文章编号:1000 - 582X(2006)05 - 0132 - 03

网络最大流模型算法及其实现*

张 静[1],邱学绍[2]
(1. 同济大学 软件学院,上海 201800,2. 郑州轻工业学院信息与计算科学系,郑州 450002)

摘 要:针对网络最大流的计算问题,提出了一种网络最大流计算模型的实现方法. 具体作法是灵活运用栈和结构数组以实现算法功能. 首先创建邻接表,其结构包含边的方向、容量、流量等信息. 然后根据邻接表采用标号法寻找增广链,在寻找过程中采用深度优先遍历和广度优先遍历的方法把点存入栈中,并用一数组保存所经过的路径. 直至找出最大流及各边的流量.

关键词:最大流;增广链;标号法;邻接表;结构数组
中图分类号:TP319,O22 **文献标识码**:A

网络最大流理论与方法在运输与网络管理中有着重要应用,Ford,Fulkererson 首先给出了网络最大流计算的有效算法[1]. 党耀国等[2]给出了网络最大流的割集矩阵算法,使得网络最大流的计算更简单,纪伟星等[3]提出了网络最大流的冲塞式算法,徐周波等[4-5]给出了代数决策图(ADD)技术,这使得处理 0 - 1 网络最大流更加有效,谢凡荣[6]给出了最大利润流算法,以便用于决策支持系统. 为了更好地用网络最大流算法来解决实际问题,作者给出了该模型算法的计算机(C++)实现. 实现该算法的关键是数据结构,可灵活运用栈和结构数组以实现算法功能. 首先创建邻接表,采用标号法寻找增广链,并用数组保存所经过的路径,直至找出网络的最大流及各边的流量.

1 算法的设计思想

1.1 标号法

网络最大流算法的基本思想,是先找出一个初始可行流,再逐步改进使可行流值不断增加,直至不能改进时为止.

为了实现这个算法可采用标号法来找增广链,对于一个点 v_i,若能找到从 v_1 到 v_i 的增广链,就给 v_i 一个标号. 首先,给发点 v_1 以标号,然后逐步扩大已标号点的范围,一旦收点 v_n 获得标号,则就找到了从 v_1 到 v_n 的增广链.

每个点的标号记为(α,β),其中的(从 1,2,3,…,n 中取值,而(只取“ + ”,“ - ”. 如设 v_5 的标号是(3, +),它说明 v_5 是从 v_3 得到的标号,或说是将要找到的增广链上 v_5 前面的一点是 v_3,边是(v_3,v_5). 若 v_5 标号为(3, -),它说明 v_5 也是从 v_3 得到标号的,但边是(v_5,v_3).

具体做法是:首先给 v_1 以标号(0, + ∞),于是 v_1 成为已标号而未检查的点,其余各点都是未标号点. 一般地,取一个已标号而未检查点 v_i,对它进行如下检查:

1)考察所有以 v_i 为起点的各边(v_i,v_j). 若(v_i,v_j)满足 $f_{ij}<w_{ij}$,且 v_j 未标号,则给 v_j 以标号$(i,+)$,将 v_j 并入已标号未检查的点的集合.

2)考察所有以 v_i 为终点的边(v_k,v_i),若在(v_k,v_i)上,$f_{ki}>0$,且 v_k 未标号,则给 v_k 以标号$(i,-)$,将 v_k 并入已标号未检查点的集合.

3)重复上述过程,直到 v_n 获得标号;或者所有已标号点都已检查过,而标号过程无法进行,并且 v_n 又没有得到标号. 在前一种情况,一旦 v_n 得到标号,则立即用“反向追踪”法寻找增广链. 出现后一情况,则算法结束,当前可行流就是最大流.

1.2 反向追踪法

设 v_n 的标号是$(i,+)$,则在增广链上 v_n 前面的点是 v_i,边是(v_i,v_n),并将边(v_i,v_n)上的流量调整为 $f_{in}+\theta$(其中 θ 是增广链的增量). 一般地,考察增广链上的点 v_i,若 v_i 的标号为$(j,+)$,则 v_i 前面的点是 v_j,边是

* 收稿日期:2005 - 12 - 17
基金项目:河南省自然基金项目(0511010100)郑州轻工业学院科研基金项目(2004012)
作者简介:张静(1980 -),男,河南平顶山人,同济大学软件学院硕士研究生,主要从事应用软件的研究.

(v_j, v_i)，并将边(v_j, v_i)上的流量调整为$f_{ji}+\theta$；若v_i的标号为$(j, -)$，则v_i前面的点还是v_j，但边为(v_i, v_j)，并将边(v_i, v_j)上的流量调整为$f_{ij}-\theta$. 这样一直追下去，直到v_1为止，便找到了从v_1到v_n的增广链μ，并且也调整了增广链上每条边的流量. 这样便得到了一个新的可行流. 然后抹掉所有标号，再从头开始进行新的标号.

2　算法的具体实现

2.1　数据结构

要实现这个算法就要记录下检查过的边，这样就需要一个结构数组 label[ ]，包含如下内容，

Label[ ] = (v, parent, m, slack).

其示意图如图 1 所示.

