



## 包含交叉顶点的最大流改进算法

罗甜甜, 赵礼峰

(南京邮电大学 理学院, 南京 210023)

**摘 要:** 最短增广链算法构建分层剩余网络后, 在面临多条相同弧数增广链且其中顶点有重合的情况下, 会因寻找增广链时未考虑增广顺序而导致流值丢失。针对该问题, 提出一种网络图中包含交叉顶点的最大流改进算法。该算法保留最短增广链算法的分层理念, 仍在分层剩余网络中寻找增广链, 在此基础上增加寻找增广链的规则, 即优先搜索与源点关联且容差最小的顶点作为下一步推进点, 确定一条增广链后即考虑与上一条有重合的顶点所在的增广链进行增广。实例分析与 BA 无标度网络建模仿真结果表明, 与最短增广链算法相比, 该算法得到的最大流值更准确, 并且效率相当。

**关键词:** 最大流; 分层剩余网络; 交叉顶点; 顶点容差; BA 无标度网络

开放科学(资源服务)标志码(OSID):



中文引用格式: 罗甜甜, 赵礼峰. 包含交叉顶点的最大流改进算法[J]. 计算机工程, 2020, 46(11): 48-52.

英文引用格式: LUO Tiantian, ZHAO Lifeng. Improved maximum flow algorithm with intersecting vertices[J]. Computer Engineering, 2020, 46(11): 48-52.

## Improved Maximum Flow Algorithm with Intersecting Vertices

LUO Tiantian, ZHAO Lifeng

(School of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

**[Abstract]** A hierarchical residual network constructed by the shortest augmented chain algorithm may lead to the loss of flow values when it faces multiple augmented chains with the same number of directed graphs and overlapped vertices, for it does not consider the augmented sequence when looking for augmented chains. In order to solve the problem, this paper proposes an improved network maximum flow algorithm with intersecting vertices in the network graph. It retains the hierarchical concept of the shortest augmented chain algorithm, and looks for augmented chains in the hierarchical residual network. On this basis, a rule is added to prioritize the search of the vertices that are related to the source vertex and with the minimum tolerance, and the vertices are taken as the new source vertices for further search. After an augmented chain is determined, the augmented chains with overlapped vertices are considered for further augmentation with the previous augmented chain. Results of example analysis and BA scale-free network modeling simulation show that, compared with the shortest augmented chain algorithm, the proposed algorithm can obtain more accurate maximum flow values while keeping same efficiency.

**[Key words]** maximum flow; hierarchical residual network; intersecting vertice; vertice tolerance; BA scale-free network

DOI: 10.19678/j.issn.1000-3428.0056170

### 0 概述

网络最大流问题紧密结合图论与运筹学, 为图论的应用开辟了新途径。解决最大流问题即讨论如何利用装置有限的容量来实现运输流量最大。针对这一问题, 很多学者提出了相应的求解算法, 主要可分为增广链算法和预流推进算法两类。经典增

广链算法有 Ford-Fulkerson 算法<sup>[1]</sup>、Edmonds-karp 算法<sup>[2]</sup>和最短增广链算法<sup>[3]</sup>, 经典预流推进算法有阻塞流算法<sup>[4]</sup>和推进重标号算法<sup>[5]</sup>。许多学者对网络最大流问题也做了进一步深入的研究<sup>[6]</sup>, 针对经典算法提出了多种改进算法<sup>[7-11]</sup>, 如利用容差进行修正的最大流 2F 算法<sup>[7]</sup>、有条件限制的最大流求解算法<sup>[9]</sup>、动态网络的最大流增量算法<sup>[10]</sup>及大规模网络

