运筹学与控制论

DOI:CNKI:50-1165/N.20120115.1809.003

混合图网络上的 s-t-流*

程 丛 电
(沈阳师范大学 数学与系统科学学院,沈阳 110034)

摘要:在混合图的框架下,给出网络上路段、路径、路径系统、路段 s-t-流、路径 s-t-流及正向路径 s-t-流等定义,并表明无圈路径系统上的最大流一定是正向路径 s-t-流。设计一个分解路段 s-t-流为路径 s-t-流的多项式时间的分解算法,并做算法分析证明其可行性与复杂性。给出并证明一个表现分解前后的路段流与路径流之间关系的分解定理。给出并证明关于路段 s-t-流的收发点的流量守恒公式。进一步讨论两种流的互相转化及其有关性质,特别地,给出了它们互相转化的方式,并证明了当它们互相转化时流值不变。此项工作改进与推广了 Ford 和 Fulkerson,Korte 和 Vygen 及其它学者关于 s-t-流的基础理论工作。

关键词:混合图;网络;s-t-流;分解;算法;最大流
中图分类号:O221;O157 文献标志码:A 文章编号:1672-6693(2012)01-0012-06

网络流(Network flow)是一项既古老又年青的数学研究内容。早在上个世纪的中叶 Ford 和 Fulkerson 关于网络流的系列工作就奠定了网络流这一学术研究领域的基础[1-3]。随着社会生产和科学技术的发展,特别是计算机的飞速发展,实际中不断地涌现出各种各样的网络流问题,顺应着这一时代的潮流,理论界关于网络流的研究蓬勃发展,有关成果与日俱增,参见文献[4-8]。尽管如此,就目前的文献状况来看,网络流的基础理论还不够完善,应用技术还需要大力地加强。本文试通过定义混合图网络上的路段 s-t- 流与路径 s-t-流,给出一个分解路段 s-t- 流为路径 s-t- 流的多项式算法和改进 Fulkerson 关于 s-t- 流的分解定理等工作,为促进网络流理论及其应用的发展做一份贡献。

1 预备知识

本节简略地介绍必要的基础知识。

定义 1 设 $G=(V,E)$ 是一个图(Graph),这里 V 表示该图的全体顶点(Vertices),E 表示其全体边(Edges);当它的每一条边都无方向时,称其为一个无向图(Undirected graph);当它的每一条边都有方向时,称其为一有向图(Directed graph);当它既有无方向的边,又有有方向的边时,称其为一混合图[9-10]。

以下当无特殊说明时,所提到的图均可为此 3 种图中的任何一种。

定义 2 设 G 是一个图,$c:E\to\mathbf{R}_+$ ($=(0,+\infty)$) 是一个 E 到 $\mathbf{R}_+$ 的单值映射,二元组合 (G,c) 叫做一个网络(Network),映射 c 称为 G 上一个容量(Capacity)。

定义 3 关于网络 (G,c),设 $e\in E$,u 和 w 是 e 的两端点,称 (u,e,w) 为一个以 e 为边的路段,记为 $r(e)$。下面分别用 $b(r)$ 和 $d(r)$ 表示其始点和终点(当 e 有向时,u 应为始点,w 应为终点;当不必指出 e 时,将 (u,e,w) 简记为 (u,w);当 e 为无向边时,用 r^- 来表示路段 (u,e,w))。设 $u,v_1,v_2,\cdots,v_l,w\in V$,$(u,v_1)$,$(v_1,v_2)$,$\cdots$,$(v_l,w)$ 是一组无重复边的路段,由这组路段依次首尾相接组成的图形叫做一条 u-w- 道路(u-w-walk);当 $u=w$ 时,称这一道路是闭的。除始点 u 和终点 w 以外没有重点的道路叫做一条 u-v- 路径(u-v-path),记为 $P=[u,$

