

# 基于宽度优先的网络最大流求解算法

邵丽萍,赵礼峰

(南京邮电大学 理学院,江苏 南京 210023)

**摘要:**网络最大流问题是经典的组合优化问题,为了降低求解大规模网络最大流的计算量,若用 Ford-Fulkerson 算法寻找增广链,则效率不高且步骤繁杂。为了改善以上不足,在原有算法的基础上作了一些改进,应用图的宽度优先搜索原理,针对单源单汇网络提出了一种新的求解最大流问题的算法。该算法的思想是:用宽度优先搜索原理,寻找一条包含剩余容量最大的弧的最短增广链后,删除饱和弧,且沿合适的路径修复包含剩余容量最大的弧的最短增广链。该算法避免了 Ford-Fulkerson 算法的标号过程,减少了反复重新寻找增广链的次数,为在大规模网络中快速获取最大流的求解提供了方便并提高了求解网络最大流的执行效率。通过实例分析与 BA 无标度网络建模仿真,验证了该算法的实用性,且新算法的运行效率高于 Ford-Fulkerson 算法。

**关键词:**最大流;剩余网络;增广链修复;宽度优先搜索;BA 无标度网络

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2019)06-0062-04

**doi:**10.3969/j.issn.1673-629X.2019.06.013

## An Algorithm for Solving Maximum Flow Based on Breadth First Search

SHAO Li-ping, ZHAO Li-feng

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

**Abstract:** The network maximum flow is a classic combinatorial optimization problem. In order to reduce the computational complexity of solving the maximum flow of a large-scale network, if the Ford-Fulkerson algorithm is used to find the augmented chain, the efficiency is not high and the steps are complicated. In order to improve the above deficiencies, we make some improvements on the basis of the original algorithm. Applying the breadth-first search principle, we propose a new algorithm for solving the maximum flow problem for single-source and single-sink networks. The idea of the algorithm is to use the breadth first search principle to find a shortest augmented chain containing the arc with the largest remaining capacity, then remove the saturated arc, repair the shortest augmented chain containing the arc with the largest remaining capacity along the appropriate path. The algorithm avoids the labeling process of the Ford-Fulkerson algorithm, reduces the number of repeated re-finding of the augmented chain, provides convenience for solving the maximum stream in a large-scale network, and improves the execution efficiency of solving the maximum flow of the network. The practicality of the algorithm is verified by an example analysis and experiments in the BA scale-free network, and it is more efficient than the Ford-Fulkerson algorithm.

**Key words:** maximum flow; remaining network; augmenting path restoration; breadth first search; BA scale-free network

## 0 引言

网络最大流问题是一个属于图论和运筹学的问题<sup>[1-3]</sup>。最初 Fulkerson 等<sup>[4]</sup>在处理最大流问题时,运用了图论的方法。后来,在 1956 年 Ford-Fulkerson 给出了求最大流问题的标号算法<sup>[5]</sup>; Dinic (1970), Edmonds 和 Karp (1972) 独立提出了改进 Ford-Fulkerson 算法的思想:每次都沿最短(即弧数最少的)

增广链进行增广,即最短增广链算法<sup>[6-7]</sup>; Karzanov (1974)提出了预流的概念,基于预流的概念,进一步提出了预流推进的算法; Cherkassky 提出了最高标号预流推进算法<sup>[8]</sup>。这些经典的算法是研究大规模网络的基础。在这些研究之后,很多学者针对特殊网络或经典算法的改进方面,提出了许多最大流问题的算法<sup>[9-16]</sup>。

收稿日期:2018-08-12

修回日期:2018-12-13

网络出版时间:2019-03-06

基金项目:国家自然科学基金青年基金项目(61304169)

作者简介:邵丽萍(1993-),女,硕士研究生,研究方向为网络最大流、最小费用流;赵礼峰,教授,硕导,研究方向为图论及其在通信中的应用。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190306.1017.092.html>