基金项目: 国家自然科学基金(61304169)。

作者简介: 罗甜甜(1994—), 女, 硕士研究生, 主研方向为最大流算法; 赵礼峰, 教授。

收稿日期: 2019-09-30 修回日期: 2019-11-28 E-mail: 2712261748@qq.com

的最大流求解算法<sup>[11]</sup>。文献[12-13]则提出了最短增广链改进算法。

最短增广链算法构建的分层剩余网络  $AD(f)$ , 使网络中的  $(v_s, v_t)$  路始终是剩余网络  $D(f)$  中的最短  $(v_s, v_t)$  路。相比  $D(f)$ , 在  $AD(f)$  中更易寻找  $(v_s, v_t)$  路, 但有时  $AD(f)$  中含有交叉顶点, 如果仍然任意选择增广链而不考虑增广先后顺序, 则可能会导致之后构建的  $AD(f)$  中某些增广链缺失。

为提高最短增广链算法的准确性, 本文提出一种基于交叉顶点的最大流改进算法。在分层剩余网络  $AD(f)$  中依次选择顶点容差较小的顶点, 以下一步推进的原则找到第一条增广链, 在此基础上寻找与该条增广链有交叉顶点的相关增广链, 从而避免在  $AD(f)$  中选择增广链的任意性, 提高算法效率。

## 1 基本概念

### 定义1 容量网络

给定一个有向图  $G(V, A)$ ,  $c$  是定义在  $A$  上的正函数,  $\forall a \in A, a = (v_i, v_j), c(a) = c_{ij}$ , 其中,  $V, A, c$  分别表示该有向图的顶点集、弧集和弧上的容量值, 称网络  $D = (V, A, c)$  为容量网络<sup>[14]</sup>。

### 定义2 剩余网络

剩余网络为  $D(f) = (V, A(f), c_f)$ , 其中,  $f$  为容量网络  $D = (V, A, c)$  上的可行流。定义:

$$A^+(f) = \{(v_i, v_j) | (v_i, v_j) \in A, f_{ij} < c_{ij}\}$$

$$A^-(f) = \{(v_i, v_j) | (v_j, v_i) \in A, f_{ji} > 0\}$$

记  $A(f) = A^+(f) \cup A^-(f)$ , 对  $\forall (v_i, v_j) \in A(f)$ , 令:

$$c_{ij}(f) = \begin{cases} c_{ij} - f_{ij}, & (v_i, v_j) \in A^+(f) \\ f_{ji}, & (v_i, v_j) \in A^-(f) \end{cases}$$

称  $c_{ij}(f)$  为剩余网络  $D(f)$  的剩余容量<sup>[14-15]</sup>。

广探法即广度优先搜索<sup>[16-18]</sup>, 是构造分层剩余网络的基础方法, 具体步骤如下:

**步骤1** 在含有源点  $v_s$  和汇点  $v_t$  的容量网络  $D = (V, A, c)$  中, 源点  $v_s$  记为标号未检查, 令  $h(v_s) = 0, l_s = -1$ 。

**步骤2** 若存在已标号未检查的顶点, 找到最先标号的顶点  $v_i$ , 转步骤3; 若所有已标号顶点都检查完, 转步骤4。

**步骤3** 考察  $v_i$  的所有出弧  $(v_i, v_j)$ , 若  $v_j$  未标号, 则将  $v_j$  记为已标号未检查, 令  $h(v_j) = h(v_i) + 1, l_j = i$ ; 若  $v_j$  已标号则不作处理; 若  $v_i$  的所有出弧全部考察完, 则顶点  $v_i$  记为已检查, 转步骤2。

**步骤4** 算法终止,  $h(v_t)$  为容量网络  $D = (V, A, c)$  中最短  $(v_s, v_t)$  路的长度。

### 定义3 分层剩余网络

根据剩余网络和广探法, 可以给出剩余网络的一个子网络, 即分层剩余网络  $AD(f)$ <sup>[14-15]</sup>, 其定义如下:

$$AD(f) = (V'(f), A'(f), c_f)$$

$$V'(f) = \{v_t\} \cup \{v_i \in V | h(v_i) < h(v_t)\}$$

$$A'(f) = \{(v_i, v_j) \in A(f) | h(v_j) = h(v_i) + 1 < h(v_t)\} \cup \{(v_i, v_i) \in A(f) | h(v_i) = h(v_t) - 1\}$$

