网络最大流部分割矩阵算法

毛 华 毛晓亮 李 斌
(河北大学数学与计算机学院 保定 071002)

摘 要 网络最大流问题是图论研究中一个经典的模块。首先,利用粗糙集属性约简的差别矩阵算法思想,定义网络的一个部分割容量矩阵。其次,通过集合的交和并运算,找出网络的所有割集,从而得到最小容量割集。之后,在最大流最小割定理的基础上,得到网络的最大流。

关键词 网络最大流,割集,部分割,最小割
中图法分类号 O224 **文献标识码** A

Partial Cut Set Algorithm for Maximum-flow of Networks

MAO Hua MAO Xiao-liang LI Bin
(School of Mathematics and Computer, Hebei University, Baoding 071002, China)

Abstract Network maximum-flow problem is a classical module in graph theory. First, based on the rough set attribute reduction algorithm of discernibility matrix, it defines a partial cut set matrix. Afterwards, it finds out all the cut by meet and join operations for sets. Furthermore, the minimum cut is yielded out. At last, it gets the maximum flow of the network with the assistance of the theorem of maximum flow minimum cut.

Keywords Network maximum-flow, Cut set, Partial cut set, Minimum cut

1 引言

网络最大流问题是计算机、图论和运筹学的重要内容,一直被广泛应用于工程和科学等许多领域[1]。近年来,由于全球自然灾害频发,各国对于救灾人员的调动、灾民的安置以及灾后重建物资的运输分配等,均给予了足够的重视,这也使得不断研究和改进网络最大流的算法成为必要。从最初Ford—Fulkerson算法[2]的提出,已有50余年了,国内外学者相继对此法进行了不同的改进,比如Dinic算法[3]、Ahuja和Orlin算法[4]、Goldberg和Rao的二分长度阻塞流算法[5]以及网络最大流的割集矩阵算法[6]等等。这些经典的算法的提出和应用,对网络最大流问题的研究和发展起到了非常重要的作用,但是实际问题的产生使得人们还在寻找更有效的算法,希望解决实际工作中对网络最大流的需求。从经典的网络最大流算法可以看出,通常解决网络最大流问题主要有两种思路[7]:一是寻找增广路,在此增广路上增流完毕以后,重新寻找新的增广路增流,直至没有增广路为止;二是从割集出发,应用最大流最小割定理,找出最小割,从而得到最大流。但是增广路本身的寻找就已经很麻烦,而应用最大流最小割定理往往计算量偏大,甚至出现 2^n 个矩阵方程[8]。为了克服这些不足,本文将采用新思路,通过一种新的矩阵形式,将部分割作为矩阵的元素,构建 $n-1$ 阶邻接矩阵,从而,利用该矩阵元素之间的交和并运算,得到网络优化的最大流。这种方式使得算法执行更加直观,运算量相对较小,可行性较强,有较大的理论和实践意义。

本文第2节为本文工作提供所需的基本概念;第3节给出网络最大流部分割矩阵算法;第4节用实例说明所给算法的有效性;最后为本文的结束语。

2 基本概念

下面给出有关网络最大流问题的基本概念,主要内容来自文献[9,10]。

定义1 一个网络 $N=(V,A)$ 是指一个连通无环弧且满足下列条件的有向图:

(1)有一个顶点子集 X,其每个顶点的入度都为0;

(2)有一个与 X 不相交的顶点子集 Y,其每个顶点的出度都为0;

(3)每条弧都有一个非负的权值,称为弧容量。

注1 ①上述网络 N 可记为 $N=(V,X,Y,A,C)$,其中 X 称为网络的发点集或源点集,Y 称为网络的收点集或汇点集,网络中其他顶点称为中转点,C 称为网络的容量函数,它是定义在弧集 A 上的非负函数。

②如果一个网络中的源点集和汇点集都只含一个顶点,则称该网络为单源单汇网络。当对任意一个网络添加两个顶点 s 和 t,分别作为人工源和人工汇后,此网络可导出一个单源单汇网络。因此,本文仅考虑单源单汇网络。