* 收稿日期:2011-07-19 网络出版时间:2012-01-15 18:09:00
资助项目:辽宁省教育厅科研项目(No. L2010514)
作者简介:程丛电,男,教授,研究方向为数学规划与泛函分析。
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20120115.1809.201201.12_003.html

$v_1, v_2, \cdots, v_l, v]$ ；当 $u = w$ 时，称 P 为一个路径圈（Circuit path or cycle path）。w-u- 路径也叫做负向的 u-w- 路径。关于 $s, t \in V$，设 $\mathcal{P}^-$ 是一族 t-s- 路径，$\mathcal{P}^+$ 是一族 s-t- 路径，$\mathcal{C}$ 是不同时过 s 与 t 的圈组成的集合，则 $\mathcal{P} = (\mathcal{P}^- \cup \mathcal{P}^+ \cup \mathcal{C})$ 称为一个 s-t- 路径系统。当 $\mathcal{C} = \varnothing$ 时，称其为无圈的路径系统；当 $\mathcal{C} = \varnothing, \mathcal{P}^- = \varnothing$ 时，称其为正向 s-t- 路径系统；当 $\mathcal{P}$ 为 G 中的全体正向 s-t- 路径时，称其为 G 上的最大正向 s-t- 路径系统。

定义 4　给定网络 (G, c)，设 $\mathcal{R} = \{r(e): e \in E\}$，$s, t \in V$，若映射 $f: \mathcal{R} \to \mathbf{R}_+$ 满足

$$\begin{cases} f(r) \leqslant c(e), e \text{ 是 } G \text{ 的有向边} \\ f(r) + f(r^-) \leqslant c(e), e \text{ 是 } G \text{ 的无向边} \end{cases}, \forall e \in E \text{ 和 } r = r(e) \tag{1}$$

$$\sum_{r \in F^-(v)} f(r) = \sum_{r \in F^+(v)} f(r), \forall v \in (V - \{s, t\}) \text{（中介点守恒公式）} \tag{2}$$

这里，$F^-(v) = \{r \in \mathcal{R}: d(r) = v\}$，$F^+(v) = \{r \in \mathcal{R}: b(r) = v\}$，则称 f 为 (G, c) 上的一个路段（弧段）s-t- 流，简称 s-t- 流，或流；$v(f) = \sum_{e \in F^+(s)} f(e) - \sum_{e \in F^-(s)} f(e)$ 称为 f 的流值。(G, c) 上的全体流记为 $F[(G, c)]$，通常称 s 为发货点，t 为收货点。

定义 5　设 $\mathcal{P}$ 为 (G, c) 上的一个 s-t- 路径系统。若映射 $y: \mathcal{P} \to \mathbf{R}_+$ 满足

$$\sum_{e \in E(P), P \in \mathcal{P}} y(P) \leqslant c(e), \forall e \in E \tag{3}$$

则称 y 为 (G, c) 上的一个路径 s-t- 流，或 $\mathcal{P}$ 上的 s-t- 流，简称 $\mathcal{P}$ 上的流，这里 $E(P) = \{e: P \text{ 经过边 } e\}$ 称

$$V(y) = \sum_{P \in \mathcal{P}^-} y(P) - \sum_{P \in \mathcal{P}^-} y(P) \tag{4}$$

为 y 的流值。$\mathcal{P}$ 上的全体流记为 $F[\mathcal{P}]$。当 $\mathcal{P}^- = \varnothing$ 或 y 在 $\mathcal{P}^-$ 上为 0 时，称 y 为 $\mathcal{P}$ 上无逆 s-t- 流；当 $(\mathcal{P}^- \cup \mathcal{C}) = \varnothing$ 或 y 在 $(\mathcal{P}^- \cup \mathcal{C})$ 上为 0 时，称 y 为 $\mathcal{P}$ 上的正向 s-t- 流。

命题 1　设 y 为路径系统 $\mathcal{P}$ 上的一个流，$v \in V(\mathcal{P})$（这里，$V(\mathcal{P})$ 表示 $\mathcal{P}$ 中路径上的点的全体），令