### 定义4 顶点容差

顶点  $v(v \in V)$  的所有出弧容量减去该顶点的所有入弧容量的值为顶点  $v(v \in V)$  的容差。

### 定义5 交叉顶点

在分层剩余网络  $AD(f)$  中, 除源点与汇点之外, 如果存在一个顶点包含在两条及两条以上的增广链中, 则称该点为交叉顶点。

### 定义6 交叉顶点原则

在含有交叉顶点的网络图中, 选择增广链时优先搜索与源点关联且容差最小的顶点作为增广链的下一步推进点。确定一条增广链后, 对与上一条有重复顶点所在的增广链进行增广。

### 定义7 顶点的度

设  $M = (V, A)$  为一个非空有向图, 顶点  $v \in V$ , 与顶点  $v$  所关联的弧的数目即为顶点的度<sup>[18]</sup>。

BA 无标度网络是被用来模拟不断增长的网络顶点的无标度网络模型<sup>[19-20]</sup>, 其创建过程如下:

1) 开始于一个包括  $m_0$  个顶点的网络, 每新增一个顶点, 都要相应地连接到  $m(m < m_0)$  个已有顶点上。

2) 已有顶点  $i$  与新顶点连接的概率为  $p_{v_i} = k_{v_i} / \sum_j k_{v_j}$ ,

其中,  $k_{v_i}, k_{v_j}$  分别表示顶点  $v_i$  与顶点  $v_j$  的度。

## 2 基于交叉顶点的最大流算法

### 2.1 算法设计思想

虽然最短增广链算法构建了分层剩余网络, 使得  $AD(f)$  的  $(v_s, v_t)$  路都是  $D(f)$  中的最短  $(v_s, v_t)$  路, 简化了网络, 但在  $AD(f)$  中仍存在多条增广链的可能, 而且有时几条增广链中会有一些重合顶点存在, 如果在这种情况下依然任意选择  $AD(f)$  中的增广链, 可能会造成某些增广链的缺失, 使得最终的最大流值偏小。针对这一情况, 本文对最短增广链算法寻找增广链的过程进行改进, 不再对  $AD(f)$  中多条增广链采取视同一律的处理方式, 而是根据交叉顶点原则来寻找增广链。

### 2.2 算法步骤

**初始化** 在容量网络  $D$  中取初始可行流  $f_k$  (一般取零流作为初始可行流), 令  $k = 1$ 。

**步骤1** 构造  $D$  关于可行流  $f_k$  的剩余网络  $D(f_k)$ , 并对该网络应用广探法构建分层剩余网络  $AD(f_k)$ 。如果在构建  $AD(f_k)$  时汇点  $v_t$  得不到标号, 则算法结束,  $f_k$  即为网络最大流; 否则转步骤2。

**步骤2** 在  $AD(f_k)$  中根据交叉顶点原则寻找增广链, 即  $(v_s, v_t)$  路  $p$ , 转步骤3; 若不存在  $(v_s, v_t)$  路,

则令  $f_{k+1} = f_k, k = k + 1$ , 转步骤 1。

**步骤 3** 沿着增广链  $p$  对  $f_k$  进行增广, 增广流值  $\delta = \min_{(v_i, v_j) \in A(p)} \{c_{ij}(f_k)\}$  且对  $\forall (v_i, v_j) \in A(p)$ , 令  $c_{ij}(f_k) = c_{ij}(f_k) - \delta$  并删去剩余容量为 0 的弧, 得到新的网络, 仍记为  $AD(f_k)$ , 转步骤 2。

### 3 算法可行性与复杂度分析

#### 3.1 算法可行性分析

本文算法是在最短增广链算法基础上的改进, 改进之处在于寻找增广链时按照交叉顶点原则寻找, 每次找到增广链进行增广后, 至少会使容量网络中的一条弧成为饱和弧。设容量网络  $D = (V, A, c)$  中共有  $m$  条弧, 若至多经过  $m$  次增广后  $D$  中不存在源点到汇点的增广链, 即在构建分层剩余网络时汇点得不到标号, 则满足算法的终止条件, 算法在有限步内即可停止, 不会陷入无限循环。

