第6章 图与网络分析

6.6 最大流问题

最大流（Max Flow）问题

- 实例：给定有向网络 $G = (V, E)$， 弧 $e = (i, j) \in E$ 上定义有容量$c_e \geq 0$。有两个特殊的顶点 $s, t \in V$，其中 s 是发点，t 是收点。
- 目标：计算从 s 到 t 的满足容量约束的最大流。

符号：

- 定义x_{ij}为弧(i, j)上从顶点i到顶点j的流量。
- 若流量$\{x_{ij}\}$在所有弧上均满足容量约束，在除s和t之外的所有顶点上都满足流守恒约束，则其为一个流，记为x。
- 流x的流值定义为s点发出的流量之和与s点接收的流量之和的差，即s点发出的“净流”流量。

基本概念

- 可行流：满足容量约束和流守恒约束的从 s 到 t 的流，简称为 (s, t)-流。
- 设 P 是 G 上从 s 到 t 的一条“路”，路上弧的方向可以任意。则：P 上方向是从 s 到 t 的弧 (i, j) 称为前向弧。
  P 上方向是从 t 到 s 的弧 (i, j) 称为后向弧。
- 流 $x = (x_{ij})$的增广路 P：P 的每个前向弧 (i, j) 都有 $x_{ij} < c_{ij}$，而 P 的每个后向弧有 $x_{ij} > 0$。
- (s, t)-割：记为(S, T)，是从 S 到 T 的所有弧（有向边）的集合。其中 (S, T) 是顶点集 V 的一个划分，$s \in S$，$t \in T$。
- 割(S, T)的容量：$c(S,T) = \sum_{i\in S}\sum_{j\in T} c_{ij}$，即割中所有弧的容量的和。

Ford-Fulkerson最大流算法

- 基本思想：不断地找增广路来增加流，当流值不能再增加时，就得到了最大流。
- 找增广路的方法：标号的方法。
- 源顶点 s 具有永久标号 $(-,\infty)$ 。
- 假设当前已标号点为 i，从 i 经过弧 (i, j) 或 (j, i) 对未标号点 j 进行标号。
- 若存在弧 (i, j) 且 $x_{ij} < c_{ij}$，则给 j 标号 $(+i,\delta(j))$，其中 $\delta(j) = \min\{c_{ij} - x_{ij}, \delta(i)\}$ 。
- 若存在弧 (j, i) 且 $x_{ji} > 0$，则给 j 标号 $(-i,\delta(j))$，其中 $\delta(j) = \min\{x_{ji}, \delta(i)\}$ 。

标号的解释

- 一般地，顶点 j 的标号为 $(\pm i, \delta(j))$。其中，
- i 表示顶点 j 是从 i 获得的标号，即在从 s 到 j 的增广路上，i 是 j 的前驱顶点。
- i 之前的符号：正号“+”表示(i, j)是前向弧，在弧(i, j)上可以增加流。
- 负号“–”表示(j, i)是后向弧，在弧(j, i)上可以减少流。
- $\delta(j)$表示可以增加或减少的流量的大小。

$i \xrightarrow{x_{ij} < c_{ij}} j$ $\qquad$ $i \xleftarrow{x_{ji} > 0} j$

$(\pm k, \delta(i)) \quad (+i, \delta(j))$ $\qquad$ $(\pm k, \delta(i)) \quad (-i, \delta(j))$

$\delta(j) = \min\{c_{ij} - x_{ij}, \delta(i)\}$ $\qquad$ $\delta(j) = \min\{x_{ji}, \delta(i)\}$

最大流算法[Ford & Fulkerson, 1957]

1 $\forall (i,j) \in E$， $x_{ij} \leftarrow 0$，得到一个零流。

2 给 s 一个永久标号$(-,\infty)$，$LIST \leftarrow \{s\}$。

（*LIST* 是已标号但未检查的顶点的集合。）

3 while $LIST \neq \varnothing$

4 从 *LIST* 中取出一个顶点 i，并将 i 从 *LIST* 中删除。

5 对每一个弧(i,j)，如果$x_{ij} < c_{ij}$且 j 未标号，则对 j 标号 $(+i,\delta(j))$，其中$\delta(j) = \min\{c_{ij} - x_{ij}, \delta(i)\}$。

6 对每一个弧(j, i)，如果$x_{ji} > 0$且 j 未标号，则对 j 标号 $(-i,\delta(j))$，其中$\delta(j) = \min\{x_{ji}, \delta(i)\}$。