定义2(导出子图) 设 V' 为图 G 的顶点集 $V(G)$ 的一个非空子集,令

到稿日期:2011-06-15 返修日期:2011-09-18 本文受保定市科学技术研究项目(11ZG005[P])资助。

毛 华(1963—),女,博士,教授,主要研究方向为数据挖掘、网络物流、格论、拟阵论,E-mail:yushengmao@263.com;**毛晓亮**(1984—),男,硕士,主要研究方向为网络物流;**李 斌**(1985—),男,硕士,主要研究方向为网络物流。

$E'=\{e'\mid e'\in E(G), e'=v_iv_j, v_i、v_j\in V'\}$，则称 G 的子图 $G'=G[V']$ 为 G 的导出子图。

定义 3(网络的流与割) 网络 $N=(V,X,Y,A,C)$ 中的一个可行流是指定义在 A 上的一个整值函数 f，使得：

(1)对 $\forall a\in A, 0\leqslant f(a)\leqslant c(a)$，(容量约束)；

(2) $\forall v\in V-(X\cup Y), \bar{f}(v)=f^{+}(v)$，(流量守恒)。

其中，$\bar{f}(v)$ 表示点 v 处入弧上的流量之和，$f^{+}(v)$ 表示点 v 处出弧上的流量之和。

定义 4 设 f 是网络 $N=(V,X,Y,A,C)$ 中的一个可行流，则必有 $\bar{f}(v)=f^{+}(v)f^{+}(v)$ 称为流 f 的流量，记为 Val f。N 中流量最大的可行流称为 N 的最大流。

定义 5 设 $N=(V,x,y,A,C)$ 是一个单源单汇网络。$S\subseteq V, \bar{S}=V-S$，用 $(S,\bar{S})$ 表示尾在 S 中而头在 $\bar{S}$ 中的所有弧的集合(即从 S 中的点指向 S 之外点的所有弧集)。若 $x\in S$，而 $y\in\bar{S}$，则称弧集 $(S,\bar{S})$ 为网络的一个割。一个割 $(S,\bar{S})$ 的容量是指 $(S,\bar{S})$ 中各条弧的容量之和，记为 $Cap(S,\bar{S})$。若网络中不存在割 K' 使得 $Cap\,K'<Cap\,K$，则称 K 为网络 N 的最小割。

引理 1 设 f 是网络 N 的一个可行流，K 是 N 的一个割，若 Val $f=Cap\,K$，则 f 是最大流，而 K 是最小割。

为下一节算法讨论的需要，给出如下的定义。

定义 6 (1)部分割：设 v 是网络 $N=(V,x,y,A,C)$ 的任意一个点，由 v 出发的所有出弧构成的集合称为 N 的一个部分割，记为 K_v。

(2)部分割容量矩阵：下面的 $n-1$ 阶部分割容量矩阵 A 称为网络 N 的部分割容量矩阵。

$$\begin{array}{c|c}
 & \begin{array}{ccccc} v_1 & v_2 & \cdots & v_{n-2} & v_{n-1}\end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1}\end{array} & \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1,n-2} & K_{1,n-1} \\ U & K_{22} & \cdots & K_{2,n-2} & K_{2,n-1} \\ \vdots & & & & \vdots \\ U & K_{n-2,2} & \cdots & K_{n-2,n-2} & K_{n-2,n-1} \\ U & K_{n-1,2} & \cdots & K_{n-1,n-2} & K_{n-1,n-1}\end{bmatrix}
\end{array}$$

其中，K_{ij} 表示由 v_i 出发，且不指向 v_j 的所有出弧的弧容量构成的集合，即约束的部分割容量；若 v_i 与 v_j 不关联，则令 $K_{ij}=U$。