#### 3.2 算法复杂度分析

设容量网络  $D = (V, A, c)$ , 其顶点数为  $n$ , 弧数为  $m$ 。本文算法整体分为两部分, 即构建分层剩余网络和寻找增广链。由于步骤 1 中构建分层剩余网络  $AD(f_k)$  的层数随  $k$  单调递增,  $AD(f_k)$  的最高层数至多是  $(n - 1)$  层, 因此算法中最多建立  $n$  个分层剩余网络。

由广探法可知, 每构建一个分层剩余网络的复杂度为  $O(m)$ , 因此, 本文算法构建分层剩余网络的总复杂度为  $O(nm)$ 。

在一个分层剩余网络中, 最多有  $m$  条弧, 由于每增广 1 次至少删去 1 条弧, 因此至多增广  $m$  次, 而每次增广的计算量为  $O(n)$ , 所以, 在一个分层剩余网络中寻找增广链的复杂度为  $O(mn)$ , 从而寻找增广链的总复杂度为  $O(n^2m)$ 。

综上所述, 本文算法的复杂度为:

$$O(nm) + O(n^2m) = O(n^2m)$$

### 4 算法实例与比较

给出以下实例: 在容量网络  $D$  中, 求从源点  $v_s$  到汇点  $v_t$  的最大流, 如图 1 所示, 其中, 弧边的数值表示该弧的容量大小, 下同。

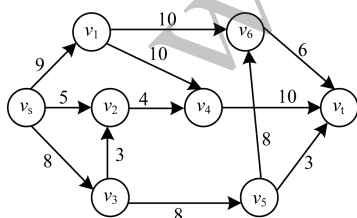


图 1 容量网络  $D$

Fig. 1 Capacity network  $D$

当容量网络中存在多条源点到汇点的路径, 且这些路径之间包含相同顶点时, 分别应用本文算法

和最短增广链算法进行求解。

本文算法求解过程如下:

1) 取零流  $f_1$  作为初始可行流, 得到剩余网络  $D(f_1)$ , 利用广探法构造分层剩余网络  $AD(f_1)$ , 见图 2。

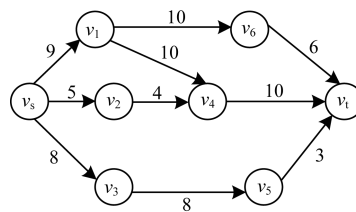


图 2 分层剩余网络  $AD(f_1)$

Fig. 2 Hierarchical residual network  $AD(f_1)$

2) 在  $AD(f_1)$  中计算与  $v_s$  相关联顶点的容差, 并将容差值标在相应顶点的旁边。选择容差值最小的顶点作为下一步推进点增广, 得到增广链  $p_1 = v_s v_2 v_4 v_t$ , 可行流  $f_2$  见图 3, 其中, 弧边的数值表示该弧的流量大小, 下同。

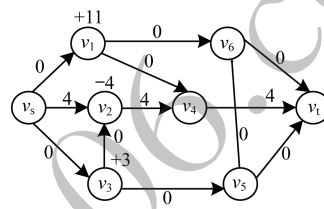


图 3 可行流  $f_2$

Fig. 3 Feasible flow  $f_2$

3) 在  $AD(f_2)$  中继续选择与  $p_1$  有交叉顶点的增广链进行增广, 得到增广链  $p_2 = v_s v_1 v_4 v_t$ , 可行流  $f_3$  见图 4。

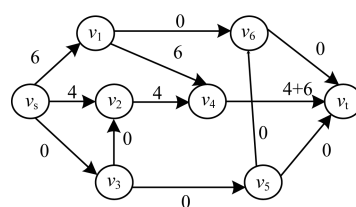


图 4 可行流  $f_3$

Fig. 4 Feasible flow  $f_3$

4) 在  $AD(f_3)$  中继续选择与  $p_2$  有交叉顶点的增广链进行增广, 得到增广链  $p_3 = v_s v_1 v_6 v_t$ , 可行流  $f_4$  见图 5。

