in E(P)}l(e).$$

定义 9.5 给定网络 $N=(D,s,t,c)$，N 上的流函数为 f，以及 N 中的无向轨道 $P(s,v)$.

(1) 若 $l(P)>0$，则称 $P(s,v)$ 是未满载轨道；

(2) 若 $l(P)=0$，则称 $P(s,v)$ 是满载轨道；

(3) 若 $l(P)>0$ 且 $v=t$，则称 $P(s,t)$ 是 N 上关于 f 的可增载轨道.

引理 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$P(s,t)$ 是 N 上关于 f 的可增载轨道，定义新的函数 $\bar{f}: E(D)\to \mathbf{R}$ 为

$$\bar{f}(e)=\begin{cases} f(e)+l(P), & e\text{是正向边}\\ f(e)-l(P), & e\text{是反向边}\\ f(e), & e\text{其它}\end{cases}$$

则 $\bar{f}$ 是网络 N 的流函数，且 $Val(\bar{f})=Val(f)+l(P)$.

证明 首先证明任给 $e \in E(D)$，都有 $c(e) \geq \bar{f}(e) \geq 0$。因为 f 是 N 的流函数，所以任给 $e \in E(D)$，都有 $c(e) \geq f(e) \geq 0$。取 $e' \in E(D)$，若 e' 不是 $P(s,t)$ 上的边，则有 $\bar{f}(e') = f(e')$，所以 $c(e') \geq \bar{f}(e') \geq 0$；若 e' 是 $P(s,t)$ 的正向边，由 $l(P)$ 的定义 $l(P) = min_{e \in E(P)} l(e)$ 知，$l(e') \geq l(P) \geq 0$，而 $l(e') = c(e') - f(e')$，故有 $c(e') = f(e') + l(e') \geq f(e') + l(P) = \bar{f}(e') \geq 0$；若 e' 是 $P(s,t)$ 的反向边，证明类似。

下面证明任给 $v \in V(D) - \{s,t\}$，都有 $\sum_{e \in \alpha(v)} \bar{f}(e) - \sum_{e \in \beta(v)} \bar{f}(e) = 0$。若 v 不是 $P(s,t)$ 上的顶点，则任给 $e \in \alpha(v)$ 或 $e \in \beta(v)$，都有 $\bar{f}(e) = f(e)$，所以 $\sum_{e \in \alpha(v)} \bar{f} - \sum_{e \in \beta(v)} \bar{f}(e) = \sum_{e \in \alpha(v)} f(e) - \sum_{e \in \beta(v)} f(e) = 0$。若 v 是 $P(s,t)$ 上的顶点，不妨设 $P(s,t) = s \ldots e_1 v e_2 \ldots t$，$e_1$ 和 e_2 都有可能是 $P(s,t)$ 的正向边或反向边，共四种情形。我们取 e_1 和 e_2 都是正向边这种情形给出证明，其它情形类似。

由于 e_1 和 e_2 都是正向边，$e_1 \in \alpha(v)$，$e_2 \in \beta(v)$，且 $\bar{f}(e_1) = f(e_1) + l(P)$，$\bar{f}(e_2) = f(e_2) + l(P)$。对于 $\alpha(v)$ 和 $\beta(v)$ 中其余的边 e，则有 $\bar{f}(e) = f(e)$。因此

$$
\begin{aligned}
&\sum_{e\in\alpha(v)} \bar{f}(e) - \sum_{e\in\beta(v)} \bar{f}(e) \\
=&[\sum_{e\in\alpha(v)-\{e_1\}} \bar{f}(e) + \bar{f}(e_1)] - [\sum_{e\in\beta(v)-\{e_2\}} \bar{f}(e) + \bar{f}(e_2)] \\
=&[\sum_{e\in\alpha(v)-\{e_1\}} f(e) + (f(e_1 + l(P))] - [\sum_{e\in\beta(v)-\{e_2\}} f(e) + (f(e_2 + l(P))] \\
=&[\sum_{e\in\alpha(v)} f(e) + l(P)] - [\sum_{e\in\beta(v)} f(e) + l(P)] \\
=&0.
\end{aligned}
$$

综上，$\bar{f}$ 是 N 的流函数。最后再证明 $Val(\bar{f}) = Val(f) + l(P)$。

设 $P(s,t) = s \ldots v \ldots e_3 t$，$e_3$ 可能是正向边，也可能是反向边。我们取 e_3 是正向边这种情形来证明，是反向边时类似。因为 e_3 是正向边，$e_3 \in \alpha(t)$ 且 $\bar{f}(e_3) = f(e_3) + l(P)$，对于 $\alpha(t)$ 和 $\beta(t)$ 中其余的边 e，则有 $\bar{f}(e) = f(e)$。所以有

$$
\begin{aligned}
Val(\bar{f}) &= \sum_{e \in \alpha(t)} \bar{f}(e) - \sum_{e \in \beta(t)} \bar{f}(e) \\
&= [\sum_{e \in \alpha(t) - \{e_3\}} \bar{f}(e) + \bar{f}(e_3)] - \sum_{e \in \beta(t)} \bar{f}(e) \\
&= [\sum_{e \in \alpha(t) - \{e_3\}} f(e) + (f(e_3 + l(P))] - \sum_{e \in \beta(v)} f(e) \\
&= [\sum_{e \in \alpha(t)} f(e) - \sum_{e \in \beta(t)} f(e)] + l(P) \\
&= Val(f) + l(P).
\end{aligned}
$$

证毕。

算法 9.1 可增载轨道算法

输入：网络 $N=(D,s,t,c)$，流函数 f.

