§ 6.4 求最大流的有效算法

给定一个具有流 F 的网络$N = (V, E)$，我们构造一个伴随有向图$N^F = (V, E')$，满足 N 中的 F 可增广路和N^F中的 x 到 y 的有向路径一一对应。N 和N^F的顶点集相同，并且对任意两个顶点 u 和 v，(u, v)是N^F中的一条边，当且仅当，或者$(u, v) \in E$并且$c(u, v) - f(u, v) > 0$，或者$(v, u) \in E$ 并且 $f(v, u) > 0$。

°在 N 中找从 x 到 y 的可增广路归结为在N^F中找 x 到 y 的有向路径。

°$L(v)$：对每个顶点设一个标号$L(v)$等于N^F中从 x 到 v 的最短距离。如果不存在从 x 到 v 的路径，则$L(v) = 0$。

°N^F中的 x 到 y 的有向路记为P^F，它在 N 中的可增广路记为 P。

°如果N^F中存在从 x 到 y 的有向路P^F，那么P^F是从 x 到 y 的最短有向路。可以从 y 向前回溯，当处理到一个顶点 v 时，取它的前一个顶点 u，满足$L(u) = L(v) - 1$。

算法 1：BFSPK(宽度优先找P^F的过程)

BEGIN

在N^F中宽度优先找 x 到每个顶点 v 的最短距离(如果$v \neq x$,L(v)>0 表示这个路径的长度；如果 L(v)=0, 那么这样的路径不存在。);

```
IF L(y)=0 THEN PATH←false
    ELSE BEGIN
  FOR 所有v ∈ V  DO  构造B′(v)  /*B′(v)是 v 的反向邻接表*/
  P^F ← (y);
  u ← y;
  WHILE  u ≠ x  DO
      BEGIN
      在 B′(u)中找一个顶点 v，使得L(u) = L(v) + 1;
       把 v 加到P^F的前端；
     u ← v;
      END
    END
  END;
```

算法：求最大流的算法

```
输入网络N = (V, E)的邻接表 A(v)，边上的容量及各边上的初始流;
PATH ←true;
WHILE PATH=true DO
   BEGIN
   构造N^F的邻接表 B(v)，对每一条边(u, v)记录Δ(u, v)和(u, v)是正向边还是反向边；
    BFSPK;
    IF PATH=true THEN
        BEGIN
        求Δ= minΔ(u, v)，(u, v) ∈ P^F
        FOR all  (u, v) ∈ P^F  DO
        IF (u, v)是P^F的向前边  THEN  f(u, v) ← f(u, v) + Δ
        ELSE  f(u, v) ← f(u, v) − Δ;
         END
```

END;
例子：见图 6.7
*用上述算法求最大流

§ 6.5 连通度与 Menger 定理

在本节，我们利用最大流最小割定理证明若干由 Menger(1927)提出的定理，下述引理提供了证明的关键。

引理 6.5：设 N 是以 x 为发点以 y 为收点的网络，并且它的每条弧都具有单位容量，则

(a)N 中最大流的值等于 N 中弧不重的有向(x,y)路的最大数目 m；并且

(b)N 中最小割的容量等于 N 中删去后就会破坏 N 中所有有向(x,y)路的那些弧的最小数目 n。

证明：设f^*是 N 中的最大流，并且D^*表示从 D 中删去所有f^*零的弧而得到的有向图。由于 N 的每条弧有单位容量，因此，对于所有$a \in A(D^*)$，都有$f^*(a) = 1$。由此推得：

(i)$d^+_{D^*}(x) - d^-_{D^*}(x) = \mathrm{val}\, f^* = d^-_{D^*}(y) - d^+_{D^*}(y)$;

(ii)$d^+_{D^*}(v) = d^-_{D^*}(v)$, 对所有$v \in V - \{x, y\}$成立。

所以(由习题 10.3.3)，在D^*中，因而也在 D 中存在$\mathrm{val}\, f^*$条弧不重的有向(x, y)路。于是

$$\mathrm{val}\, f^* \leq m \qquad (6.10)$$

现在设$P_1, P_2, \cdots, P_m$是 N 中任一组 m 条弧不重的有向(x, y)路，并由

$$f(a) = \begin{cases} 1, & 若\ a\ 是 \bigcup_{i=1}^{m} P_i 的弧 \\ 0, & 否则 \end{cases}$$