$F_P^-(v) = \{P: P \in \mathcal{P}$，存在 u 使 $(u, v) \in \mathcal{R}(P)\}$，$F_P^+(v) = \{P: P \in \mathcal{P}$，存在 w 使 $(v, w) \in \mathcal{R}(P)\}$

这里 $\mathcal{R}(P) = \{r: r \in P\}$，则当 $v \neq s, t$ 时，有 $F_P^-(v) = F_P^+(v)$，且

$$\sum_{P \in F_P^+(v)} y(P) - \sum_{P \in F_P^-(v)} y(P) = 0 \text{（中介点守恒公式）} \tag{5}$$

证明　设 $P \in \mathcal{P}$，由于 $v \neq s, t$，当存在 $u \in V(\mathcal{P})$ 使得 $(u, v) \in \mathcal{R}(P)$ 时，有唯一的 $w \in V(\mathcal{P})$，使得 $(v, w) \in \mathcal{R}(P)$；反之亦然。因此，命题成立。　　证毕

定义 6　给定的网络 (G, c) 和 $s, t \in V$，问题在 (G, c) 上找一流 f^* 使得

$$V(f^*) = \max\{V(f): f \in F[(G, c)]\} - OPT(G, c) \tag{6}$$

称为 (G, c) 上的最大 s-t- 流问题，简称最大流问题（Maximum flow problems）。其中的 f^* 叫做 (G, c) 上的一个最大流或最大流问题 (6) 式的一个解。设 $\mathcal{P}$ 为 (G, c) 上的一个路径系统，找 $\mathcal{P}$ 上的流 y^* 使得

$$V(y^*) = \max\{V(y): y \in F[\mathcal{P}]\} = OPT(\mathcal{P}) \tag{7}$$

称为 $\mathcal{P}$ 上的最大流问题。

命题 2　无圈路径系统 $\mathcal{P}$ 上的最大流一定是正向流。

2 分解算法

这节将给出一个将路段 s-t- 流分解为路径 s-t- 流的算法，并进行算法分析和根据所建构的算法给出 s-t- 流的分解定理及有关推论。

算法 1　（子程序）输入：网络 (G, c)，$s, t \in V$ 和 (G, c) 上的一个满足 $\{r \in \mathcal{R}: g(r) > 0\} \neq \varnothing$ 的 s-t- 流 g；输出：a）(G, c) 上的一条 s-t- 流路径或一个路径圈（不同时过 s 和 t）P^* 与正的权 $w(P^*)$；b）(G, c) 上的一个满足 $|\{r \in \mathcal{R}: g^*(r) > 0\}| < |\{r \in \mathcal{R}: g^*(r) > 0\}|$ 的 s-t- 流 g^*。这里 $|\{\cdot\}|$ 表示集合 $\{\cdot\}$ 中的元素的个数。

1）从 $\{r \in \mathcal{R}: g(r) > 0\}$ 中取 r^*，并置 $j := 0, i := 1, v_j := b(r^*), v_i := d(r^*)$。

2) 若 $v_i \neq s,t$,执行 i);若 $v_i = t$,执行 ii);若 $v_i = s$,执行 iii)。

i) 若 $v_i \in \{v_j,\cdots,v_{i-1}\}$,从 $\{v_j,\cdots,v_{i-1}\}$ 中找 $v_{i'}$ 使 $v_{i'} = v_i$,令 $P^* = [v_{i'},v_{i'+1},\cdots,v_i]$,转3);否则,从$\mathcal{R}$中找$(v_i,v_{i+1})$ 使 $g((v_i,v_{i+1})) > 0$,并置 $i := i+1$,然后转2)。