注 2 设 N 为任一给定网络，A 为 N 的部分割容量矩阵，取矩阵 A 第一行包含 K_{11} 的子集 $Q=\{K_{11}, K_{1j_1}, \cdots, K_{1j_{d-1}}\}$，以同种方式取定列，构成原矩阵的 d 阶子阵 A_j^d，

$$\begin{bmatrix} K_{11} & K_{1j_1} & \cdots & K_{1j_{d-1}} \\ K_{j_11} & K_{j_1j_1} & \cdots & K_{j_1j_{d-1}} \\ \vdots & \vdots & \vdots & \vdots \\ K_{j_{d-1}1} & K_{j_{d-1}j_1} & \cdots & K_{j_{d-1}j_{d-1}}\end{bmatrix}$$

取定 A_j^d 的第 p 行，作如下运算：$\bigcap_{q=1}^{d-1}K_{pq}$，进而得到：$M^d(j)=\bigcup_{p=1}^{d-1}\bigcap_{q=1}^{d-1}K_{pq}, j_0=1$。由于任意两个部分割 K_{pq}^1, K_{pq}^2 表示的是不同点的出弧组成的集合，因此值相同的弧容量在并运算时仍应加以区别。比如：$A=\{1,2,3\}, B=\{1,2\}, A\cup B=\{1,2,1,2\}$。故在集合 $M^d(j)$ 中将值相同的元素视为不同元素。

定理 1 $M^d(j)$ 为网络 N 的一个割集。

证明：根据定义 6，显然，$\bigcap_{q=1}K_{pq}$ 即为 $f^{+}(v_p)$，$\bigcup_{p=1}\bigcap_{q=1}K_{pq}$ 即为 Q 中所有点的出弧，且不含内部弧，即网络的一个割集。

当遍历整个部分割集容量矩阵 A 时，即得网络的所有割集。

由以上证明可知，当得到一个网络的所有割集以后，只需计算出各个割集的容量即可，进而比较大小，找出最小割，由最大流最小割定理，得到网络的最大流。

定理 2 设 $K=(S,\bar{S})$ 为网络 N 的一个割集，若 $G(S)$ 是非连通的，则 K 一定不是网络 N 的最小割。

证明：若 $G(S)$ 非连通，则存在两个连通分支：$G(S_1), G(S_2)$，由这两个连通分支形成的割集为 $CapK=CapK_1+CapK_2$，显然，$CapK_1<Cap$，故 K 必不是最小割。

由定理 2 可知，当割集 $(S,\bar{S})$ 是最小割时，$G(S)$ 必是连通的，因此，在本文算法中只需找出 N 的所有连通子图对应的割集即可。

3 最大流部分割容量矩阵算法

本节将给出本文的核心内容，即网络的最大流部分割容量矩阵算法，该算法将利用下面的子算法给出。

算法的第一个部分：

算法 1 在部分割容量矩阵 A 中取出所有的割集

Step 1 输入网络的部分割容量矩阵 A；

Step 2 在 A 中取出所有 d 阶子阵；

Step 3 令 $j=1, M=\phi$；

Step 4 $M^d(j)=\bigcup_{p=1}^{d-1}\bigcap_{q=1}^{d-1}K_{pq}, M=M\cup\{M^d(j)\}$；

Step 5 若 $j<m$，则 $j=j+1$，转 Step 4；
若 $j=m$，则停止，输出 M。

为了算法 2 的描述清楚简洁，给出一个符号：对任意集合 $S=\{a_1,a_2,a_3,\cdots a_{n-1},a_n\}$，设置记号 $\mathrm{Plus}(S)=\sum_{i=1}^{n}a_i$。

算法 2 求最小割

$M(j)$ 表示输出的割集的集合；F 表示割集容量。

Step 1 $d=1, F=f^{+}(v_1)$；

Step 2 由算法 1 中输出的 M 得：$\mathrm{Plus}(M^d(j))$，令 $F=\min\{\mathrm{Plus}(M^d(j)), F\}$；