7 将新标号的点加入 *LIST*。

Ford-Fulkerson最大流算法

if t 被标号 **then**

由点 t 开始回溯到 s 找到一条 st-增广路，在这条路上增加流量δ(t)。（增广。）

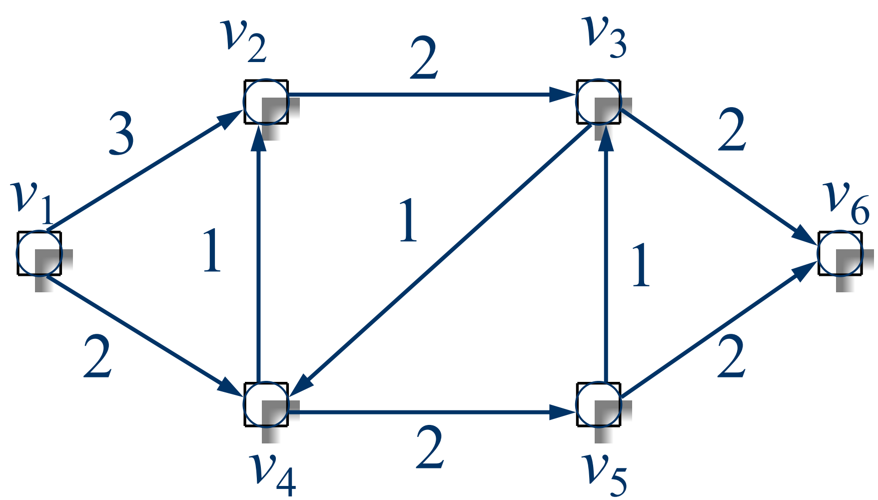
清除除 s 外所有顶点的标号，$LIST \leftarrow \{s\}$。

endif

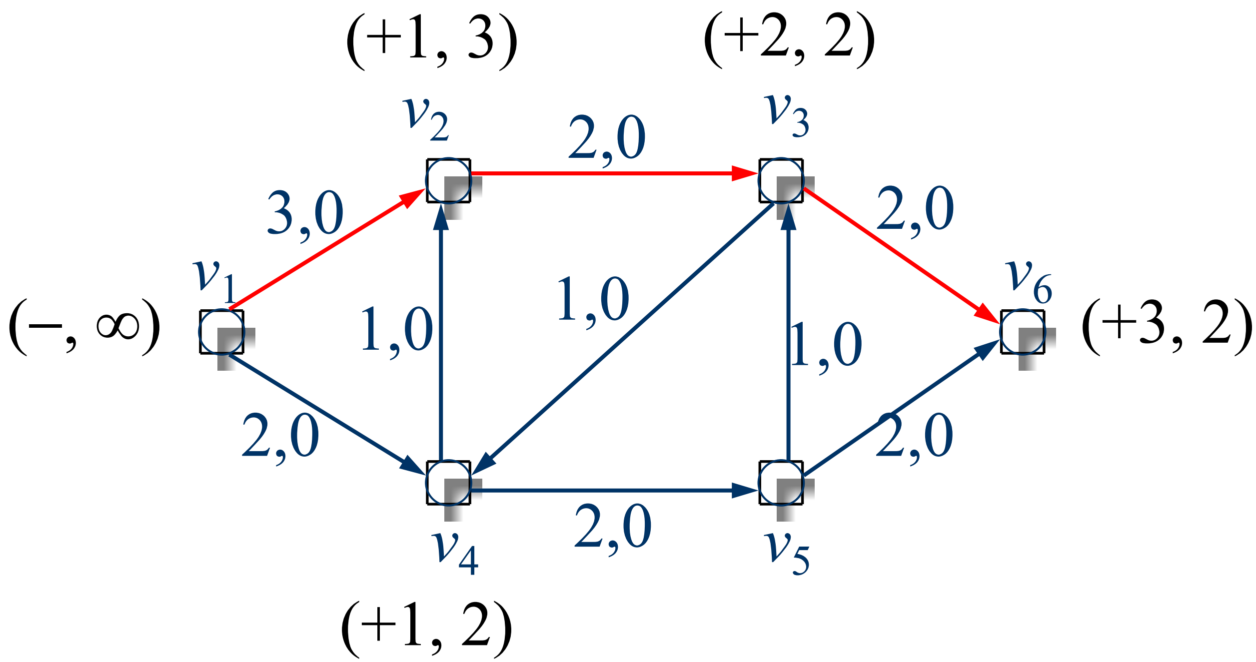
endwhile

return $\{x_{ij}\}$。（当前的流$\{x_{ij}\}$即为最大流，当前已标号的顶点的集合形成一个最小割。）

例1



例1



$$i \xrightarrow{x_{ij} < c_{ij}} j$$

$$(\pm k, \delta(i)) \qquad (+i, \delta(j))$$

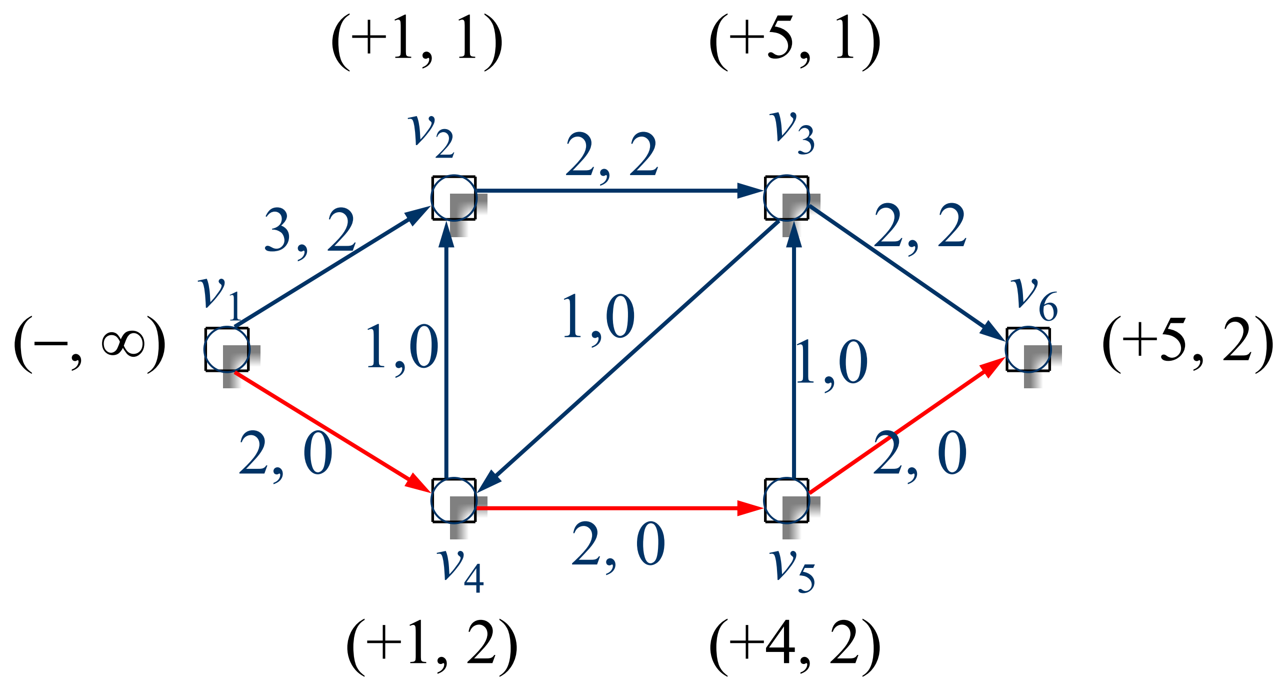
$$\delta(j) = \min\{c_{ij} - x_{ij}, \delta(i)\}$$