定义 A 上的一个函数 f，显然 f 是 N 中值为 m 的一个流。由f^*是最大流，我们有

$$\mathrm{val}\, f^* \geq m \qquad (6.11)$$

于是从(6.10)式和(6.11)式推得

$$\mathrm{val}\, f^* = m\ 。$$

设$\tilde{K} = (S, \bar{S})$是 N 中的最小割。则在$N - \tilde{K}$中，$\bar{S}$的顶点从 S 中的任何顶点出发都不可到达；特别是，y 从 x 出发不可到达。于是$\tilde{K}$是一个删去它就会破坏所有有向(x, y)路的弧集，并且有

$$\mathrm{cap}\, \tilde{K} = |\tilde{K}| \geq n \qquad (6.12)$$

现在设 Z 是删去它就会破坏所有有向(x, y)路的 n 条弧的集，并且用S'表示在$N - Z$中从 x 出发可到达的所有顶点的集。由于$x \in S'$而$y \in \overline{S'}$，所以$K = (S', \overline{S'})$是 N 中的一个割。此外，根据S'的定义，$N - Z$不能包含$(S', \overline{S'})$的弧，因此$K \subseteq Z$。由于$\tilde{K}$是最小割，我们断定：

$$\mathrm{cap}\, \tilde{K} \leq \mathrm{cap}\, K = |K| \leq |Z| = n \qquad (6.13)$$

于是(6.12)式和(6.13)式一起产生

$$\mathrm{cap}\, \tilde{K} = n\ 。 \qquad \blacksquare$$

定理 6.6: 设 x 和 y 是有向图 D 的两个顶点。则 D 中弧不重的有向(x,y)路的最大数目等于删去后就会破坏 D 中所有有向(x,y)路的那些弧的最小数目。

证明：对 D 的每条弧都指定单位容量，得到一个以 x 为发点以 y 为收点的网络 N。于是定理从引理 6.5 和最大流最小割定理 6.4 推得。■

采用简单的技巧可以立即得到定理 6.6 对于无向图的变形。

定理 6.7：设 x 和 y 是图 G 的两个顶点。则 G 中边不重的(x,y)路的最大数目等于删去后就会破坏 G 中所有(x,y)路的那些边的最小数目。

证明：应用定理 6.6 于 G 的伴随有向图 D(G)即可。 ■

*图 G=(V,E)的伴随有向图 D(G)：$V(D)=V(G), E(D)=\{<u,v>,$
$<v,u> \mid (u,v)\in E(G)\}$。

推论 6.7：图 G 是 k 边连通的当且仅当 G 中任意两个相异顶点被至少 k 条边不重的路所连。

证明：直接从定理 6.7 和 k 边连通的定义推得。 ■

*一个图$G=(V,E)$是 k 边连通的，是说：G 是连通的，并且至少要
在 G 中删除 k 条边，才能使 G 不连通。

*图$G=(V,E)$的边连通度$\kappa'(G)$等于 G 中最小边割集E'的边数$|E'|$。

定理 6.8：设 x 和 y 是有向图 D 的两个顶点，并且 D 没有从 x 到 y 的弧。则 D 中内部不相交的有向(x,y)路的最大数目等于删去后就会破坏 D 中所有有向(x,y)路的那些顶点的最小数目。

证明：从 D 出发构作新的有向图D'如下：

(i)把每个顶点$v\in V-\{x,y\}$分裂成两个新的顶点v'和 v''，并用弧(v',v'')连接它们；

(ii)把 D 中以$v\in V-\{x,y\}$为头的每条弧用以v'为头的新弧来代替，而把 D 中以$v\in V-\{x,y\}$为尾的每条用以v''为尾的新弧来代替。图 6.8 对这种构作方法作了直观说明。见图 6.8。

于是，D'中的每条有向(x,y)路都对应着收缩所有(v',v'')型的弧之后得到的 D 中的一条有向(x,y)路；并且反之，D 中的每条有向(x,y)路也都对应着分裂这条路的每个内部顶点之后得到的D'中的一条有向
(x,y)路。此外，D'中两条有向(x,y)路是弧不重的，当且仅当 D 中对应的路是内部不相交的。由此推得，D'中弧不重的有向(x,y)路的最大数目等于 D 中内部不相交的有向(x,y)路的最大数目。类似地，D'中删去后就会破坏所有有向(x,y)路的那些弧的最小数目等于 D 中删去后就会破坏所有有向(x,y)路的那些顶点的最小数目。于是此定理从定理 6.6 推得。 ■