输出：一条可增载轨道，或指出当前流函数是最大流.

(1) $S=s$；令 $prev(s)=*$.

(2) 若 $t\in S$，则已经找到可增载轨道，通过 $prev(t)$ 回溯输出可增载轨道，算法停止；否则，转第 (3) 步.

(3) 若存在 $u\in S$，$v\in\bar{S}$，使得 $(u,v)\in E(D)$ 且边 (u,v) 未满载，即 $f((u,v))<c((u,v))$（(u,v) 是正向边），则令 $S\leftarrow S\cup\{v\}$，$prev(v)=u$，转第 (2) 步；否则，转第 (4) 步.

(4) 若存在 $u\in S$，$v\in\bar{S}$，使得 $(v,u)\in E(D)$ 且边 (v,u) 正载，即 $f((u,v))>0$（(v,u) 是反向边），则令 $S\leftarrow S\cup\{v\}$，$prev(v)=u$，转第 (2) 步；否则，输出无可增载轨道，算法停止.

引理 9.2 给定网络 $N=(D,s,t,c)$，流函数 f. 若 N 中存在关于 f 的可增载轨道，则算法 9.1 一定能够找到一条可增载轨道.

证明 用反证法. 设算法 9.1 结束时没有找到可增载轨道，但 N 中存在关于 f 的可增载轨道 $P(s,t)$. 考虑算法结束时得到的顶点子集 S. 因为算法没有找到可增载轨道，故有 $s\in S$，$t\in\overline{S}$. 从而在 $P(s,t)$ 上存在相继的两个顶点 u、v 使得 $u\in S$，$v\in\overline{S}$. 若 $(u,v)\in E(D)$，(u,v) 是 $P(s,t)$ 上的正向边. 因为 $P(s,t)$ 是可增载轨道，所以 $f((u,v))<c((u,v))$，由算法第 (3) 步知，应该 $v\in S$，矛盾. 类似，若 $(v,u)\in E(D)$，(v,u) 是 $P(s,t)$ 上的反向边，也可以得到 $v\in S$，矛盾. 所以若 N 中存在关于 f 的可增载轨道，则算法一定能够找到一条可增载轨道. 证毕.

算法 9.2 Ford-Fulkerson 最大流算法

输入： 网络 $N = (D, s, t, c)$.

输出： 最大流函数 f.

(1) 取初始流函数 f. 比如说可以取 $f(e) \equiv 0$.

(2) 调用可增载轨道算法. 若找到可增载轨道 $P(s,t)$，则构造新的流函数 $\bar{f}$ 如下：

$$\bar{f}(e) = \begin{cases} f(e) + l(P), & e\text{是正向边} \\ f(e) - l(P), & e\text{是反向边} \\ f(e), & e\text{其它} \end{cases}$$

令 $f \leftarrow \bar{f}$，转第 (2) 步. 否则，没有找到可增载轨道，输出 f 是最大流. 停止.

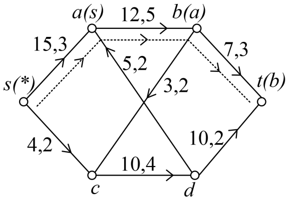
定理 9.2 给定网络 $N=(D,s,t,c)$，Ford-Fulkerson 算法得到的流函数 f 一定是最大流.

证明 Ford-Fulkerson 算法结束时，算法在网络 N 中没有找到关于 f 的可增载轨道. 设 2F 算法结束时，得到节点顶点子集合为 S，其中 $s\in S$，$t\in\overline{S}$. 由可增载轨道算法的第 (3) 步知，任给 $(u,v)\in(S,\overline{S})$，$f((u,v))=c((u,v))$；而由可增载轨道算法的第 (4) 步知，任给 $(u,v)\in(\overline{S},S)$，$f((u,v))=0$. 所以 2F 算法结束时，我们有

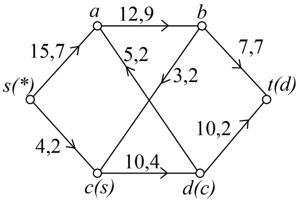
$$\begin{aligned} Val(f) &= \sum_{e\in(S,\overline{S})} f(e) - \sum_{e\in(\overline{S},S)} f(e) \\ &= \sum_{e\in(S,\overline{S})} c(e) - \sum_{e\in(\overline{S},S)} 0 \\ &= C(S,\overline{S}). \end{aligned}$$

由推论 9.2 知，f 是最大流，$(S,\overline{S})$ 是最小截. 证毕.

例 针对图 9.1 中定义的网络与流函数，图 (a) 给出了一个找可增载轨道算法的示例. 在图 (a) 中，首先令 $prev(s) = *$，$S = \{s\}$，由于 $s \in S$ 且 $f((s,a)) = 3 < 15 = c((s,a))$，所以将顶点 a 加入 S，$S = \{s,a\}$；并且记 $prev(a) = s$，说明是从 s 找未满载轨道找到 a 的. 类似地，依次将顶点 b 与 t 加入 S，使得 $S = \{s,a,b,t\}$，且记 $prev(b) = a$、$prev(t) = b$. 由于 $t \in S$，汇 t 被标记，由算法的第（2）步，输出一个可增载轨道 $P_1(s,t) = s - a - b - t$. 而 $l(P_1) = min\{15-3, 12-5, 7-3\} = 4$. 由引理 9.1，我们可以将图 (a) 中的流函数修改为图 (b) 中的流函数，使得流量增加 4.