$$i \xleftarrow{x_{ji} > 0} j$$

$$(\pm k, \delta(i)) \qquad (-i, \delta(j))$$

$$\delta(j) = \min\{x_{ji}, \delta(i)\}$$

例1

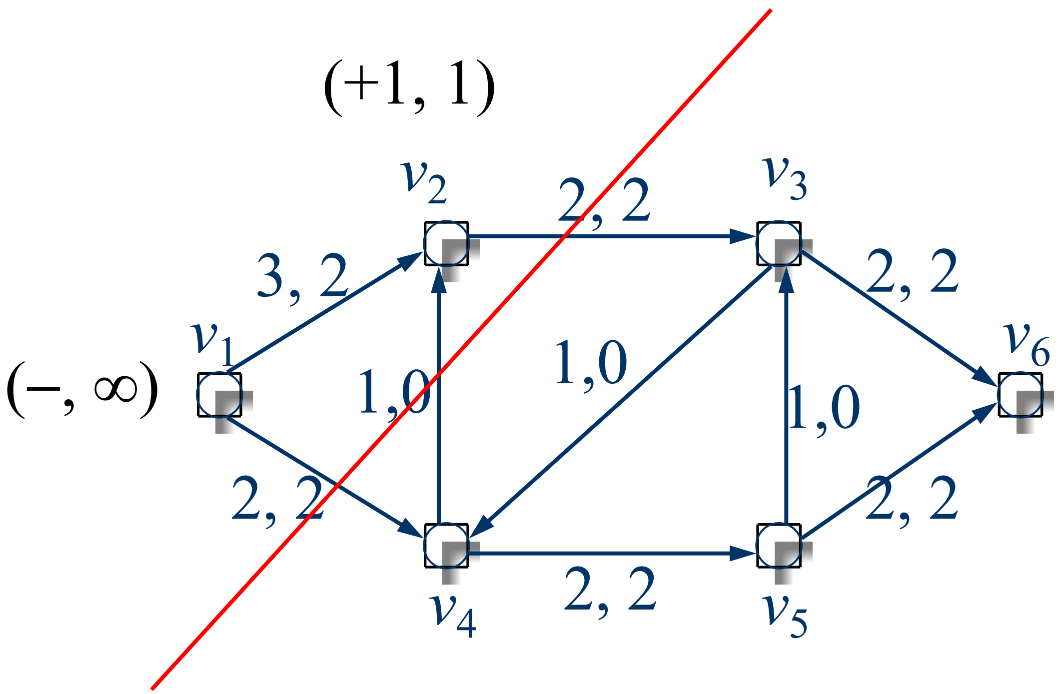


$$i \xrightarrow{x_{ij} < c_{ij}} j \qquad\qquad i \xleftarrow{x_{ji} > 0} j$$

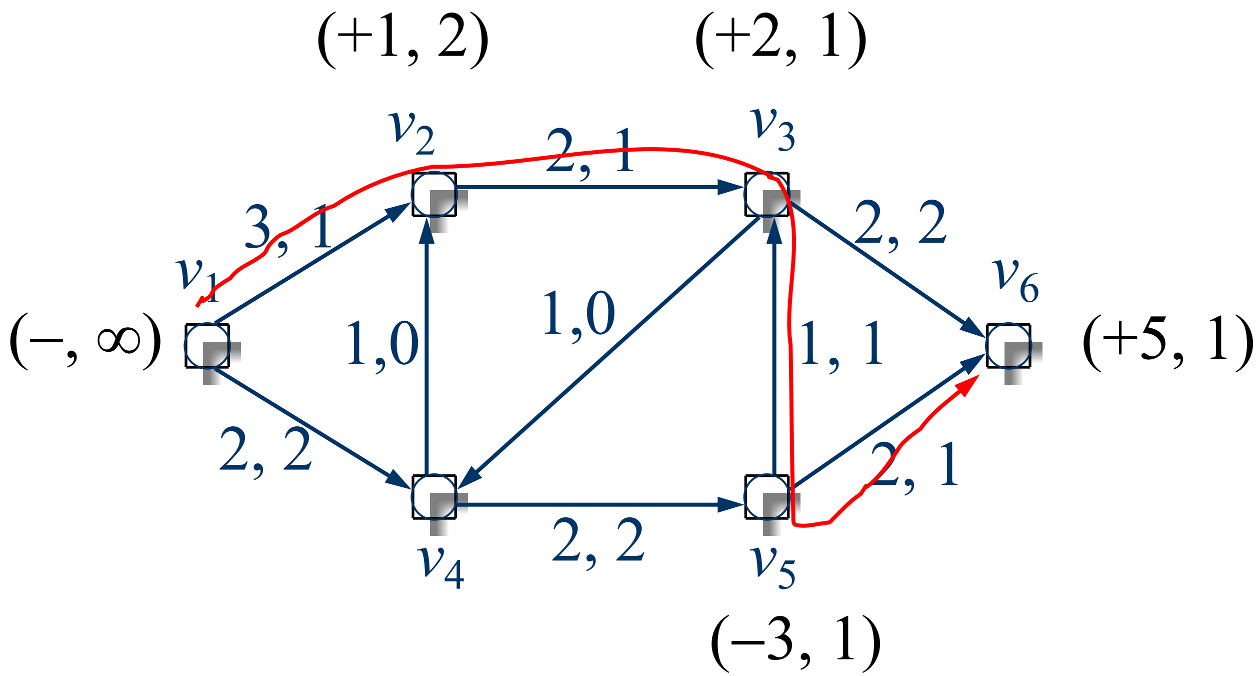
$$(\pm k, \delta(i)) \quad (+i, \delta(j)) \qquad (\pm k, \delta(i)) \quad (-i, \delta(j))$$