Step 3 若 $d<n-1$，则 $d=d+1$，转 Step 2；
若 $d=n-1$，停止，输出 F。

4 举例

本节将通过具体的实例，说明第 3 节中给出的求最大流部分割容量矩阵算法是有效的。

设网络如图 1 所示。

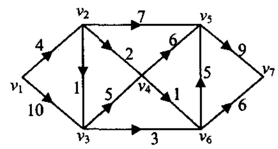

图 1 网络实例

首先，依据算法的要求，做出此网络的 $n-1$ 部分割矩阵。

$$\begin{array}{c|c}
 & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6\end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6\end{array} & \begin{bmatrix} \{4,10\} & \{10\} & \{4\} & U & U & U \\ U & \{1,2,7\} & \{2,7\} & \{1,7\} & \{1,2\} & U \\ U & U & \{3,5\} & \{3\} & U & \{5\} \\ U & U & U & \{6,1\} & \{1\} & \{6\} \\ U & U & U & U & \{9\} & U \\ U & U & U & U & \{6\} & \{5,6\}\end{bmatrix}
\end{array}$$

其次,依据算法 1 和定理 2,需考虑的全体子阵如下。

1 阶子阵:

$[\{4,10\}]$

2 阶子阵:

$$\begin{bmatrix} \{4,10\} & \{10\} \\ U & \{1,2,7\} \end{bmatrix}, \begin{bmatrix} \{4,10\} & \{4\} \\ U & \{3,5\} \end{bmatrix}$$

3 阶子阵:

$$\begin{bmatrix} \{4,10\} & \{10\} & \{4\} \\ U & \{1,2,7\} & \{2,7\} \\ U & U & \{3,5\} \end{bmatrix}, \begin{bmatrix} \{4,10\} & \{10\} & U \\ U & \{1,2,7\} & \{1,7\} \\ U & U & \{6,1\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & U \\ U & \{1,2,7\} & \{1,2\} \\ U & U & \{9\} \end{bmatrix}, \begin{bmatrix} \{4,10\} & \{4\} & U \\ U & \{3,5\} & \{3\} \\ U & U & \{6,1\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{4\} & U \\ U & \{3,5\} & \{5\} \\ U & U & \{5,6\} \end{bmatrix}$$

4 阶子阵:

$$\begin{bmatrix} \{4,10\} & \{10\} & \{4\} & U \\ U & \{1,2,7\} & \{2,7\} & U \\ U & U & \{3,5\} & \{5\} \\ U & U & U & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & U & U \\ U & \{1,2,7\} & \{1,7\} & \{1,2\} \\ U & U & \{6,1\} & \{1\} \\ U & U & U & \{9\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & U & U \\ U & \{1,2,7\} & \{1,7\} & U \\ U & U & \{6,1\} & \{6\} \\ U & U & U & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{4\} & U & U \\ U & \{3,5\} & U & \{5\} \\ U & U & \{9\} & U \\ U & U & \{6\} & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{4\} & U & U \\ U & \{3,5\} & \{3\} & U \\ U & U & \{6,1\} & \{1\} \\ U & U & U & \{9\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{4\} & U & U \\ U & \{3,5\} & \{3\} & \{5\} \\ U & U & \{6,1\} & \{6\} \\ U & U & U & \{5,6\} \end{bmatrix}$$

5 阶子阵:

$$\begin{bmatrix} \{4,10\} & \{10\} & \{4\} & U & U \\ U & \{1,2,7\} & \{2,7\} & \{1,7\} & U \\ U & U & \{3,5\} & \{3\} & U \\ U & U & U & \{6,1\} & \{1\} \\ U & U & U & U & \{9\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & \{4\} & U & U \\ U & \{1,2,7\} & \{2,7\} & \{1,2\} & U \\ U & U & \{3,5\} & U & \{5\} \\ U & U & U & \{9\} & U \\ U & U & U & \{6\} & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & U & U & U \\ U & \{1,2,7\} & \{1,7\} & \{1,2\} & U \\ U & U & \{6,1\} & \{1\} & \{6\} \\ U & U & U & \{9\} & U \\ U & U & U & \{6\} & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{4\} & U & U & U \\ U & \{3,5\} & \{3\} & U & \{5\} \\ U & U & \{6,1\} & \{1\} & \{6\} \\ U & U & U & \{9\} & U \\ U & U & U & \{6\} & \{5,6\} \end{bmatrix}$$

$$\begin{bmatrix} \{4,10\} & \{10\} & \{4\} & U & U \\ U & \{1,2,7\} & \{2,7\} & \{1,7\} & U \\ U & U & \{3,5\} & \{3\} & \{5\} \\ U & U & U & \{6,1\} & \{6\} \\ U & U & U & U & \{5,6\} \end{bmatrix}$$