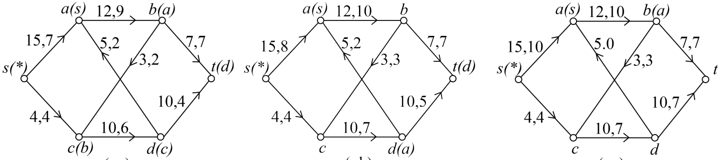
(a)



(b)

基于图 (a) 中得到的可增载轨道，我们可以将流函数更新为图 (b) 所示. 在图 (b) 中，可以得到一条可增载轨道 $P_2(s,t) = s - c - d - t$，我们可以进一步将流函数更新为如图 (c) 所示. 之后我们又得到两条可增载轨道，并且将流函数做进一步更新，参见图 (d) 与图 (e). 最后，将可增载轨道算法应用于图 (e) 所示的流函数中，算法结束时，有 $S = \{s, a, b\}$. 由于 $t \notin S$，算法无法得到可增载轨道.

在图 (e) 中，我们得到 $S = \{s, a, b\}$，$\bar{S} = \{c, d, t\}$，$(S, \bar{S}) = \{(s, c), (b, c), (b, t)\}$，$(\bar{S}, S) = \{(d, a)\}$.$(S, \bar{S})$ 中的边都是满载，$(\bar{S}, S)$ 中的边都是零载. 所以，$Val(f) = C(S, \bar{S})$，f 是最大流，$(S, \bar{S})$ 是最小截.



(c) (d) (e)

推论 9.3(最大流最小截定理) 在网络中，最大流的流量 = 最小截的截量.

定义 9.6 一个容量有上下界的网络可以定义为一个五元组 $N=(D,s,t,b,c)$，其中:

(1) D 是一个弱连通的有向图.

(2) $s\in V(D)$，$t\in V(D)$，分别称为源与汇.

(3) $b,c: E(D)\to \mathbf{R}$ 分别为容量上、下界函数. 任给 $e\in E(D)$，$c(e)\geq b(e)\geq 0$ 为边 e 的容量上界与容量下界.

定义 9.7 网络 $N=(D,s,t,b,c)$ 上的流函数定义为 $f\colon E(D)\to\mathbf{R}$，要求满足：

(1) 任给 $e\in E(D)$，都有 $c(e)\geq f(e)\geq b(e)$，

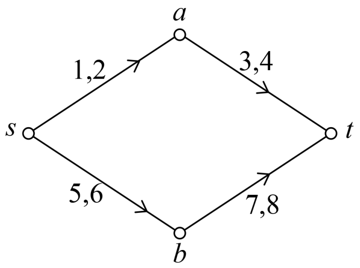
(2) 任给 $v\in V(D)-\{s,t\}$，都有 $\sum_{e\in\alpha(v)}f(e)-\sum_{e\in\beta(v)}f(e)=0$.

其中，$\alpha(v)$ 是所有以 v 为头的边集，而 $\beta(v)$ 则是所有以 v 为尾的边集.f 的流量定义为

$$\begin{aligned}Val(f)&=\sum_{e\in\alpha(t)}f(e)-\sum_{e\in\beta(t)}f(e)\\&=\sum_{e\in\beta(s)}f(e)-\sum_{e\in\alpha(s)}f(e).\end{aligned}$$

对于一般的网络，流函数总是存在的，比如说可以取流函数 $f(e)=0$. 而容量有上下界的网络不一定存在流函数，我们称容量有上下界网络的流函数为**可行流**.

例 9.5 如图为一个非常简单的网络，其中每条边上有两个数值，分别为该边容量的下界与上界. 这个网络就不存在流函数. 比如说，(s, v_1) 是唯一一条以 v_1 为头的有向边，而 (v_1, t) 则是唯一一条以 v_1 为尾的有向边. 由于 $c((s, v_1)) < b((v_1, t))$，所以不可能存在流函数 f，满足 $\sum_{e\in\alpha(v_1)} f(e) - \sum_{e\in\beta(v_1)} f(e) = 0$. 因而这个网络不存在可行流.



定义 9.8 给定容量有上下界网络 $N=(D,s,t,b,c)$，定义 N 的伴随网络为一般的网络 $N'=(D',s',t',c')$，其中:

(1) $V(D')=V(D)\cup\{s',t'\}$，其中，$s',t'\notin V(D)$，

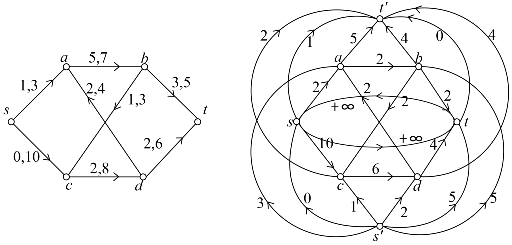
(2) $E(D')=E(D)\cup\{(s',v),(v,t')|v\in V(D)\}\cup\{(s,t),(t,s)\}$，

(3) s' 与 t' 分别为伴随网络 N' 的源与汇，

(4) 容量函数 c' 定义为:

$$c'(e)=\begin{cases} c(e)-b(e), & e\in E(D) \\ \sum_{e\in\alpha(v)} b(e), & e=(s',v), v\in V(D) \\ \sum_{e\in\beta(v)} b(e), & e=(v,t'), v\in V(D) \\ +\infty, & e=(s,t)\text{或}(t,s). \end{cases}$$

例 9.6 图 (a) 为一个容量有上下界网络，而图 (b) 则为其伴随网络.