$$\delta(j) = \min\{c_{ij} - x_{ij}, \delta(i)\} \qquad \delta(j) = \min\{x_{ji}, \delta(i)\}$$

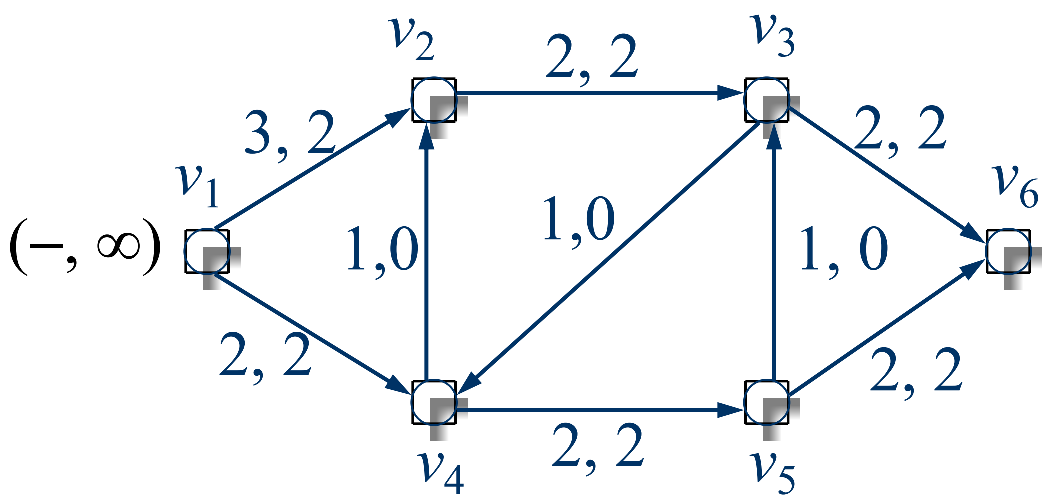
例1



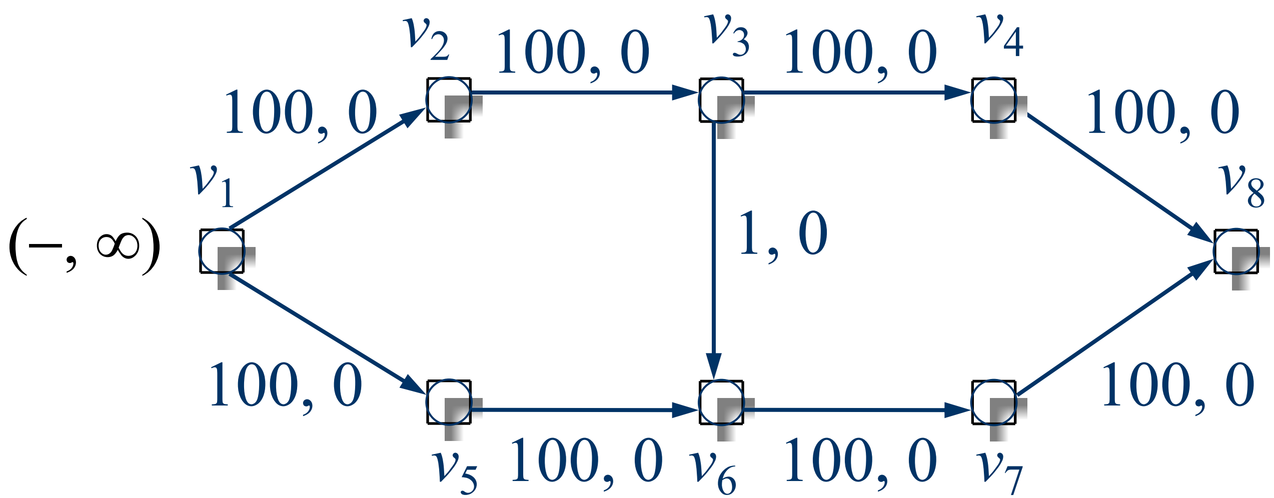
例2：增广路通过逆向弧的例子