针对 Ford-Fulkerson 算法存在标号的繁琐、增广链寻找的任意性和运行效率慢的缺陷,文中提出了一种基于宽度优先的网络最大流求解算法。该算法利用宽度优先搜索原理,寻找一条包含剩余容量最大的弧的最短增广链,且删除饱和弧后,运用了修复最短增广链的方法<sup>[17]</sup>,简化了网络且避免了反复寻找新的增广链,达到了优化 Ford-Fulkerson 算法的目的。

## 1 基本概念

### 1.1 最大流的模型

定义 1<sup>[6]</sup>:定义一个含容量的网络,记为  $G = (V, A, c)$ , 定义流  $f_{st}$  为网络  $G = (V, A, c)$  中从始点  $s$  到终点  $t$  的流,如果  $f_{st}$  满足下列条件:

$$\sum_j f(i, j) - \sum_j f(j, i) = \begin{cases} f_{st}, & i = s \\ 0, & i \neq s, t \\ -f_{st}, & i = t \end{cases}$$

$$0 \leq f(i, j) \leq c(i, j), \forall (i, j) \in A(G)$$

则  $f_{st}$  是网络  $G$  的一个可行流。在所有满足条件的可行流中,流量最大的可行流被称为最大流,记作  $f_{\max}$ 。上式中满足  $f(i, j) < c(i, j)$  的弧为非饱和弧,满足  $f(i, j) = c(i, j)$  的弧为饱和弧。可行流的定义满足流量守恒,体现在两方面:一方面是始点  $s$  的所有流量全部到达终点  $t$ ,另一方面是始点  $s$  和终点  $t$  以外的所有节点只通过流量而不储存流量。

### 1.2 剩余网络(增量网络)

剩余网络<sup>[5-6]</sup>(residual network)是指在一张含可行流的容量网络图中所有非饱和弧及所有节点的集合,亦可用反向弧在网络中标记出饱和弧及当前流。记剩余网络为  $U(f) = (V, A(f), c(f))$ , 需满足两个关系:

(1)  $\forall (i, j) \in A(G)$ , 若  $f(i, j) < C_c(i, j)$ , 则  $C_u(i, j) = C_c(i, j) - f(i, j)$ ;

(2)  $\forall (i, j) \in A(G)$ , 若  $f(i, j) > 0$ , 则  $C_u(j, i) = f(i, j)$ 。

## 2 一种改进的最大流算法

### 2.1 算法思想

改进算法的主要思想是:找一条从始点  $v_s$  到终点  $v_t$  包含剩余容量最大的弧的最短增广链  $P$ , 然后对其不断修复,减少重新寻找可增广链的执行次数。首先,在容量网络  $G = (V, A, c)$  中,取任意一个可行流  $f$  (一般取零流)。在容量网络  $G = (V, A, c)$  中,使用宽度优先原则,搜索一条从始点  $v_s$  到终点  $v_t$  包含剩余容量最大的弧的最短增广链  $P$ , 然后在最短增广链  $P$  上删除饱和弧和调整相应弧的容量。进一步是增广链修复的

过程,搜索满足条件的节点,找到一条包含删除弧两个端点的新的增广链。具体修复的过程是:取距离始点  $v_s$  最近的删除弧  $(v_s, v_m)$  及距离终点  $v_t$  最近的删除弧  $(v_q, v_n)$ , 得到断点为  $v_s, v_n$ , 寻找包含  $v_s$  到  $v_n$  的一条路径,且修复的最短增广链包含剩余容量最大的弧,如图 1 所示。若找不到满足条件的节点,则意味着修复结束。在剩余容量网络中重新找一条包含剩余容量最大的弧的最短增广链,重复以上操作。若在剩余容量网络中,找不到从始点  $v_s$  到终点  $v_t$  包含剩余容量最大的弧的最短增广链时,则整个算法终止。