(a) (b)

在伴随网络中，引入了虚拟的源 s' 与汇 t'. 对应于容量有上下界网络 N 的每条边 e，在伴随网路 N' 中也有一条边 e，该边在伴随网络中的容量函数定义为 e 在 N 中容量的上、下界之差，即 $c'(e) = c(e) - b(e)$. 而对应于 N 的每个顶点 v，在伴随网络中定义了一条边 (s', v)，其容量函数 $c'((s', v))$ 则定义为 $\alpha(v)$ 中所有的边的容量下界之和.

中国科学技术大学 University of Science and Technology of China

定理 9.3 给定网络 $N=(D,s,t,b,c)$，其伴随网络为 $N'=(D',s',t',c')$，则 N 中存在可行流，当且仅当 N' 中最大流使得任给 $v\in V(D)$，边 (s',v) 都满载，即若 N' 的最大流为 f'，有 $f'((s',v))=c'((s',v))$.

证明 设 N' 的最大流为 f'，f' 使得任给 $v\in V(D)$，边 (s',v) 都满载，即 $f'((s',v))=c'((s',v))$. 因为 $E(D)\subset E(D')$，我们可以根据 f' 定义 N 上的边权函数 $f:E(D)\to\mathbf{R}$，

$$f(e)=f'(e)+b(e),e\in E(D).$$

下面我们来证明 f 是 N 上的一个可行流.
首先，因为 f' 是 N' 上的流函数，所以任给 $e\in E(D)\subset E(D')$，都有

$$0\le f'(e)\le c'(e)=c(e)-b(e).$$

因此有 $$b(e)\le f(e)=f'(e)+b(e)\le c'(e)+b(e)=c(e).$$
这样边权函数 f 满足容量有上下界网络流函数定义中的条件 (1).

下面我们来证明 f 满足容量有上下界网络流函数定义中的条件 (2).
任给 $v \in V(D)-\{s,t\}$，由于 f' 是 N' 上的流函数，f' 在 D' 中的顶点 v 处流量均衡，即

$$\sum_{e\in\alpha_{D'}(v)} f'(e) - \sum_{e\in\beta_{D'}(v)} f'(e) = 0. \qquad (9.8)$$

其中，$\alpha_{D'}(v)$、$\beta_{D'}(v)$ 分别表示在 D' 中以 v 为头的边集合与以 v 为尾的边集合. 在下面的公式中，我们分别用 $\alpha_D(v)$、$\beta_D(v)$ 表示在 D 中以 v 为头的边集合与以 v 为尾的边集合. 由于 $\alpha_{D'}(v)-\alpha_D(v)=\{(s',v)\}$，$\beta_{D'}(v)-\beta_D(v)=\{(v,t')\}$，所以有

$$\sum_{e\in\alpha_{D'}(v)} f'(e) = \sum_{e\in\alpha_D(v)} f'(e) + f'((s',v)),$$

$$\sum_{e\in\beta_{D'}(v)} f'(e) = \sum_{e\in\beta_D(v)} f'(e) + f'((v,t')).$$

因为 f' 使得任给 $v \in V(D)$，边 (s', v) 都满载，即 $f'((s', v)) = c'((s', v))$ $= \sum_{e\in\alpha_D(v)} b(e)$，我们同样有 $f'((v, t')) = c'((v, t')) = \sum_{e\in\beta_D(v)} b(e)$. 因此

$$\sum_{e\in\alpha_{D'}(v)} f'(e) = \sum_{e\in\alpha_D(v)} f'(e) + \sum_{e\in\alpha_D(v)} b(e) = \sum_{e\in\alpha_D(v)} [f'(e) + b(e)] = \sum_{e\in\alpha_D(v)} f(e), \tag{9.9}$$

$$\sum_{e\in\beta_{D'}(v)} f'(e) = \sum_{e\in\beta_D(v)} f'(e) + \sum_{e\in\beta_D(v)} b(e) = \sum_{e\in\beta_D(v)} [f'(e) + b(e)] = \sum_{e\in\beta_D(v)} f(e). \tag{9.10}$$

综合公式 9.8、9.9 与 9.10，我们有

$$\sum_{e\in\alpha_D(v)} f(e) - \sum_{e\in\beta_D(v)} f(e) = \sum_{e\in\alpha_{D'}(v)} f'(e) - \sum_{e\in\beta_{D'}(v)} f'(e) = 0.$$

f 满足容量有上下界网络 N 流函数定义的第二个条件. 综上，f 是容量有上下界网络 N 的一个可行流.

反之，若 N 存在可行流，设为 f. 我们构造 N' 上的边权函数如下：

$$f'(e)=\begin{cases} f(e)-b(e), & e\in E(D) \\ c'(e), & \text{存在}v\in V(D)\text{，使得}e=(s',v)\text{或}(v,t'). \end{cases}$$

将上面的证明过程逆过来，类似可以证明，f' 是 N' 的最大流，且使得任给 $v\in V(D)$，f' 使得边 (s',v) 满载，即 $f'((s',v))=c'((s',v))$. 证毕.

算法 9.3 容量有上下界网路的最大流算法

输入：容量有上下界网路 $N=(D,s,t,b,c)$.

输出：最大流函数 f，或断定 N 没有可行流.

(1) 构造 N 的伴随网络 $N'=(D',s',t',c')$.

(2) 用 2F 算法求出 N' 的最大流函数 f'.

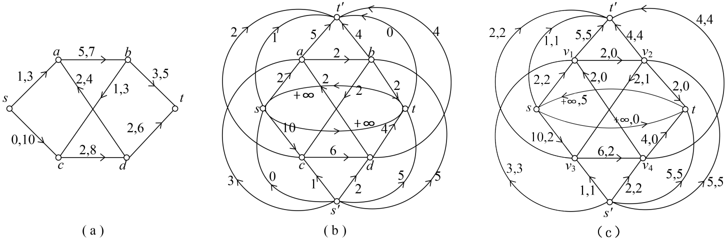
(3) 若 f' 满足，任给 $v\in V(D)$，f' 使得边 (s',v) 满载，即 $f'((s',v))=c'((s',v))$，则转第 (4) 步；否则，输出结论 “N 没有可行流”，算法停止.