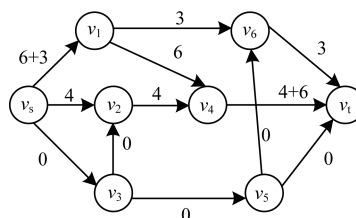
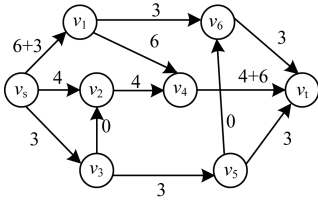


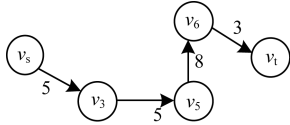
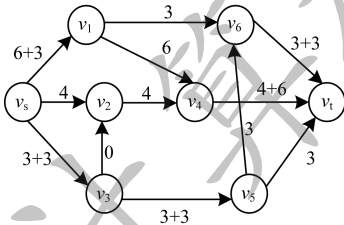
图 5 可行流  $f_4$

Fig. 5 Feasible flow  $f_4$

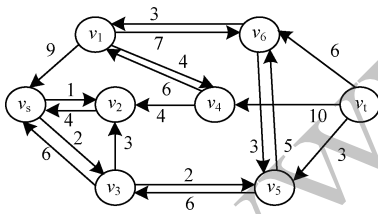
5) 此时,在  $AD(f_4)$  中没有与  $p_1 \sim p_3$  有交叉顶点的增广链,只剩下增广链  $p_4 = v_s v_3 v_5 v_t$ , 由此得到可行流  $f_5$ , 见图 6。

图 6 可行流  $f_5$ Fig. 6 Feasible flow  $f_5$ 

6) 此时,在  $AD(f_5)$  中没有  $(v_s, v_t)$  路, 令  $f_5 = f_6$ , 重新构造分层剩余网络  $AD(f_6)$ , 见图 7, 并在其中找到增广链  $p_5 = v_s v_3 v_5 v_6 v_t$ , 可行流  $f_7$  见图 8。

图 7 分层剩余网络  $AD(f_6)$ Fig. 7 Hierarchical residual network  $AD(f_6)$ 图 8 可行流  $f_7$ Fig. 8 Feasible flow  $f_7$ 

7) 此时,  $AD(f_7)$  中没有  $(v_s, v_t)$  路, 令  $f_7 = f_8$ , 构建剩余网络  $D(f_8)$ , 见图 9。由于  $D(f_8)$  中不存在  $(v_s, v_t)$  路, 因此  $f_8$  为容量网络  $D$  的最大流, 流值为 19。

图 9 剩余网络  $D(f_8)$ Fig. 9 Residual network  $D(f_8)$ 

最短增广链算法求解过程如下:

1) 取零流  $f_1$  作为初始可行流, 构建分层剩余网络  $AD(f_1)$ , 见图 2。

2) 在图 2 中找增广链, 找到增广链  $p_1 \sim p_4$ :

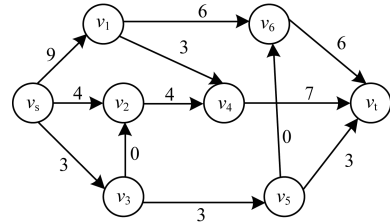
$$p_1 = v_s v_1 v_6 v_t, \delta_1 = 6$$

$$p_2 = v_s v_1 v_4 v_t, \delta_2 = 3$$

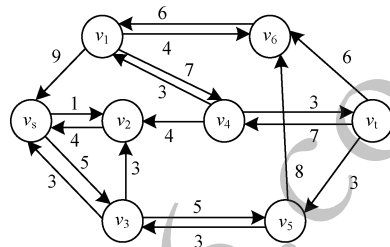
$$p_3 = v_s v_2 v_4 v_t, \delta_3 = 4$$

$$p_4 = v_s v_3 v_5 v_t, \delta_4 = 3$$

增广之后, 得到可行流  $f'_2$ , 见图 10。

图 10 可行流  $f'_2$ Fig. 10 Feasible flow  $f'_2$ 

3) 继续构建剩余网络  $D(f'_2)$ , 见图 11。发现  $D(f'_2)$  中不存在  $(v_s, v_t)$  路, 算法结束。 $f'_2$  记为容量网络  $D$  的最大流, 流值为 16。