ii) 若 $v_j \in \{v_{j+1},\cdots,v_i\}$,从 $\{v_{j+1},\cdots,v_i\}$ 中找 $v_{j'}$ 使 $v_{j'} = v_j$,令 $P^* = [v_j,v_{j+1},\cdots,v_{j'-1},v_{j'}]$,转3);若 $v_j = s$,令 $P^* = [v_j,v_{j+1},\cdots,v_{i-1},v_i]$,然后转3);否则,从$\mathcal{R}$中找$(v_{j-1},v_j)$ 使 $g((v_{j-1},v_j)) > 0$,置 $j := j-1$,转 ii)。

iii) 若 $v_j \in \{v_{j+1},v_{j+2},\cdots,v_i\}$,从 $\{v_{j+1},v_{j+2},\cdots,v_i\}$ 中找 $v_{j'}$ 使 $v_{j'} = v_j$,令 $P^* = [v_j,v_{j+1},\cdots,v_{j'-1},v_{j'}]$,转3);若 $v_j = t$,令 $P^* = [v_j,v_{j+1},\cdots,v_{i-1},v_i]$,然后转3);否则,从$\mathcal{R}(G)$中找$(v_{j-1},v_j)$ 使 $g((v_{j-1},v_j)) > 0$,置 $j := j-1$,转 iii)。

3) 置 $w(P^*) = \min\{g(r):r \in \mathcal{R}(P^*)\}$,做 g^* 如下

$$g^*(r) = \begin{cases} g(r), r \notin \mathcal{R}(P^*) \\ g(r) - w(P^*), r \in \mathcal{R}(P^*) \end{cases}, r \in \mathcal{R}$$

停机。

命题3 算法1是可行的,其运算的复杂性为 $O(n^3)$,这里 $m = |E|$,$n = |V|$。

证明 设$(v_0,v_1) \in \mathcal{R}$且$g((v_0,v_1)) > 0$,在$\mathcal{R}$中找$(v_1,v_2),\cdots,(v_{i-1},v_i)$使$[v_0,v_1,v_2,\cdots,v_{i-1},v_i]$为$G$中的路径且 $g((v_{k-1},v_k)) > 0(1 \leqslant k \leqslant i)$,当 v_i 为 s,或 t,或与某已出现的点重合时停止搜索,称这一做法为一个(v_0,v_1)出发的向前运动;在$\mathcal{R}$中找$(v_{-1},v_0),(v_{-2},v_{-1}),\cdots,(v_j,v_{j+1})$ 使$[v_j,v_{j-1},\cdots,v_{-1},v_0]$为 G 中的路径,且 $g((v_k,v_{k+1})) > 0(j \leqslant k \leqslant -1)$。当 v_j 为 s,或 t,或与已出现的某点重合时停止搜索,称这一做法为一个从(v_0,v_1)出发的向后运动,不论是向前还是向后运动,只要不碰上 s,或 t,或重点就可以继续进行;另一方面,由于 V 中只有 n 个点,它们最多走 n 步,而绝不会无限地进行,这也就是说其结果必然是因为碰上了 s,t 或重点而停止。根据这一事实和算法1的具体过程可知,算法1经有限次运算后停机,并在停机时输出一条 s-t- 路径或一个路径圈 P^*。至于结论输出权 $w(P^*)$ 和 g^* 是显然的。

最后,根据 $w(P^*)$ 的定义,命

$$h(r) = \begin{cases} 0, r \notin \mathcal{R}(P^*) \\ w(P^*), r \in \mathcal{R}(P^*) \end{cases}, r \in \mathcal{R}$$

则 h 是一个流且 $h \leqslant g$,从而 $g^* = g - h$ 也是一个流,且至少在一个路段上有 $g^* = 0$ 而 $g \neq 0$。这也就是说,输出 g^* 是满足 $|r \in \mathcal{R}:g^*(r) > 0| < |r \in \mathcal{R}:g(r) > 0|$ 的 s-t- 流。