(4) 根据 f'，构造 N 的一个可行流 f: 任给 $e\in E(D)$，

$$f(e)=f'(e)+b(e).$$

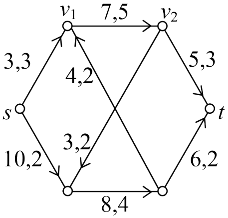
(5) 以 f 作为初始流函数，用 2F 算法求出 N 的最大流，算法停止.

例 9.7 求图 (a) 中网络的最大流.

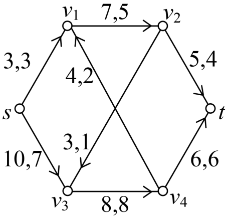


解 按照算法 9.3，我们先求出 N 的伴随网络 N'，参见图 (b). 然后求出 N' 的最大流 f'，参见图 (c)，其中每条边上的第一个参数为伴随网路的容量函数 c'，第二个参数为流函数 f'.

例题



(d)



(e)

由于任给 $v \in V(D)$，f' 使得边 (s', v) 满载，N 有可行流. 我们根据 f'，对 D 中的每条边 e，令 $f(e) = f'(e) + b(e)$，求出 N 的一个可行流 f，参见图 (d)，其中每条边上的两个参数分别为容量有上下界网络容量的上界和流函数值. 最后在 f 的基础上，用 2F 算法求出 N 的最大流 f^*，参见图 (e)，f^* 的流量为

$$Val(f^*) = \sum_{e \in \alpha_D(t)} f^*(e) - \sum_{e \in \beta_D(t)} f^*(e) = 6 + 4 = 10.$$

定义 9.9 一个有供需约束的网络可以定义为一个六元组 $N=(D,X,Y,\sigma,\rho,c)$，其中:

(1) D 是一个弱连通的有向图.

(2) $X=\{x_1,x_2,\ldots,x_m\}\subseteq V(D)$ 是源集合，每个 $x_i(1\le i\le m)$ 表示一个产地.

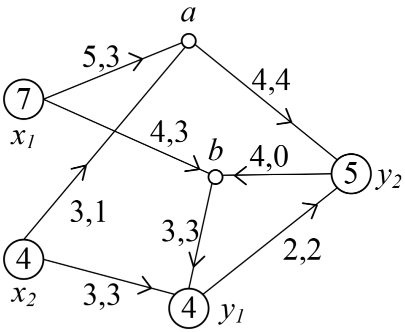
(3) $Y=\{y_1,y_2,\ldots,y_n\}\subseteq V(D)$ 是汇集合，每个 $y_j(1\le j\le n)$ 表示一个消费市场.

(4) $\sigma:X\to\mathbf{R}$，$\sigma(x_i)$ 表示产地 x_i 的产量，$1\le i\le m$.

(5) $\rho:Y\to\mathbf{R}$，$\rho(y_j)$ 表示消费市场 y_j 的需求量，$1\le j\le n$.

(6) $c:E(D)\to\mathbf{R}$ 为容量函数. 任给 $e\in E(D)$，$c(e)$ 为边 e 的容量.

例 9.8 如图是一个有供需约束的网络. 其中，每条边上的第一个参数为该边的容量，而第二个参数则是下面定义的流函数在该边上流量.x_i 的参数即为 $\sigma(x_i)(1 \le i \le 2)$，$y_i$ 的参数即为 $\rho(y_i)(1 \le j \le 2)$.



定义 9.10 给定有供需约束的网络 $N=(D,X,Y,\sigma,\rho,c)$，N 上的流函数 $f: E(D)\to \mathbf{R}$，要求满足：

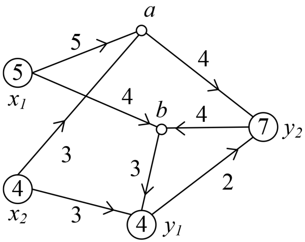
(1) 任给 $e\in E(D)$，都有 $0\le f(e)\le c(e)$，

(2) 任给 $v\in V(D)-X\cup Y$，都有 $\sum_{e\in\alpha(v)} f(e)-\sum_{e\in\beta(v)} f(e)=0$.

(3) 任给 $1\le i\le m$，都有 $\sum_{e\in\beta(x_i)} f(e)-\sum_{e\in\alpha(x_i)} f(e)\le \sigma(x_i)$，表示从顶点 x_i 处实际运出量不能超过其产量.

(4) 任给 $1\le j\le n$，都有 $\sum_{e\in\alpha(y_j)} f(e)-\sum_{e\in\beta(y_j)} f(e)\ge \rho(y_j)$，表示实际运入顶点 y_j 的总量大于等于其消费需求.