图 11 剩余网络  $D(f'_2)$ Fig. 11 Residual network  $D(f'_2)$ 

可以看出, 最短增广链算法在分层剩余网络中选取增广链的任意性会导致增广链缺失, 从而使最后得到的最大流值偏小。

## 5 仿真与结果分析

### 5.1 实验环境与设计

本文使用的实验仿真平台是 MATLAB2016a, 处理器为 Intel® Core™ i3-4030U CPU @ 1.90 GHz, 内存 4 GB, Windows7 版本 64 位操作系统。

仿真实验采用的是 BA 无标度随机网络, 将网络规模设为顶点数为 500、1 000、1 500、2 000、2 500、3 000、3 500。在给定的网络规模下, 对本文算法和最短增广链算法进行 10 次仿真实验。其中, 参数  $f_1$  和  $t_1$  分别表示最短增广链算法的最大流值及其运行时间的平均值, 参数  $f_2$  和  $t_2$  分别表示本文算法的最大流值及其运行时间的平均值。

### 5.2 实验结果分析

实验结果如表 1 所示, 其中列出了 2 种算法在不同网络规模下得到的最大流值及算法的平均运行时间。可以看出, 本文算法较最短增广链算法结果更精准, 并且运行时间相差较小。

表 1 本文算法与最短增广链算法在不同网络规模下的实验结果  
Table 1 Experimental results of the proposed algorithm and the shortest extended chain algorithm in different network sizes

顶点数量	最大流值		平均运行时间/s	
	$f_1$	$f_2$	$t_1$	$t_2$
500	22	118	0.106 3	0.089 9
1 000	74	140	0.075 2	0.077 2
1 500	25	25	0.058 7	0.062 3
2 000	35	96	0.059 9	0.061 5
2 500	71	71	0.079 3	0.083 9
3 000	39	133	0.105 5	0.087 4
3 500	62	165	0.132 6	0.141 5

## 6 结束语

最短增广链算法在分层剩余网络中寻找增广链的任意性可能导致增广链缺失。针对该问题,本文提出一种基于交叉顶点的改进算法。在寻找增广链时不再视同一律,而是遵循顶点交叉原则,从而避免增广链缺失的情况,提高算法的准确性。实验结果表明,该算法得到的最大流值比最短增广链算法更准确,并且运行效率相当。后续将在保证准确率的基础上,进一步提高算法的运行效率。