在s-t增广路上增加值为1的流之后

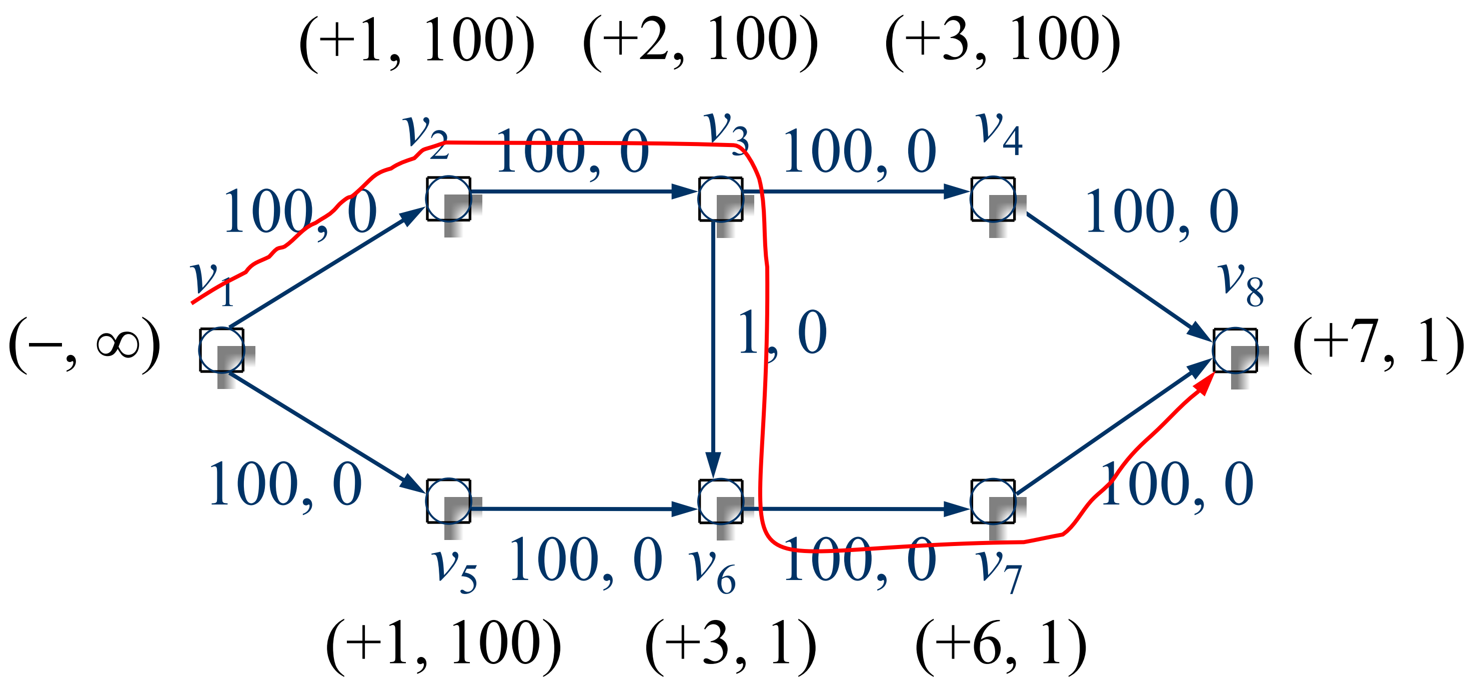


例3



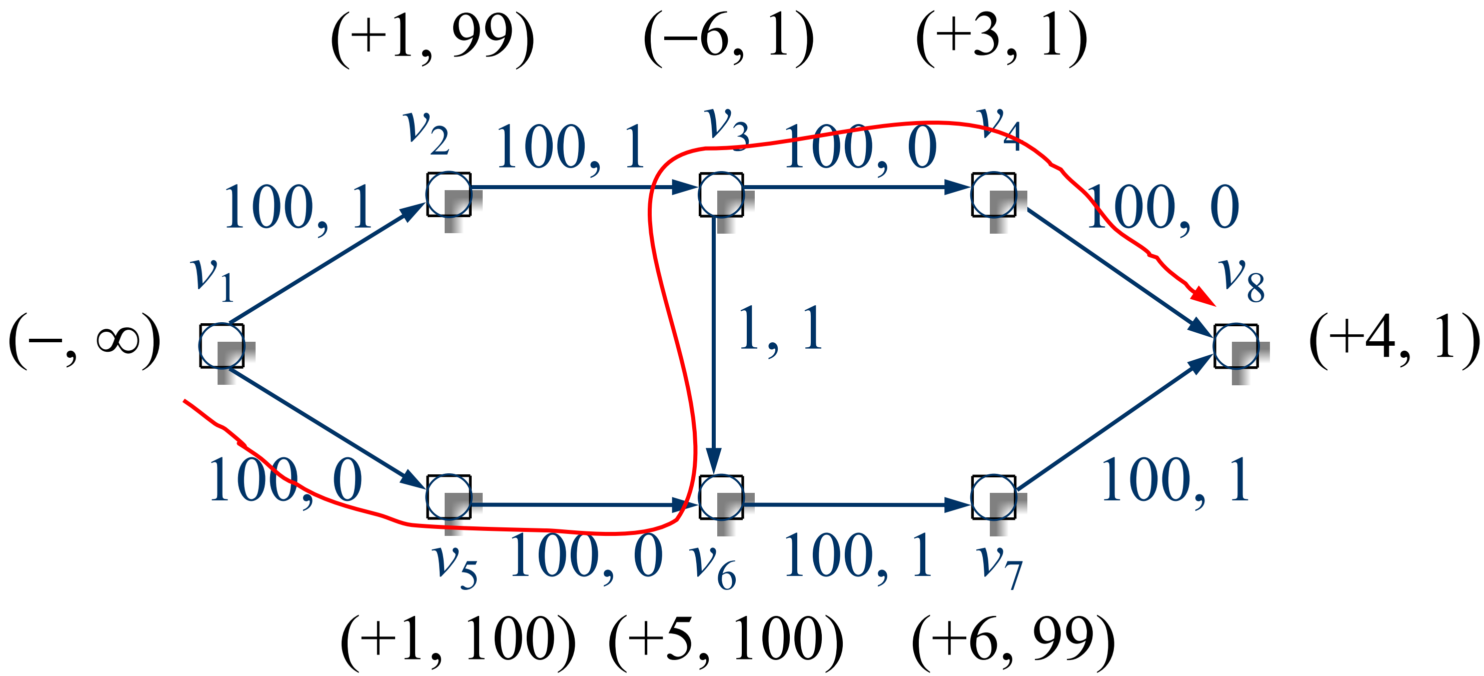
v_1
$(-, \infty)$
v_2
v_3
v_4
v_5
v_6
v_7
v_8
100, 0
100, 0
100, 0
100, 0
1, 0
100, 0
100, 0
100, 0
100, 0