例 满足定义的流函数 f 称为有供需约束的可行流. 有供需约束的网络不一定有可行流，比如说，若所有产地的总产量小于所有消费市场的需求量，即 $\sum_{1\le i\le m}\sigma(x_i)<\sum_{1\le j\le n}\rho(y_j)$, 则一定不存在可行流, 如图所示



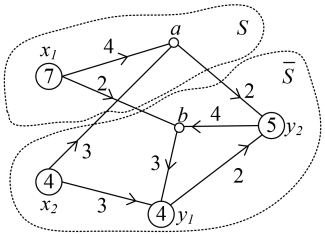
(a)

定理 9.4 给定有供需约束的网络 $N=(D,X,Y,\sigma,\rho,c)$.N 有可行流的充要条件是：任给 $S\subseteq V(D)$，都满足

$$C((S,\overline{S}))\geq \rho(Y\cap \overline{S})-\sigma(X\cap \overline{S}), \tag{9.7}$$

其中，$C((S,\overline{S}))=\sum_{e\in(S,\overline{S})}c(e)$ 为截 $(S,\overline{S})$ 的截量，$\rho(Y\cap \overline{S})=\sum_{y_j\in Y\cap\overline{S}}\rho(y_j)$，$\sigma(X\cap \overline{S})=\sum_{x_i\in X\cap\overline{S}}\sigma(x_i)$.

例 9.9 例如，在图 (b) 中，若我们取 $S=\{x_1, a\}$，
则 $\overline{S}=\{x_2, y_1, y_2, b\}$，$Y\cap\overline{S}=\{y_1, y_2\}$，
$X\cap\overline{S}=\{x_2\}.\rho(Y\cap\overline{S})-\sigma(X\cap\overline{S})=\rho(y_1)+\rho(y_2)-\sigma(x_2)=(4+5)-4=5$，
而 $(S,\overline{S})=\{(x_1,b),(a,y_2)\}$，$C((S,\overline{S}))=c((x_1,b))+((a,y_2))=2+2=4$.
所以，图 (b) 对应的有供需约束的网络没有可行流.



(b)

中国科学技术大学 University of Science and Technology of China

为了证明定理 9.4 的正确性，我们先介绍附加网络及其性质，后面再利用附加网络的性质来证明定理 9.4.

定义 9.11 给定有供需约束的网络 $N=(D,X,Y,\sigma,\rho,c)$，定义 N 的**附加网络** $N'=(D',x_0,y_0,c')$ 为:

(1) $V(D')=V(D)\cup\{x_0,y_0\}$，其中 $x_0,y_0\notin V(D)$.

(2) $E(D')=E(D)\cup\{(x_0,x_i)|i=1,\ldots,m\}\cup\{(y_j,y_0)|j=1,\ldots,n\}$.

(3) x_0 与 y_0 分别为 N' 的源与汇.

(4) 容量函数 c' 定义为:

$$c'(e)=\begin{cases} c(e), & e\in E(D),\\ \sigma(x_i), & e=(x_0,x_i),\\ \rho(y_j), & e=(y_j,y_0).\end{cases}$$

引理 9.3 有供需约束的网络 $N = (D, X, Y, \sigma, \rho, c)$ 存在可行流，当且仅当其附加网络 $N' = (D', x_0, y_0, c')$ 的最大流 f' 满足：任给 $1 \leq j \leq n$，$f'((y_j, y_0)) = c'((y_j, y_0))$.

证明 由附加网络 N' 的定义可知，

$$(V(D') - \{y_0\}, \{y_0\}) = \{(y_j, y_0)|1 \leq j \leq n\}$$

是 N' 的一个截. 若 N' 的流函数 f' 满足：任给 $1 \leq j \leq n$，$f'((y_j, y_0)) = c'((y_j, y_0))$，则

$$Val(f') = \sum_{1 \leq j \leq n} f'((y_i, y_0)) = \sum_{1 \leq j \leq n} c'((y_j, y_0)) = C((V(D') - \{y_0\}, \{y_0\})).$$

由推论 9.2 知，f' 是 N' 的最大流.

将附加网络 N' 上的流函数 f' 限定到网络 N 上，得到 N 上的一个边权函数 f，即任给 $e \in E(D) \subset E(D')$，

$$f(e) = f'(e).$$

由附加网络的定义，任给 $v \in V(D) - X - Y$，都有 $\alpha_D(v) = \alpha_{D'}(v)$，$\beta_D(v) = \beta_{D'}(v)$，所以有

$$\sum_{e \in \alpha_D(v)} f(e) - \sum_{e \in \beta_D(v)} f(e) = \sum_{e \in \alpha_{D'}(v)} f'(e) - \sum_{e \in \beta_{D'}(v)} f'(e) = 0.$$

所以，f 满足有供需约束网路流函数的第 (2) 个条件.

因为 f' 是 N' 的流函数，所以任给 $1 \leq i \leq m$，

$$\sum_{e \in \alpha_{D'}(x_i)} f'(e) - \sum_{e \in \beta_{D'}(x_i)} f'(e) = 0.$$

按照附加网络的定义，$\alpha_D(x_i) = \alpha_{D'}(x_i) - \{(x_0, x_i)\}$，$\beta_D(x_i) = \beta_{D'}(x_i)$. 所以有

$$\begin{aligned}
\sum_{e \in \beta_D(x_i)} f(e) - \sum_{e \in \alpha_D(x_i)} f(e) &= \sum_{e \in \beta_D(x_i)} f'(e) - \sum_{e \in \alpha_D(x_i)} f'(e) \\
&= \sum_{e \in \beta_{D'}(x_i)} f'(e) - [\sum_{e \in \alpha_{D'}(x_i)} f'(e) - f'((x_0, x_i))] \\
&= f'((x_0, x_i)) \leq c'((x_0, x_i)) = \sigma(x_i).
\end{aligned}$$

所以，f 满足有供需约束网路流函数的第 (3) 个条件.