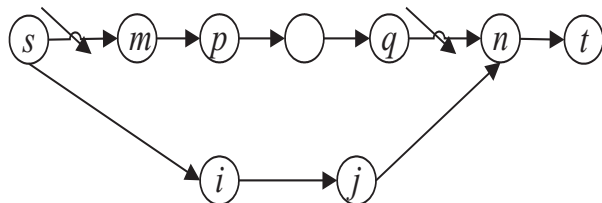


图 1 增广链的修复过程

### 2.2 算法步骤

在容量网络  $G = (V, A, c)$  中,开始于任意一个可行流  $f$  (一般取零流),执行以下步骤:

Step1: 从始点  $v_s$  出发,找一条从始点  $v_s$  到终点  $v_t$  包含剩余容量最大的弧的最短增广链  $P$  (宽度优先原则)。如果不存在,结束,  $f$  就是  $G$  的最大流;

Step2: 求出该最短增广链  $P$  上各弧容量的最小值  $\delta$ 。删除饱和弧,并在最短增广链  $P$  的各弧上减去  $\delta$ ;

Step3: 修复增广链  $P$ , 转 Step2, 如果不能修复,则转 Step1。

### 2.3 算法可行性分析

在新算法的操作过程中,删除包含剩余容量最大的弧的最短增广链上的饱和弧,对最大流的求解并没有影响。因为在以后执行算法的过程中,这条弧的容量不可能减少或增加<sup>[18]</sup>,只是简化了重新寻找可增广链的过程,且在剩余容量网络修复过程中,避免了重复搜索相关的弧。设容量网络  $G = (V, A, c)$  有  $m$  条弧,每次增广后,至少删除一条饱和弧,则最多经过  $m$  次增广后,网络  $G = (V, A, c)$  中就不存在从始点  $v_s$  到终点  $v_t$  的可增广链,此时算法终止,即新算法不会陷入死循环。

### 2.4 算法复杂度分析

在容量网络  $G = (V, A, c)$  中,设节点数为  $n$ , 弧数为  $m$ 。在 Step1 中执行一次宽度优先搜索的复杂度为  $O(m)$ , 且沿宽度优先搜索寻找包含剩余容量最大的弧的最短增广链  $P$  时,步数不超过  $O(n)$ 。在执行 Step2 时,每次至少删除一条饱和弧,故最多执行  $m$  次寻找包含剩余容量最大的弧的最短增广链,所以新算法的时间复杂度为:  $O(mn^2) + O(m^2) = O(mn^2)$ 。

新算法在实际的操作过程中,降低了时间复杂度。由于寻找包含剩余容量最大的弧的最短增广链、删除饱和弧及修复增广链,都会减少下一步寻找包含剩余容量最大的弧的最短增广链的搜索次数。

3 数值实例与分析

例:如图 2 所示,求从始点  $v_s$  到终点  $v_t$  的网络最大流。其中每条弧旁的数字表示容量。

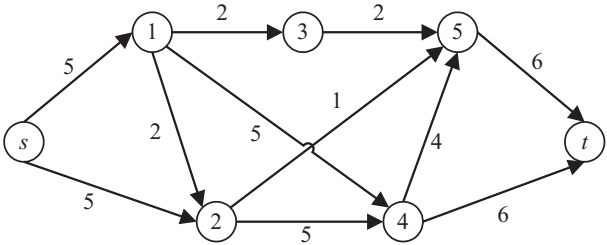


图 2 容量网络  $G$

解:(1)在容量网络  $G = (V,A,c)$  中,将初始可行流取为零流  $f$ ;

(2)从始点  $v_s$  出发,找一条包含剩余容量最大的弧的最短增广链。选取最短增广链  $P = v_s v_2 v_4 v_t$ ,  $\delta =$

$\min\{5,6,6\} = 5$ ,将  $\delta$  的值加到流值  $f$  上,得到新的可行流,仍记为  $f$ 。删除饱和弧  $(v_s, v_2)$  及  $(v_2, v_4)$ ,并在最短增广链  $P$  的各弧上减去  $\delta$ ,如图 3 所示。