例3



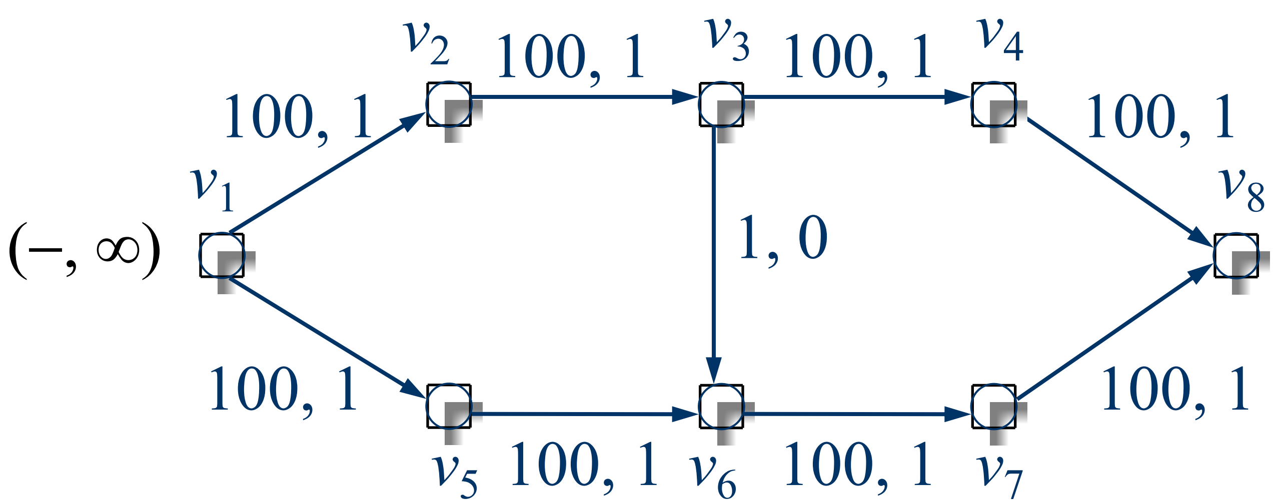
(+1, 100)
(+2, 100)
(+3, 100)
v_2
100, 0
v_3
100, 0
v_4
100, 0
100, 0
v_1
v_8
(−, ∞)
1, 0
(+7, 1)
100, 0
100, 0
v_5
100, 0
v_6
100, 0
v_7
(+1, 100)
(+3, 1)
(+6, 1)

例3



(+1, 99)
(−6, 1)
(+3, 1)
v_2
100, 1
v_3
100, 0
v_4
100, 1
100, 0
v_1
v_8
(−, ∞)
1, 1
(+4, 1)
100, 0
100, 1
v_5
100, 0
v_6
100, 1
v_7
(+1, 100)
(+5, 100)
(+6, 99)

例3



v_2
100, 1
v_3
100, 1
v_4
100, 1
100, 1
v_1
v_8
(−, ∞)
1, 0
100, 1
100, 1
v_5
100, 1
v_6
100, 1
v_7

时间复杂度

- 设弧数为 m，每找一条增广路最多需要进行 $2m$ 次弧检查。
- 不失一般性，假设所有弧容量都是整数。则最多需要 v 次增广，其中 v 是最大流值。
- 所以，总的时间复杂度量为 $O(mv)$。

引理1

引理 1：设 x 是一个 s, t-可行流，(S, T)是一个 s, t-割。则 x 的流值等于该割中前向弧的流量之和减去后向弧的流量之和。

- 证明：流 x 的值 $v = \sum_{j\in N^+(s)} x_{sj} - \sum_{j\in N^-(s)} x_{js}$

$$= \sum_{i\in S}\left(\sum_{j\in N^+(i)} x_{ij} - \sum_{j\in N^-(i)} x_{ji}\right)$$

（由流守恒约束）

$$= \sum_{i\in S}\left(\left(\sum_{j\in N^+(i)\cap S} x_{ij} + \sum_{j\in N^+(i)\cap T} x_{ij}\right) - \left(\sum_{j\in N^-(i)\cap S} x_{ji} + \sum_{j\in N^-(i)\cap T} x_{ji}\right)\right)$$

$$= \sum_{i\in S}\left(\left(\sum_{j\in N^+(i)\cap S} x_{ij} - \sum_{j\in N^-(i)\cap S} x_{ji}\right) + \left(\sum_{j\in N^+(i)\cap T} x_{ij} - \sum_{j\in N^-(i)\cap T} x_{ji}\right)\right)$$

引理1

- $$=\sum_{i\in S}\left(\sum_{j\in N^{+}(i)\cap T} x_{ij}-\sum_{j\in N^{-}(i)\cap T} x_{ji}\right)]]$$