同理，按照附加网络的定义，$\alpha_D(y_j)=\alpha_{D'}(y_j)$，$\beta_D(y_j)=\beta_{D'}(y_j)-\{(y_j,y_0)\}$. 若 f' 满足：任给 $1\le j\le n$，$f'((y_j,y_0))=c'((y_j,y_0))$，则有

$$\begin{aligned}\sum_{e\in\alpha_D(y_j)} f(e)-\sum_{e\in\beta_D(y_j)} f(e) &= \sum_{e\in\alpha_D(y_j)} f'(e)-\sum_{e\in\beta_D(y_j)} f'(e)\\ &= \sum_{e\in\alpha_{D'}(y_j)} f'(e)-[\sum_{e\in\beta_{D'}(y_j)} f'(e)-f'((y_j,y_0))]\\ &= f'((y_j,y_0))=c'((y_j,y_0))=\rho(y_j).\end{aligned}$$

所以，f 满足有供需约束网路流函数的第 (4) 个条件. 至于第 (1) 个条件，f 继承了 f' 的性质，自然满足，所以 f 是 N 上的一个可行流.

反之，若 N 上存在可行流 f，则可以证明其附加网络 N' 上一定存在流函数 f'，使得任给 $1\le j\le n$，边 (y_j,y_0) 都满载. 留作习题 9. 证毕.

定理 9.4 的证明：由引理 9.3 知，N 有可行流等价于其附加网络 N' 上的最大流 f' 满足：任给 $1 \le j \le n$，$f'((y_j, y_0)) = c'((y_j, y_0))$. 任给 $S \subseteq V(D)$，$(S \cup \{x_0\}, \overline{S} \cup \{y_0\})$ 是 N' 的截. 由引理 9.3 的证明过程可知，$(V(D') - \{y_0\}, \{y_0\})$ 为 N' 的最小截. 由最小截的截量可知，

$$C((S \cup \{x_0\}, \overline{S} \cup \{y_0\})) \ge C((V(D') - \{y_0\}, \{y_0\})) = \sum_{1 \le j \le n} \rho(y_j) = \rho(Y). \tag{9.8}$$

而由附加网络的定义知，

$$(S \cup \{x_0\}, \overline{S} \cup \{y_0\}) = (S, \overline{S}) \cup (Y \cap S, \{y_0\}) \cup (\{x_0\}, X \cap \overline{S}).$$

所以 (9.8) 式等价于

$$C(S, \overline{S}) + \rho(Y \cap S) + \sigma(X \cap \overline{S}) \ge \rho(Y),$$

等价于

$$C(S, \overline{S}) \ge [\rho(Y) - \rho(Y \cap S)] - \sigma(X \cap \overline{S}) = \rho(Y \cap \overline{S}) - \sigma(X \cap \overline{S}).$$

证毕.

算法 9.4 有供需约束网路的可行流算法

输入：有供需约束网路 $N = (D, X, Y, \sigma, \rho, c)$.

输出：N 的可行流函数 f，或断定 N 没有可行流.

(1) 构造 N 的附加网络 $N' = (D', x_0, y_0, c')$.

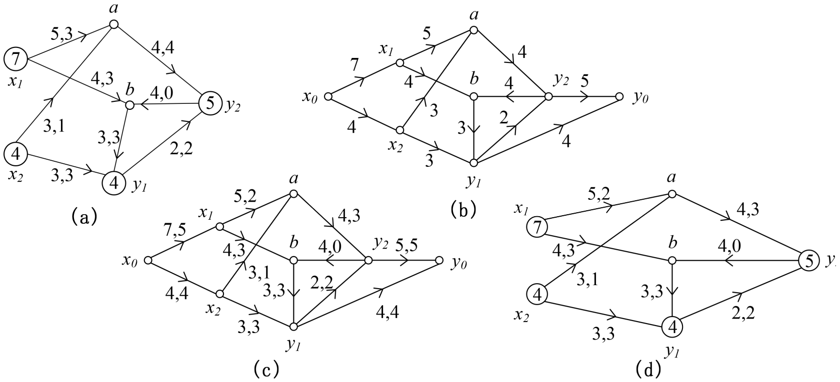
(2) 用 2F 算法求出 N' 的最大流函数 f'.

(3) 若 f' 满足：任给 $1 \le j \le n$，f' 使得边 (y_j, y_0) 满载，即 $f'((y_j, y_0)) = c'((y_j, y_0))$，则转第 (4) 步；否则，输出结论 "$N$ 没有可行流"，算法停止.

(4) 将 f' 限制到网络 N 上. 即任给 $e \in E(D) \subset E(D')$，令 $f(e) = f'(e)$.f 就是 N 的可行流. 算法停止.

例题

例 例如，针对图 (b) 表示的附加网络，我们可以用 2F 算法求出其最大流，如图 (c) 所示. 其中，每条边上的第一个参数表示附加网络中边的容量，而第二个参数则为最大流的函数值. 可以看出，其最大流满足引理 9.3 的条件，所以对应于图 (a) 中的有供需约束网络存在可行流，对应的可行流如图 (d) 所示



(a) (b) (c) (d)

定理 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，则有

$$Val(f)=\sum_{e\in(S,\overline{S})}f(e)-\sum_{e\in(\overline{S},S)}f(e).$$

推论 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，则有

$$Val(f)\leq C(S,\overline{S}).$$

推论 9.2 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$(S,\overline{S})$ 是其一个截，若 $Val(f)=C(S,\overline{S})$，则 f 是最大流，$(S,\overline{S})$ 是最小截.

引理 9.1 设 f 是网络 $N=(D,s,t,c)$ 的流函数，$P(s,t)$ 是 N 上关于 f 的可增载轨道，定义新的函数 $\bar{f}: E(D)\to \mathbf{R}$ 为

$$\bar{f}(e)=\begin{cases} f(e)+l(P), & e\text{是正向边}\\ f(e)-l(P), & e\text{是反向边}\\ f(e), & e\text{其它}\end{cases}$$

则 $\bar{f}$ 是网络 N 的流函数，且 $Val(\bar{f})=Val(f)+l(P)$.