定理 6.9：设 x 和 y 是图 G 的两个不相邻的顶点。则 G 中内部不相交的(x,y)路的最大数目等于删去后就会破坏所有(x,y)路的那些顶点的最小数目。

证明：对 G 的伴随有向图 D(G)应用定理 6.8 即可。立即可以得到下述推论：

推论 6.9：一个$\nu\geq k+1$的图 G 是 k 连通的当且仅当 G 的任何两个相异顶点被至少 k 条内部不相交的路所连。

*一个图$G=(V,E)$是 k 连通的，是说：G 是连通的，并且至少要在 G 中删除 k 个顶点，才能使 G 不连通。

*图$G=(V,E)$的连通度$\kappa(G)$等于 G 中最小顶点割的顶点数。完全图K_n的连通度$\kappa(K_n)=n-1$.

算法 1：（求无向图 G 的边连通度的算法）

输入无向图 G 并构造$\overline{G}$；/*$\overline{G}$是 G 的伴随有向图*/
选定顶点 u；
$\kappa' \leftarrow |E|$；
FOR 所有$v \in V - \{u\}$ DO
BEGIN
找最大流 F（在$\overline{G}$中，$x = u$ 且 $y = v$）；
IF $F < \kappa'$ THEN $\kappa' \leftarrow F$；
END
输出κ'。

算法 2：（求无向图$G = (V, E)$的点连通度的算法）
输入 G 并构造$\overline{G}$；
/*$\overline{G}$是 G 的伴随有向图对应的图D'如定理 6.8*/
$\kappa \leftarrow n$；
$i \leftarrow 0$；
WHILE $i \ngtr \kappa$ DO
BEGIN
$i \leftarrow i + 1$；
FOR $j := i + 1$ TO n DO
BEGIN
IF $(v_i, v_j) \notin E$ THEN 找最大流 F(在$\overline{G}$中, $x = v_i$且 $y = v_j$)；
IF $F < \kappa$ THEN $\kappa \leftarrow F$；
END
END
输出κ；

§6.6 最小代价流算法

°给定网络 N，其中每条边(u, v)上有一个容量$c(u, v)$和一个非负的代价参数$a(u, v)$, $a(u, v)$等于在边上运输一个单位的流的代价。
°本节的问题是在网络 N 中找一个最大流 F（F 的值为 V），并在 N 中找流值为 V 的流中代价最小的那个流 F。
表示为线性规划问题：

$$\text{minimise} \sum_{(u,v)} a(u,v) f(u,v) \qquad \text{(i)}$$

约束条件：

$$\sum_{v} \big(f(u,v) - f(v,u)\big) = 0, \quad \text{for all } u \neq x \text{ or } y \qquad \text{(ii)}$$

$$\left(\sum_{v} \big(f(x,v) - f(v,x)\big) - V\right) = 0 \qquad \text{(iii)}$$

$$\left(\sum_{v} \big(f(y,v) - f(v,y)\big) + V\right) = 0 \qquad \text{(iV)}$$

$$f(u,v) \le c(u,v), \quad \text{for all } (u,v) \qquad \text{(v)}$$

$$\text{非负条件：} f(u,v) \ge 0, \qquad \text{for all } (u,v) \qquad \text{(vi)}$$

*解释以上的公式

变换目标函数：

$$\text{maximise}\left(pV - \sum_{(u,v)} a(u,v)f(u,v)\right) \qquad \text{(vii)}$$

其中 p 是网络路径上允许的代价，p 从小到大取$0,1,2,\cdots$，当第一次取到某个 p 值时，网络能求出最大流$F = V$，则这时的最大流就是最小代价的最大流。

算法：最小代价流算法

FOR 所有 $u \in V$ DO $\pi(u) \leftarrow 0$;

FOR 所有$(u,v) \in E$ DO $f(u,v) \leftarrow 0$;

$TEST \leftarrow true$;

执行标号过程；/*求最大流的网络标号过程*/

IF 收点被标记 THEN /*则流可增广*/

BEGIN

修改边上的流，如果$V = V'$则停止；

$TEST \leftarrow true$

GOTO 4;

END;

IF 收点没有被标记到 THEN

BEGIN

IF TEST THEN 如果 V 流已饱和网络，则停止；

修改顶点数$\pi(u)$；

$TEST \leftarrow false$;

GOTO 4;

END;

例子：见图 6.9 。