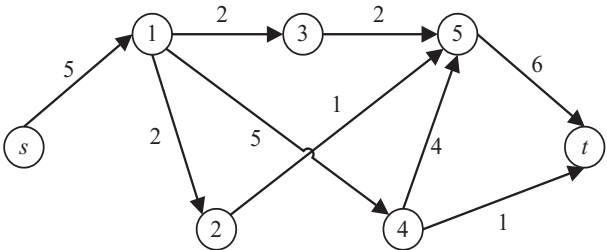


图 3 删除饱和弧后的网络图

(3)修复最短增广链  $P$ ,其中  $v_s, v_4$  为断点,修复可得  $P = v_s v_1 v_4 v_t$  为最短增广链,  $\delta = \min\{5,5,1\} = 1$ ,将  $\delta$  的值加到流值  $f$  上,得到新的可行流,仍记为  $f$ 。

删除饱和弧  $(v_4, v_t)$ ,并在最短增广链  $P$  的各弧上减去  $\delta$ ,如图 4 所示。

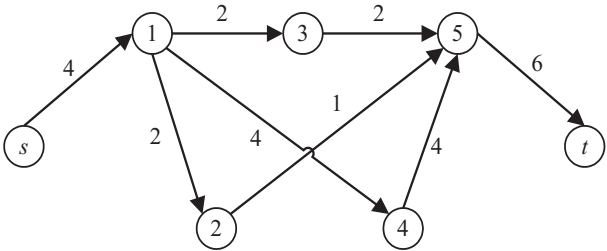


图 4 修复后删除饱和弧的网络图

(4)修复最短增广链  $P$ ,其中  $v_4, v_t$  为断点,修复可得  $P = v_s v_1 v_3 v_5 v_t$  为最短增广链,  $\delta = \min\{4,4,4,6\} = 4$ ,将  $\delta$  的值加到流值  $f$  上,得到新的可行流,仍记为  $f$ 。删除饱和弧  $(v_s, v_1)$ ,  $(v_1, v_4)$  及  $(v_4, v_5)$ ,并在最短增

广链  $P$  的各弧上减去  $\delta$ 。

(5)此时在容量网络  $G = (V,A,c)$  中,不存在从始点  $v_s$  到终点  $v_t$  的可增广链。于是可得图 5 所示的最大流  $f$ ,流值为 10。

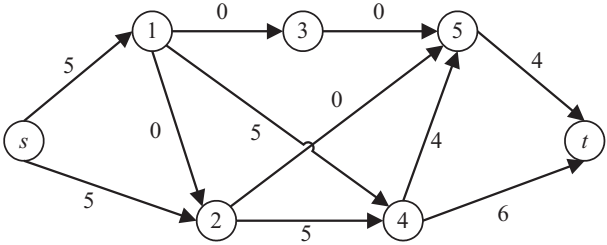


图 5 最大流  $f$

注:用该算法只需一次寻找增广链,两次修复增广链。若用 Ford-Fulkerson 算法,则需要三次寻找增广链,两次修复增广链的过程才能得到最大流值 10,过程复杂,空间占用大,运行时间长,这里就不给出详细的求解过程。

## 4 算法的仿真与比较

将该算法在 MATLAB2012b 软件中进行仿真实验。仿真实验所采用的随机网络是由 BA 无标度网络的方法随机生成的,仿真实验的网络规模设为 500, 1 000, 1 500, 2 000, 2 500, 3 000, 3 500 个节点,且针对给出的网络规模均进行五次仿真实验求平均值。在这样的仿真条件下,利用宽度优先原则对 Ford-Fulkerson 算法与文中算法进行平均运行时间的比较,如图 6 所示。