综上,算法1是可行的。现再进一步分析其复杂性。该算法共3大步,步1)仅几次运算;由于向前运动和向后运动最多走 n 步,而每步至多 n 次运算就可以完成,所以步2)的运算次数是 $O(n^2)$;由于一个 s-t- 路径或路径圈中最多含有 n 个路段,步3)的运算次数是 $O(n)$,所以算法1的复杂性为 $O(n^3)$。 证毕

算法2 (分解算法)输入:网络(G,c),$s,t \in V$和(G,c)上的一个$\{r:r \in \mathcal{R},f(r) > 0\} \neq \varnothing$的 s-t- 流f。输出:a)(G,c)上的一个 s-t- 路径系统$\mathcal{P}$;b)$\mathcal{P}$上的一个流 y,且 $y(P) \neq 0,\forall P \in \mathcal{P}$。

1) 置 $k := 0$,$f_k = f$,$\mathcal{P} := \varnothing$。

2) 若 $\{r:r \in \mathcal{R},f(r) > 0\} \neq \varnothing$,置 $K = k$,停机;否则,置 $g := f_k$,调用算法1求出 P^*,$w(P^*)$ 和 g^*,然后,令 $P_k = P^*$,$\mathcal{P} := \mathcal{P} \cup \{P_k\}$,$y(P_k) = w(P^*)$;再置 $k := k+1$,$f_k = g^*$,转1)。

定理1 算法2是可行的,其运算复杂性为 $O(mn^3)$,且$|\mathcal{P}| \leqslant 2m$。

证明 可行性是显然的。关于复杂性,根据命题1,在 k 的每次增值以后,$|\{r:r \in \mathcal{R},f_k(r) > 0\}|$都至少减1,且$|\mathcal{P}|$增加1。又$|\mathcal{R}| \leqslant 2m$,而当$|\{r:r \in \mathcal{R},f_k(r) > 0\}| = 0$时停机,所以运算的复杂性为 $O(mn^3)$,且$|\mathcal{P}| \leqslant 2m$。 证毕

注1 当存在着一个与 n 无关的常数 L,使得每条边的平行边的数目都不超过该常数,即 $m \leqslant Ln^2$ 时,有 $O(mn^3) = O(n^5)$,因而算法2是一多项式算法,由于这一条件在大多数情况下都是成立的,所以一般来说算法2是一多项式算法。

定理 2 （分解定理）关于算法 2 中的流 f 和 y，有

$$f(r) = \sum_{r \in \mathcal{R}(P), P \in \mathcal{P}} y(P), \forall r \in \mathcal{R} \tag{8}$$

$$V(f) = \sum_{P \in \mathcal{P}^-} y(P) - \sum_{P \in \mathcal{P}^-} y(P) = V(y) \tag{9}$$

这里，$\mathcal{P}^+$与$\mathcal{P}^-$分别为$\mathcal{P}$中的正与负 s-t- 路径。

证明　关于 $k = 0,1,\cdots,K-1$，做映射 h_k 如下

$$h_k(r) = \begin{cases} 0, r \notin \mathcal{R}(P_k) \\ y(P_k), r \in \mathcal{R}(P_k) \end{cases}, r \in \mathcal{R}$$

则 h_k 是(G,c) 上的一个流，且 $f_k = f_{k-1} - h_{k-1}, k = 1,2,\cdots,K$。由于 $f_K = 0$，有

$$0 = f_{K-1} - h_{K-1} = (f_{K-2} - h_{K-2}) - h_{K-1} = \cdots = f_0 - (\sum_{k=0}^{K-1} h_k)$$

$$f(r) = \sum_{r \in \mathcal{R}(P_k), P_k \in \mathcal{P}} h_k + \sum_{r \notin \mathcal{R}(P_k), P_k \in \mathcal{P}} h_k = \sum_{r \in \mathcal{R}(P), P \in \mathcal{P}} y(P), \forall r \in \mathcal{R}$$