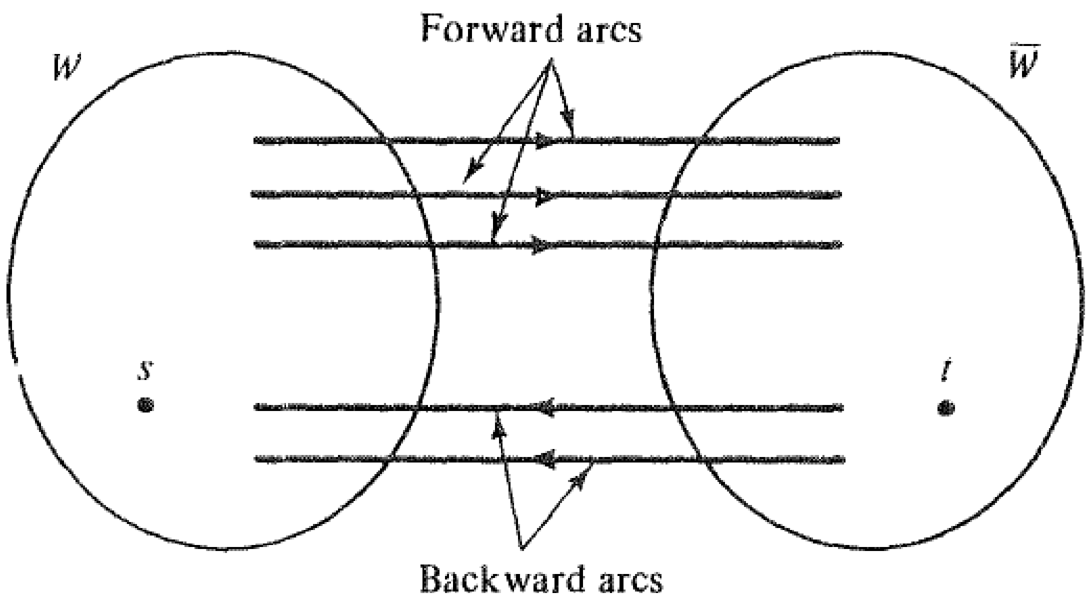


Figure 6–1 A cut in a flow network.

引理2

引理 2：任意一个 *s*，*t*-可行流的流值都不超过任意一个 *s*，*t*-割的容量。

- **证明：设 *x* 是一个 *s*, *t*-流，流值为 *v*。(*S*, *T*)是一个 *s*, *t*-割。**
- **则** $v=\sum_{i\in S}\left(\sum_{j\in N^+(i)\cap T}x_{ij}-\sum_{j\in N^-(i)\cap T}x_{ji}\right)$ **（由引理 1）**

$\le\sum_{i\in S}\sum_{j\in N^+(i)\cap T}x_{ij}$ **（由非负约束）**

$\le\sum_{i\in S}\sum_{j\in N^+(i)\cap T}c_{ij}$ **（由容量约束）**

$=c(S,T)$。‖

增广路定理

定理 6.6.1（增广路定理） 一个可行流 x 是最大流 当且仅当 不存在关于它的从 s 到 t 的增广路。

- 证明：($\Rightarrow$)。若存在增广路，则 x 的流值可以严格增加，流 x 就不是最大的。
- ($\Leftarrow$)。已知可行流 x 不存在从 s 到 t 的增广路。下面证 x 是最大流。
- 定义 S 是从 s 出发经过任一条增广路可达的顶点的集合。特别地，$s \in S$。定义 $T = V - S$。
- 由 S 和 T 的定义，可知对任意的弧 $(i, j) \in (S, T)$，有 $x_{ij} = c_{ij}$；对任意的弧 $(j, i) \in (T, S)$，有 $x_{ji} = 0$。

增广路定理

- 首先，从 s 经过增广路 P 可达 i。
- 若 $x_{ij} < c_{ij}$，则由增广路的定义，$P \cup \{(i, j)\}$是从 s 到 j 的增广路，与 $j \in T$ 矛盾。
- 类似地，若 $x_{ji} > 0$，$P \cup \{(i, j)\}$也是从 s 到 j 的增广路，与 $j \in T$ 矛盾。

- 由引理 1，流值 $v = \sum_{i \in S} \left(\sum_{j \in N^+(i) \cap T} x_{ij} - \sum_{j \in N^-(i) \cap T} x_{ji} \right) = \sum_{i \in S} \sum_{j \in T} c_{ij}$。
- 由引理 2，任意流的流值都小于等于任意割的容量。现在 x 的流值等于割(S, T)的容量，因此 x 是最大流。

最大流最小割定理

定理 6.6.3 一个流网络上最大流的**流值**等于该网络上最小割的**容量**。

- 证明：设 x^*是一个最大流，v^*是其流值。(S^*, T^*)是一个最小割，c(S^*, T^*)是其容量。
- 由引理 2，$v^* \le c(S^*, T^*)$。
- 由增广路定理的证明，存在割(S, T)，使得 $v^* = c(S, T)$。
- 因为(S^*, T^*)是最小割，有 $c(S, T) \ge c(S^*, T^*)$，即 $v^* \ge c(S^*, T^*)$。
- 于是 $v^* = c(S^*, T^*)$。

Max-Flow Min-Cut定理

- **Lester R. Ford, Delbert R. Fulkerson.**
  Maximal flow through a network.
  ***Canadian Journal of Mathematics*, vol. 8, 1956.**
- 以及
- **P. Elias, A. Feinstein, C. E. Shannon.**
  A note on the maximum flow through a network.
  ***IRE Transaction on Information Theory IT2*, 117-119, 1956.**

整流定理

定理 6.6.2 如果网络中所有弧的容量都是整数，则最大流的值也为整数。

- 证明：从 0 流开始沿 *s*-*t* 增广路增加流。由于弧上的容量都是整数，每次在增广路上增加的流值都为整数。当最后不能增加时，就得到了最大流，其值仍为整数。