### 参考文献

- [1] FORD L R, FULKERSON D R. Maximum flow through a network[J]. Canadian Journal of Mathematics, 1956, 8(5): 399-404.
- [2] EDMONDS J, KARP R M. Theoretical improvements in algorithmic efficiency for networks flow problems[J]. Journal of the ACM, 1972, 19(2): 248-264.
- [3] DINIC E A. Algorithm for solution of a problem of maximum flow in networks with power estimation[J]. Soviet Mathematics Doklady, 1970, 11(8): 1277-1280.
- [4] KARZANOV A V. Determining the maximum flow in a network by the method of preflows[J]. Soviet Mathematics Doklady, 1974, 15(3): 434-437.
- [5] GOLDBERG A V, TARJAN R E. A new approach to the maximum flow problem[J]. Journal of the ACM, 1988, 35(4): 921-940.
- [6] ZHANG Xianchao, CHEN Guoliang, WAN Yingyu. Research progress on the problem of network maximum flow[J]. Journal of Computer Research and Development, 2003, 40(9): 1281-1292. (in Chinese)  
张宪超, 陈国良, 万颖瑜. 网络最大流问题研究进展[J]. 计算机研究与发展, 2003, 40(9): 1281-1292.
- [7] CHEN Jing, SHAN Rui. Tolerance correction network maximum flow 2F algorithm[J]. Journal of Changchun University of Technology (Natural Science Edition), 2008, 29(6): 713-716. (in Chinese)  
陈静, 单锐. 容差修正网络最大流 2F 算法[J]. 长春工业大学学报(自然科学版), 2008, 29(6): 713-716.
- [8] DENG Guoqiang, TANG Min, CHEN Guangxi. An improved algorithm for maximum flow problem[C]//Proceedings of International Conference on Communications, Circuits and Systems. Washington D. C., USA: IEEE Press, 2009: 1-5.
- [9] CALISKAN C. A specialized network simplex algorithm for the constrained maximum flow problem[J]. European Journal of Operational Research, 2010, 210(2): 137-147.
- [10] ZHANG Baili, WANG Yuanyuan, HONG Liang, et al. Fast incremental solution of maximum flow in dynamic networks[J]. Journal of Southeast University (Natural Science Edition), 2017, 47(3): 450-455. (in Chinese)  
张柏礼, 王媛媛, 洪亮, 等. 动态网络中最大流快速增量求解[J]. 东南大学学报(自然科学版), 2017, 47(3): 450-455.
- [11] WEI Huazhen, ZHAO Shu, CHEN Jie, et al. Solution of maximum flow in large-scale networks based on label propagation[J]. Computer Science and Exploration, 2017, 11(10): 1609-1620. (in Chinese)  
魏华珍, 赵姝, 陈洁, 等. 基于标签传播的大规模网络最大流求解方法[J]. 计算机科学与探索, 2017, 11(10): 1609-1620.
- [12] ZHAO Lifeng, YAN Ziheng. Maximum flow algorithm based on augmented chain repair[J]. Journal of Computer Applications, 2015, 35(5): 1246-1249. (in Chinese)  
赵礼峰, 严子恒. 基于增广链修复的最大流求解算法[J]. 计算机应用, 2015, 35(5): 1246-1249.
- [13] ZHAO Lifeng, JI Yabao. Improved shortest augmented chain algorithm for maximum flow problem[J]. Computer Technology and Development, 2016, 26(8): 52-54, 59. (in Chinese)  
赵礼峰, 纪亚宝. 最大流问题的改进最短增广链算法[J]. 计算机技术与发展, 2016, 26(8): 52-54, 59.
- [14] XIE Zheng, LI Jianping. Network algorithms and complexity theory[M]. Changsha: National University of Defense Technology Press, 2003. (in Chinese)  
谢政, 李建平. 网络算法与复杂性理论[M]. 长沙: 国防科技大学出版社, 2003.
- [15] LI Mingzhe, JIN Jun, SHI Duanyin. Graph theory and algorithms[M]. Beijing: Machinery Industry Press, 2010. (in Chinese)  
李明哲, 金俊, 石端银. 图论及其算法[M]. 北京: 机械工业出版社, 2010.
- [16] YANG Zhiming. Analysis and implementation of breadth first search traversal algorithm for graphs[J]. Agricultural Network Information, 2009(12): 136-137. (in Chinese)  
杨智明. 图的广度优先搜索遍历算法的分析与实现[J]. 农业网络信息, 2009(12): 136-137.
- [17] WANG Haiying, HUANG Qiang, LI Chuantao, et al. Graph theory algorithm and its MATLAB implementation[M]. Beijing: Beihang University Press, 2010. (in Chinese)  
王海英, 黄强, 李传涛, 等. 图论算法及其 MATLAB 实现[M]. 北京: 北京航空航天大学出版社, 2010.
- [18] HAO Rongxia. Guide to graph theory[M]. Beijing: Beijing Jiaotong University Press, 2014. (in Chinese)  
郝荣霞. 图论导引[M]. 北京: 北京交通大学出版社, 2014.
- [19] WANG Xiaomin. The relationship between the construction of scale-free network models and mathematical methods[D]. Lanzhou: Northwest Normal University, 2017. (in Chinese)  
王晓敏. 无标度网络模型的构建和数学方法间的联系[D]. 兰州: 西北师范大学, 2017.
- [20] ALBERT B. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509-512.