(8) 式成立。

根据(8) 式，有

$$\sum_{r \in F^+(s)} f(r) = \sum_{r \in F^+(s)} \sum_{P \in \mathcal{P}, r \in \mathcal{R}(P)} y(P) = \sum_{r \in F^+(s)} [\sum_{P \in (\mathcal{P}^- \cup \mathcal{C}_s), r \in \mathcal{R}(P)} y(P)] =$$

$$\sum_{r \in F^+(s)} [\sum_{P \in \mathcal{P}^-, r \in \mathcal{R}(P)} y(P)] + \sum_{r \in F^+(s)} [\sum_{P \in \mathcal{C}_s, r \in \mathcal{R}(P)} y(P)] = \sum_{P \in \mathcal{P}^-} y(P) + \sum_{P \in \mathcal{C}_s} y(P)$$

$$\sum_{r \in F^-(s)} f(r) = \sum_{P \in \mathcal{P}^-} y(P) + \sum_{P \in \mathcal{C}_s} y(P)$$

其中，C_s 表示$\mathcal{P}$中所有过点 s 的圈。由此，(9) 式成立。　　证毕

注 2　当 $s = t$ 时，由于在算法 1 的 2) 中总是先考虑去掉圈，所以在最后的输出中，$\mathcal{P}^- = \mathcal{P}^+ = \varnothing$。

注 3　关于 s-t- 流的分解问题，1962 年 Ford 和 Fulkerson 给出与证明了文献[3] 中的定理 2. 2，2000 年 Korte 和 Vygen 给出与证明了文献[4] 中的定理 8. 8，本文的分解算法和分解定理改进与推广了这两个定理。

定义 7　关于 s-t- 流 f，分别称算法 2 所给出的$\mathcal{P}$和 y 为由 f 导出的路径系统和路径流；称 y 在$(\mathcal{P}^- \cup \mathcal{P}^+)$ 上的限制为 f 的去圈导出流。

推论 1　关于算法 2 中的 f，$\mathcal{P}$和 y，设 $r \in F^+(s)$，且 $f(r) > 0$，则存在一个过 s 而不过 t 的路径圈集$\mathcal{C}_s$ 使得 $f(r) = \sum_{r \in \mathcal{R}(P), P \in (\mathcal{P}^- \cup \mathcal{C}_s)} y(P)$。

证明　由(8) 式，$f(r) = \sum_{r \in \mathcal{R}(P), P \in \mathcal{P}^- \cup \mathcal{P}^- \cup \mathcal{C})} y(P)$。因为 $r \in F^+(s)$，所以 $\{P : P \in \mathcal{P}^-, r \in \mathcal{R}(P)\} = \varnothing$；令 $\{P : P \in \mathcal{C}, r \in \mathcal{R}(P)\} = \mathcal{C}_s$，则其为一个过 s 而不过 t 的路径圈集，从而推论 1 成立。　　证毕

推论 2　关于 s-t- 流 f，有(收发点守恒公式)

$$\sum_{r \in F^+(s)} f(r) - \sum_{r \in F^-(s)} f(r) = \sum_{r \in F^-(t)} f(r) - \sum_{r \in F^+(t)} f(r)$$

证明　根据(8) 式，仿照(9) 式的证明可证：$V(f) = \sum_{r \in F^-(t)} f(r) - \sum_{r \in F^+(t)} f(r)$，从而推论 2 成立。　　证毕

推论 3　当 f 是网络(G,c) 上的最大 s-t- 流时，关于算法 2 中的$\mathcal{P}$有$\mathcal{P}^- = \varnothing$。

证明　若$\mathcal{P}^- \neq \varnothing$，从$\mathcal{P}^-$中任取路径 P，并命

$$f_1(r) = \begin{cases} f(r), r \notin \mathcal{R}(P) \\ y(P), r \in \mathcal{R}(P) \end{cases}, r \in \mathcal{R}$$

则 f_1 是(G,c) 上的一个 s-t- 流，且由(8) 式与 $y(P) \neq 0$ 可知 $V(f_1) > V(f)$，这与 f 是最大流是矛盾的。因此，$\mathcal{P}^- = \varnothing$。　　证毕