6 阶子阵:A。

从而得到的割集为:

$M^1(1)=\{4,10\}$

$M^2(1)=\{10,1,2,7\}$,$M^2(2)=\{4,3,5\}$

$M^3(1)=\{2,7,3,5\}$,$M^3(2)=\{10,1,7,6,1\}$,$M^3(3)=\{10,1,2,9\}$,$M^3(4)=\{4,3,6,1\}$,$M^3(5)=\{4,5,5,6\}$

$M^4(1)=\{2,7,5,5,6\}$,$M^4(2)=\{10,1,1,9\}$,$M^4(3)=\{10,1,7,6,5,6\}$,$M^4(4)=\{4,5,9,6\}$,$M^4(5)=\{4,3,1,9\}$,$M^4(6)=\{4,6,5,6\}$

$M^5(1)=\{7,3,1,9\}$,$M^5(2)=\{2,5,9,6\}$,$M^5(3)=\{10,1,9,6\}$,$M^5(4)=\{4,9,6\}$,$M^5(5)=\{7,6,5,6\}$

再者,由算法 2,可以求出所有的极小割分别为:

$\min F(M(1))=F\{10,4\}=14$

$\min F(M(2))=F\{4,3,5\}=12$

$\min F(M(3))=F\{4,3,6,1\}=14$

$\min F(M(4))=F\{4,3,1,9\}=17$

$\min F(M(5))=F\{3,1,9\}=13$

$\min F(M(1))=F\{9,6\}=15$

于是得到 $F=\min F_i=12$,即最大流为 12。

由最小割集 $R=\{3,4,5\}$ 可知,只要保证 v_1v_2,v_3v_4,v_3v_6 弧上的流值达到其最大容量,网络的流值即可达到最大流 12。

结束语 网络最大流问题的研究和应用对于当今社会是一项非常有意义的工作。本文从网络的部分割容量矩阵出发,通过集合的并与交运算,非常直观地找出了网络的所有割集,使得最大流最小割定理的实用价值进一步提高。算法实例结果表明,此方法是有效的。

算法 1 中的子阵取法仅考虑 S 连通与否,因此对于非整数流值网络,此算法也是有效的。

参 考 文 献

[1] Ahujia R K, Magnanti T L, Orlin J B. Network flows: Theory, algorithms and application[M]. New Jersey: Rentice Hall, 1993

[2] Korte B, Vygen J. Combinatorial optimization theory and algorithms[M]. Berlin, Heidelberg: Springer-Verlag, 2000

[3] Bondy J A, Murty U S R. Graph Theory with Applications[M]. New York: Elsevier Science publishing co. inc., 1976

(下转第 246 页)

文的特性，而且在模型层上建模了上下文的性质。通过在ASCCD语料库上的实验说明，本方法能够获得90.34%的检测准确率，比基线系统高了6.09%。今后，我们将进一步探索用于韵律间断检测的特征和建模方法，并利用本互补模型方法对实际的连续语音进行韵律间断的检测。

参考文献

[1] Wightman C W, Ostendorf M. Automatic labeling of prosodic patterns [J]. IEEE Trans. on Audio and Speech Processing, 1994, 2(4): 469-481

[2] Ananthakrishnan S, Narayanan S S. An Automatic Prosody Recognizer Using a Coupled Multi-stream Acoustic Model and a Syntactic-Prosodic Language Model[C]// IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing. 2005: 269-272

[3] Ananthakrishnan S, Narayanan S. Automatic prosodic even detection using acoustic, lexical and syntactic evidence [J]. IEEE Trans. on Audio, Speech, and Language Processing, 2008, 16 (1): 216-228