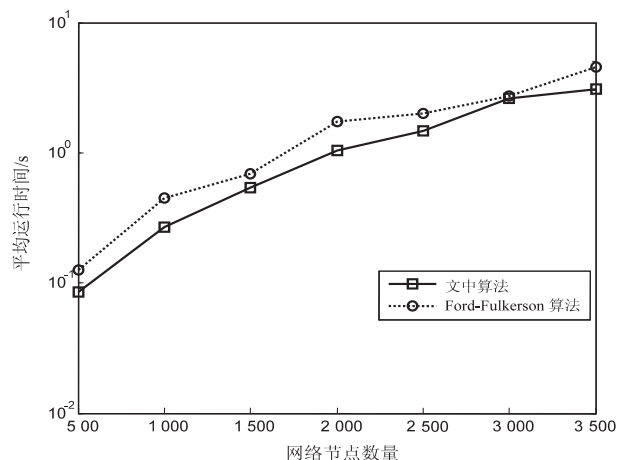


图6 平均运行时间的比较曲线

由图6可以得到,在对 BA 无标度网络进行最大流求解过程中,在相同的网络规模下,文中算法要比 Ford-Fulkerson 算法所用的运行时间短,这与之前的分析相吻合。

## 5 结束语

最大流问题是具有广泛应用的经典组合优化问题,文中是在 Ford-Fulkerson 算法的基础上,提出了一种基于宽度优先的网络最大流求解算法。该算法避免了标号的过程,并且借助宽度优先搜索原理、寻找包含剩余容量最大的弧的最短增广链、删除饱和弧及增广链修复的方法,把复杂的网络简单化,计算量也得到了降低,使整个算法的执行效率得以提高。通过数值实例验证了该算法的简便性、实用性,且仿真实验说明了该算法的运行速率要比 Ford-Fulkerson 算法快。

## 参考文献:

- [1] ARBELAEZ P, MAIRE M, FOWLKES C, et al. Contour detection and hierarchical image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(5): 898-916.
- [2] LI C, HUANG R, DING Z, et al. A level set method for image segmentation in the presence of intensity in homogeneities with application to MRI[J]. IEEE Transactions on Image Processing, 2011, 20(7): 2007-2016.
- [3] 钱颂迪. 运筹学[M]. 北京: 清华大学出版社, 2012: 312-320.
- [4] FULKERSON D R, DANTZIG G B. Computation of maximal flows in network[J]. Naval Research Logistics Quarterly, 1995, 2(4): 277-283.
- [5] 谢政. 网络算法与复杂性理论[M]. 长沙: 国防科技大学出版社, 2003.
- [6] EDMONDS J, KARP R M. Theoretical improvements in algorithmic efficiency for networks flow problems[J]. Journal of ACM, 1972, 19: 248-264.
- [7] 赵礼峰, 纪亚劲. 基于最短增广链的最大流改进算法[J]. 计算机技术与发展, 2017, 27(8): 88-91.
- [8] GOLDBERG A V, RAO S. Beyond the flow decomposition barrier[J]. Journal of ACM, 1998, 45: 783-797.
- [9] 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.
- [10] 赵礼峰, 纪亚宝. 最大流最小截问题的遗传算法研究[J]. 计算机技术与发展, 2017, 27(4): 69-72.
- [11] 张宪超, 江贺. 一个新的最大流问题增载轨算法[J]. 小型微型计算机系统, 2006, 27(9): 1726-1730.
- [12] GOLDBERG A V, HED S, KAPLAN H, et al. Faster and more dynamic maximum flow by incremental breadth-first search[C]//ESA 2015. Berlin: Springer, 2015: 619-630.
- [13] 赵礼峰, 刘艳清. 定流值比例的最小双费用流算法研究[J]. 计算机技术与发展, 2017, 27(4): 94-97.
- [14] 谢凡荣. 求解网络最大流问题的一个算法[J]. 运筹与管理, 2004, 13(4): 37-40.
- [15] 纪伟, 戴理显, 王永红. 网络最大流的“冲塞式”求法[J]. 运筹与管理, 2003, 12(3): 38-42.
- [16] 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.
- [17] 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.
- [18] 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.