分解算法和分解定理表明了两种 s-t- 流之间的密切联系，并且使得可以在多项式时间内将路段流转化

为路径流,在许多方面,路径流优越于路段流。从定义来看,路段流的约束分有向边与无向边两种情况,而路径流的约束则是统一的,拿最大流问题来说,推论3说明路段最大流的去圈导出流是一个正向流,这使得从路径流方面考虑最大流问题可以摆脱‘正负混杂”的烦恼。就二者相应的LP来说,一个带有表现中介点守恒要求的复杂等式,而另一个则没有。受到以上事实的促动,最后再进一步讨论一下两种流的转换问题。

3 流的转换

处理流的问题经常需要转换其表出形式,本节在上面工作的基础上对两种流的转换问题做进一步的研究。

命题4 设 y 为$\mathcal{P}$上的路径流,做 f 如下

$$f(r)=\begin{cases}\sum\limits_{r\in\mathcal{R}(P),P\in\mathcal{P}}y(P), & \{P\in\mathcal{P}:r\in\mathcal{R}(P)\}\neq\varnothing\\ 0, & \{P\in\mathcal{P}:r\in\mathcal{R}(P)\}=\varnothing\end{cases},r\in\mathcal{R}$$

则 f 是一个路段流且 $V(y)=V(f)$,称其为由 y 生成的路段流。

证明 由(3)式和(5)式分别知,(1)式和(2)式成立,因此 f 是 (G,c) 上的流。最后,再根据 f 的定义,(4)式以及上面的有关过程得

$$V(f)=\sum_{r\in F^+(s)}f(r)-\sum_{r\in F^-(s)}f(r)=\sum_{r\in F^+(s)}\Big[\sum_{r\in\mathcal{R}(P),P\in\mathcal{P}}y(P)\Big]-\sum_{r\in F^-(s)}\Big[\sum_{r\in\mathcal{R}(P),P\in\mathcal{P}}y(P)\Big]=$$

$$\sum_{P\in\mathcal{P}^-\cup\mathcal{C}_s}y(P)-\sum_{P\in(\mathcal{P}^-\cup\mathcal{C}_s)}y(P)=\sum_{P\in\mathcal{P}^-}y(P)-\sum_{P\in\mathcal{P}^-}y(P)=V(y)$$ 证毕

根据定理2和命题4,立刻可以得到下面的结论。

命题5 由路段流 f 导出的路径流 y 的生成路段流恰好是 f。

命题6 给定网络 (G,c),设 $\tilde{\mathcal{P}}^+$ 是 G 上的最大正向 s-t- 路径系统。

1) 当 f 是 (G,c) 上的一个最大流时,其导出的正向流 y(去圈导出流)在 $\tilde{\mathcal{P}}^+$ 上的不变延拓 $\tilde{y}$(当 $y(P)$ 无定义时规定 $\tilde{y}=0$)是 $\tilde{\mathcal{P}}^+$ 上的最大流,且 $V(f)=V(y)$;

2) 当 y 是 $\tilde{\mathcal{P}}^+$ 上的最大流时,其生成的路段流 f 是 (G,c) 上的一个最大流,且 $V(f)=V(y)$;

3) $OPT[(G,c)]=OPT[\mathcal{P}^+]$。

证明 在1)的条件下,如果 $\tilde{y}$ 不是 $\tilde{\mathcal{P}}^+$ 上的最大流,那么必存在 $y_1\in F[\tilde{\mathcal{P}}^+]$,使得 $V(y_1)>V(\tilde{y})$。设 f_1 为由 y_1 生成的 (G,c) 上的流,则

$$V(f_1)=V(y_1)>V(\tilde{y})=V(f)$$

矛盾。从而,1)成立。在2)的条件下,如果 f 不是 (G,c) 上的最大流,那么必存在 (G,c) 上的流 f_1 使得 $V(f_1)>V(f)$。设 y_1 为由 f_1 导出的正向流在 $\tilde{\mathcal{P}}^+$ 上的不变延拓,则