[4] Ananthakrishnan S, Narayanan S. Fine-grained Pitch Accent and Boundary Tone Labeling with Parametric F0 Features[C]// IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing. 2008

[5] Sridhar V K R, et al. Exploiting Acoustic and Syntatic Features for Automatic Prosody Labeling in a Maximum Entropy Framework[J]. IEEE Trans. Audio, Speech, and Language Processing, 2008, 16(4): 797-811

[6] Hasegawa-Johnson M, Chen K, Cole J, et al. Simultaneous Recognition of Words and Prosody in the Boston University Radio Speech Corpus[J]. Speech Communication, 2005, 46(3/4): 418-439

[7] Jeon J H, Liu Yang. Automatic Prosodic Events Detection Using Syllable-based Acoustic and Syntactic Features[C]// IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing. 2009: 4565-4568

[8] Hu W, Huang T, Xu B. Study on Prosodic Boundary Location in Chinese Mandarin[C]// IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. 2002: I-501-I-504

[9] 邵艳秋，韩纪庆，刘挺，等. 自然风格言语的汉语句重音自动判别研究[J]. 声学学报，2006，31(3): 203-210

[10] Chen Xiao-xia, Li Ai-jun, Sun Guo-hua, et al. An Application of SAMPA-C in Standard Chinese[C]// Proc. ICSLP. Beijing, 2000

[11] Li Ai-jun. Chinese Prosody and Prosodic Labeling of Spontaneous Speech[C]// Speech Prosody 2002. Aix-en-Provence

[12] 倪崇嘉，刘文举，徐波. 汉语韵律短语的时长与音高研究[J]. 中文信息学报，2009，23(4): 82-87

[13] Boersma P, Weenik D. Praat: doing phonetics by computer[CP/OL]. http://www.praat.org

[14] Tseng H, Chang P, Andrew G, et al. A Conditional Random Field Word Segmenter[C]// Fourth SIGHAN Workshop on Chinese Language Processing. 2005

[15] Chang Pi-chuan, Galley M, Manning C. Optimizing Chinese Word Segmentation for Machine Translation Performance[C]// ACL Third Workshop on Statistical Machine Translation. 2008

[16] The Stanford Postagger [CP/OL]. http://nlp.stanford.edu/software/tagger.shtml

[17] Sun Xue-jing. Pitch Accent Prediction Using Ensemble Machine Learning[C]// Proceedings of the International Conference on Spoken Language Processing. 2002: 963-956

[18] Freund Y, Schapire R E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting[J]. Journal of Computer and System Sciences, 1997, 55(1): 119-139

[19] Lafferty J D, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]// Proc. International Conference on Machine Learning (ICML'01). 2001: 282-289

[20] Garner S. Weka: the waikato environment for knowledge analysis[C]// Proc. the New Zealand Computer Science Research Students Conf. 1995

[21] LIBSVM-A Library for Support Vector Machines [CP/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[22] CRF++: Yet Another CRF toolkit[CP]. http://crfpp.sourceforge.net/

(上接第231页)

[4] Ahujia R K, Orlin J B. Distance-directed augmenting path algorithms for the maximum flow problem[J]. Naval Research Logistics Quarterly, 1991, 38: 413-430

[5] Glodberg A V, Rao S. Beyond the flow decomposition barrier [J]. J Assoc Comput Mach, 1988, 45(5): 783-797

[6] 邹豪思，王志远. 网络最大流的矩阵算法[J]. 内蒙古大学学报，2001，32(4): 465-469

[7] 孙泽宇，丁国强，程志谦. 网络最大流求解算法的研究[J]. 微计算机信息，2010(3)

[8] 党耀国，刘思峰，方志耕. 网络最大流的割集矩阵算法[J]. 系统工程理论与实践，2003，9: 125-128

[9] Tutte W T. Graph Theory [M]. Cambridge: Cambridge university press, 2001

[10] 高随祥. 图论与网络流理论[M]. 北京：高等教育出版社，2009