算法 9.1　可增载轨道算法

输入：网络 $N=(D,s,t,c)$，流函数 f.

输出：一条可增载轨道，或指出当前流函数是最大流.

(1) $S=s$；令 $prev(s)=*$.

(2) 若 $t\in S$，则已经找到可增载轨道，通过 $prev(t)$ 回溯输出可增载轨道，算法停止；否则，转第 (3) 步.

(3) 若存在 $u\in S$，$v\in \overline{S}$，使得 $(u,v)\in E(D)$ 且边 (u,v) 未满载，即 $f((u,v))<c((u,v))$ ((u,v) 是正向边)，则令 $S\leftarrow S\cup\{v\}$，$prev(v)=u$，转第 (2) 步；否则，转第 (4) 步.

(4) 若存在 $u\in S$，$v\in \overline{S}$，使得 $(v,u)\in E(D)$ 且边 (v,u) 正载，即 $f((u,v))>0$ ((v,u) 是反向边)，则令 $S\leftarrow S\cup\{v\}$，$prev(v)=u$，转第 (2) 步；否则，输出无可增载轨道，算法停止.

引理 9.2　给定网络 $N=(D,s,t,c)$，流函数 f. 若 N 中存在关于 f 的可增载轨道，则算法 9.1 一定能够找到一条可增载轨道.

算法 9.2 Ford-Fulkerson 最大流算法

输入：网络 $N=(D,s,t,c)$.

输出：最大流函数 f.

(1) 取初始流函数 f. 比如说可以取 $f(e)\equiv 0$.

(2) 调用可增载轨道算法. 若找到可增载轨道 $P(s,t)$，则构造新的流函数 $\bar{f}$ 如下：

$$\bar{f}(e)=\begin{cases} f(e)+l(P), & e\text{是正向边} \\ f(e)-l(P), & e\text{是反向边} \\ f(e), & e\text{其它} \end{cases}$$

令 $f\leftarrow \bar{f}$，转第 (2) 步. 否则，没有找到可增载轨道，输出 f 是最大流. 停止.

定理 9.2 给定网络 $N=(D,s,t,c)$，2F 算法得到的流函数 f 一定是最大流.

推论 9.3(最大流最小截定理) 在网络中，最大流的流量 $=$ 最小截的截量.

定理 9.3 给定网络 $N=(D,s,t,b,c)$，其伴随网络为 $N'=(D',s',t',c')$，则 N 中存在可行流，当且仅当 N' 中最大流使得任给 $v\in V(D)$，边 (s',v) 都满载，即若 N' 的最大流为 f'，有 $f'((s',v))=c'((s',v))$.

算法 9.3 容量有上下界网路的最大流算法
输入：容量有上下界网路 $N=(D,s,t,b,c)$.
输出：最大流函数 f，或断定 N 没有可行流.

(1) 构造 N 的伴随网络 $N'=(D',s',t',c')$.

(2) 用 2F 算法求出 N' 的最大流函数 f'.

(3) 若 f' 满足，任给 $v\in V(D)$，f' 使得边 (s',v) 满载，即 $f'((s',v))=c'((s',v))$，则转第 (4) 步；否则，输出结论 “N 没有可行流”，算法停止.

(4) 根据 f'，构造 N 的一个可行流 f: 任给 $e\in E(D)$，
$$f(e)=f'(e)+b(e).$$

(5) 以 f 作为初始流函数，用 2F 算法求出 N 的最大流，算法停止.

定理 9.4 给定有供需约束的网络 $N=(D,X,Y,\sigma,\rho,c)$.N 有可行流的充要条件是：任给 $S\subseteq V(D)$，都满足

$$C((S,\overline{S}))\geq \rho(Y\cap\overline{S})-\sigma(X\cap\overline{S}), \qquad (9.7)$$

其中，$C((S,\overline{S}))=\sum_{e\in(S,\overline{S})}c(e)$ 为截 $(S,\overline{S})$ 的截量，$\rho(Y\cap\overline{S})=\sum_{y_j\in Y\cap\overline{S}}\rho(y_j)$，$\sigma(X\cap\overline{S})=\sum_{x_i\in X\cap\overline{S}}\sigma(x_i)$.

引理 9.3 有供需约束的网络 $N=(D,X,Y,\sigma,\rho,c)$ 存在可行流，当且仅当其附加网络 $N'=(D',x_0,y_0,c')$ 的最大流 f' 满足：任给 $1\leq j\leq n$，$f'((y_j,y_0))=c'((y_j,y_0))$.

算法 9.4 有供需约束网路的可行流算法
输入：有供需约束网路 $N=(D,X,Y,\sigma,\rho,c)$.
输出：N 的可行流函数 f，或断定 N 没有可行流.

(1) 构造 N 的附加网络 $N'=(D',x_0,y_0,c')$.

(2) 用 2F 算法求出 N' 的最大流函数 f'.

(3) 若 f' 满足：任给 $1\le j\le n$，f' 使得边 (y_j,y_0) 满载，即 $f'((y_j,y_0))=c'((y_j,y_0))$，则转第 (4) 步；否则，输出结论 "$N$ 没有可行流"，算法停止.

(4) 将 f' 限制到网络 N 上. 即任给 $e\in E(D)\subset E(D')$，令 $f(e)=f'(e)$.f 就是 N 的可行流. 算法停止.