$$V(y_1)=V(f_1)>V(f)=V(y)$$

矛盾。从而,2)成立。最后,由1)和2)与命题4和命题5知3)成立。 证毕

本节内容进一步提示了两种 s-t- 流的内在联系。特别地,由于一个流在圈上的流量不会影响它的流值,根据命题2,推论3和命题6,对于最大流的存在与求解等问题,仅需在最大正向 s-t- 路径系统上进行考虑。

4 结束语

本文在混合图的框架下给出了路段 s-t- 流与路径 s-t- 流的定义,并设计了一个分解路段 s-t- 流为路径 s-t- 流的多项式算法 根据这一算法改进了 Ford 和 Fulkerson 与 Korte 和 Vygen 关于分解路段 s-t- 流的工作[3-4]。此外,本文还讨论了两种流的转化及有关性质。这些工作对于深化、推广和应用网络流理论具有一定的意义。最后,衷心地期望本文的工作能够促进网络流的理论与应用的发展。

致谢 衷心地感谢越民义老师的大力指导和帮助!

参考文献：

[1] Ford L R, Fulkerson D R. Maximal flow through a network [J]. Canadian Journal of Mathematics, 1956, 8: 399 – 404.

[2] Ford L R, Fulkerson D R. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem [J]. Canadian Journal of Mathematics, 1957, 9: 210 – 218.

[3] Ford L R, Fulkerson D R. Flows in networks [M]. Princeton: Priceton University Press, 1962.

[4] Sheu J B. An emergency logistics distribution approach for quick response to urgent relief demand in disasters [J]. Transp Res: Part E: Logist Transp Rev, 2007, 43: 687 – 709.

[5] Yi W, Kumar A. Ant colony optimization for disaster relief operations [J]. Transp Res: Part E: Logist Transp Rev, 2007, 43: 660 – 672.

[6] 程丛电，唐恒永，赵传立. 一个多物资网络流问题的逼近算法[J]. 辽宁大学学报：自然科学版，2008，35(2)：170 – 171.

[7] 程丛电，李振鹏. 具有全局性公平满意度的最大多物资网络流问题[J]. 应用数学学报，2011，34(3)：502 – 517.

[8] 姜雨，张洪海，夏洪山. 多机场网络流量策略[J]. 系统工程的理论与实践，2011，31(2)：379 – 384.

[9] 谢金星，邢文训，王振波. 网络优化[M]. 2 版. 北京：清华大学出版社，2009.

[10] Korte B, Vygen J. Combinatorial Optimization Theory and Algorthms [M]. Berlin: Springer – Verlag, 2000.

Operations Research and Cybernetics

s-t-Flow on the Network with Hybrid Graph

CHENG Cong-dian

(College of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034, China)

Abstract: Under the framework of hybrid graph, the present work firstly defines the road, path, path system, *s-t*-flows, path *s-t*- flows and positive path *s-t*- flows, etc. on the network, and shows the maximum flow on a path system without cycles to be a positive path *s-t*- flow. Then this paper designs a polynomial time algorithm to decompose an *s-t*-flow into a path *s-t*- flow, as well as prove its feasibility and complexity. Propose and prove a theorem that explains the relation between *s-t*-flow and the *s-t*-path flow with the decomposition. Propose and prove the *s-t*-flow conservation rule between the source and the sink. Finally, it further discusses the transformation between the two kinds of flows, and a few of related problems; in particular, it establishes the formulae for their translating and proves that their value does not change when they translate each other. These works improve and extend the previous corresponding works of Ford and Fulkerson (1962), Korte and Vygen (2000) as well as others.

Key words: hybrid graph; network; *s-t*-flow; decomposition; algorithm; maximum flow

(责任编辑　黄　颖)