最大流问题的LP

定义 x_{ij} 为弧(i, j)上从顶点 i 到顶点 j 的流量。则最大流问题的 LP 可写为：

$$
\begin{array}{lll}
\max & v & \\
\text{s.t.} & \displaystyle\sum_{j\in N^{+}(i)} x_{ij} - \sum_{j\in N^{-}(i)} x_{ji} = v, & i = s \\
& \displaystyle\sum_{j\in N^{+}(i)} x_{ij} - \sum_{j\in N^{-}(i)} x_{ji} = -v, & i = t \\
& \displaystyle\sum_{j\in N^{+}(i)} x_{ij} - \sum_{j\in N^{-}(i)} x_{ji} = 0, & \forall i \neq s, t \\
& x_{ij} \leq c_{ij}, & \forall (i, j) \in E \\
& x_{ij} \geq 0, & \forall (i, j) \in E
\end{array}
\qquad \textbf{(LP)}
$$

图6.6.1

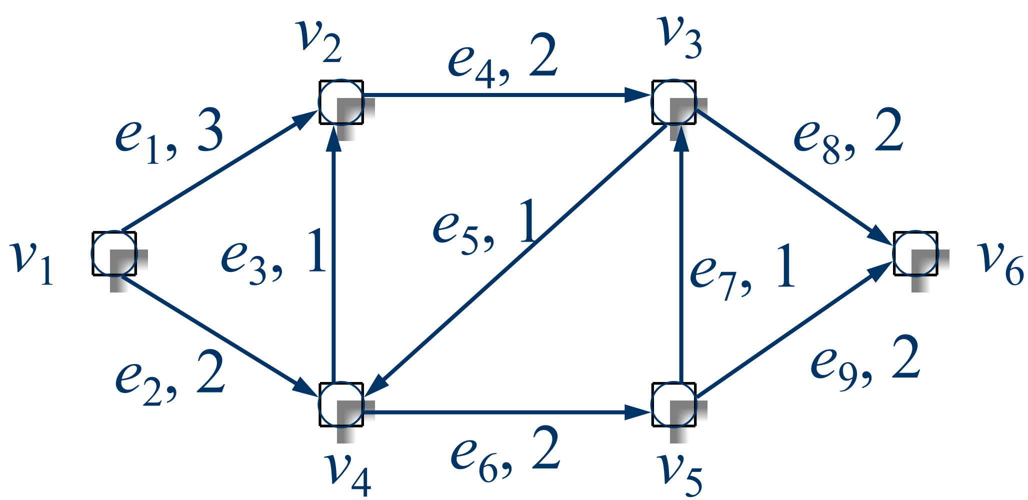


图6.6.1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	v	
	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9		
价	0	0	0	0	0	0	0	0	0	1	右
s	1	1	0	0	0	0	0	0	0	1	0
v_2	–1	0	–1	1	0	0	0	0	0	0	0
v_3	0	0	0	–1	1	0	–1	1	0	0	0
v_4	0	–1	1	0	–1	1	0	0	0	0	0
v_5	0	0	0	0	0	–1	1	0	1	0	0
t	0	0	0	0	0	0	0	–1	–1	–1	0
e_1	1									0	3
e_2		1								0	2
e_3			1							0	1
⋮				⋮						⋮	⋮
e_9									1	0	2

最大流问题的LP的简写形式

- 定义 A 为图 G 的点弧关联矩阵。
- 定义向量 d 为：

$$d_i=\begin{cases}1, & i=s\\ -1, & i=t\\ 0, & \text{o.w.}\end{cases}$$ 。

- 则最大流问题的 LP 可简写为：

$$\begin{aligned}\max\quad & v\\ \text{s.t.}\quad & Ax+dv=0\\ & x\le c\\ & x\ge 0\end{aligned}$$ 。

最大流问题LP的对偶

线性规划(LP)的对偶为：

$$
\begin{array}{lll}
\min & \sum\limits_{(i,j)\in E} c_{ij}\beta_{ij} & \\
\text{s.t.} & \beta_{ij} \ge \alpha_j - \alpha_i, & \forall (i,j)\in E \\
& \alpha_t - \alpha_s \ge 1 & \\
& \alpha_i \text{无限制}, & \forall i \in V \\
& \beta_{ij} \ge 0, & \forall (i,j)\in E
\end{array}
\qquad \textbf{(DP}_1\textbf{)}
$$

可以证明，(DP$_1$)与(DP$_2$)等价：

$$
\begin{array}{lll}
\min & \sum\limits_{(i,j)\in E} c_{ij}\beta_{ij} & \\
\text{s.t.} & \beta_{ij} \ge \alpha_j - \alpha_i, & \forall (i,j)\in E \\
& \alpha_t - \alpha_s \ge 1 & \\
& \alpha_i \ge 0, & \forall i \in V \\
& \beta_{ij} \ge 0, & \forall (i,j)\in E
\end{array}
\qquad \textbf{(DP}_2\textbf{)}
$$

最小割问题

最小 *s*-*t* 割问题

- **实例：有向图 $G=(V,E)$，弧 $e=(i,j)$上定义有容量 $c_e \geq 0$。源顶点 *s* 和目标顶点 *t*。**
- **目标：求一个最小 *s*-*t* 割。**

最小割问题的整数规划：

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}\beta_{ij} \\
\text{s.t.} \quad & \beta_{ij} \geq \alpha_j - \alpha_i, && \forall (i,j)\in E \\
& \alpha_t - \alpha_s \geq 1 \\
& \alpha_i \in \{0,1\}, && \forall i \in V \\
& \beta_{ij} \in \{0,1\}, && \forall (i,j)\in E
\end{aligned}
$$

可以看出，对偶规划(DP_2)正是最小割问题的线性规划松弛。

谢谢大家