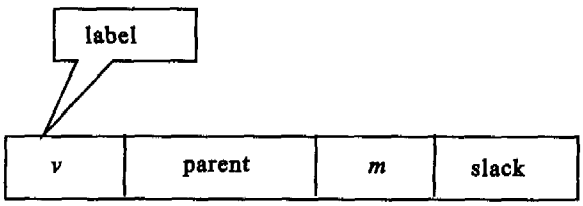

图 1　结构数组 label 示意

其中 v 表示当前点，parent 表示从哪个结点到达 v 的，m 表示此边是前向边还是后向边，slack(v) 表示能够从源点传送到 v 的流量的上界. 前向边和后向边被不同对待，如果结点 v 是从 u 经过一条前向边到达的，则 Label[ ] = $(v, u, 1, \text{slack}(v))$，其中，$\text{slack}(v) = \min(\text{slack}(u), w(u,v)-f(u,v))$，$w(u,v)-f(u,v)$表示当前边的容量和该边负载的流量的差；如果 v 是从 u 经过一条后向边到达的，则 Label[ ] = $(v, u, -1, \text{slack}(v))$，其中，$\text{slack} = \min(\text{slack}(u), f(v,u))$.

要实现该算法需要建立网络的存储结构，采用图的邻接表结构，如图 2 所示. 在每个表结点中含有如下信息：Adj 是表结点在 AdjList 数组中的位置，w 表示容量，f 表示流量，nextarc 指向下一条边，同时增加一个方向域 t 拟记录边的方向. 头结点 data 存放结点信息，firstarc 指向第一个邻接点，同时增加一标志域 b 记录是否被检查过. 为了尽快找到增广链，同时需要记录标过号的值点，需要创建一个栈，采用图的深度优先遍历和广度优先遍历结合对邻接表进行扫描，每扫描完一个点，如不是汇点就压栈保存，同时将扫描过的边存放到 label 数组中，当扫描完与一个结点邻接的所有结点时，再从栈中弹出一个结点继续搜索. 当到达汇点时便通过 label 数组反向追踪改变流量值. 然后清空 label 数组，抹掉标号，重新进行标号，直到栈中为空时算法结束. 此时从源点发出的流量值的和就是所要找的最大流，最后将各边的流值输出.

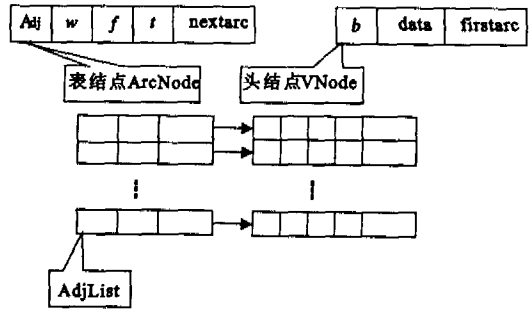

图 2　网络的存储结构 1 示意图

2.2　网络最大流数值算法

1）根据网络构造邻接表，并输入初始可行流 f 的值.

2）将源点压入栈中.

3）判断栈是否为空，如不空从栈中取出一点，并扫描以此点为边尾的邻接表. 否则算法结束，找到最大流转到 8).

4）判断扫描到的边是否满足条件，若是则将此边存入 label 数组中. 否则转 3).

5）如扫描的点是汇点转 6)，否则将此点存入栈中转 3).

6）通过 label 数组采用反向追踪法找到增广链，并通过邻接表改变流量值 f.

7）清空 label 数组和栈，转到 2).

8）输出最大流和各边的流量值.

3　伪代码和流程图

3.1　伪代码

Augmentpath(源点为 v_1，汇点为 v_n的网络)

While(从 v_1到 v_n的增广链经过的每条边 e)

If(是前向边 e)

$f(e) += \text{slaxc}(v_n)$;

else $f(e) -= \text{slack}(v_n)$;

Ford Fulkerson Algorithm(源点为 v_1，汇点为 v_n的网络)

将所有边和结点的流量值设置为 0;

Label[s] = (v_1, *NULL*, 0, ∞);

Stack = { v_1 };//已标号点的集合

While(stack 不为空)

将一个结点 v 从 *stack* 中弹出;

For(所有同 v 邻接的且没有标号的点)

If(vu 是前向边，且 slack(v) >0)

Label[u] = (u, v, 1, min(slack(v), $w(vu)$ − $f(vu)$));//将 u 标号

Else if(vu 是后向边且$f(uv)>0$)

Label[u] = (u,v, -1,min(slack(v),$f(uv)$));

//将 u 标号

If(u 被标记)

If($u==v_n$)

{Augmentpath(network);

Stack = { v_1 }; }//清空栈,并寻找另外的一个增广链

Else 将 u 压入 stack 中.

3.2 流程图

流程图如图 3.

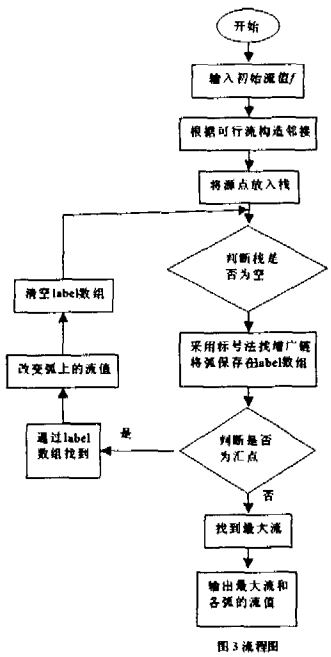

图3 流程图

4 结 语

借助邻接表和结构数组,以实现网络最大流算法中的标号过程、增广链的寻找以及增广链上各边流量的调整,并给出了算法的伪代码,进而使算法的功能得以实现.

参考文献

[1] 胡云权. 运筹学教程[M]. 北京:清华大学出版社,2003. 268 - 275.

[2] 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(9): 125 - 128.

[3] 纪伟,戴理昱,王永红. 网络最大流的"冲赛式"求法. 运筹与管理[J],2004,6(3): 39 - 42.

[4] 徐周波,古天龙. 网络最大流问题的代数决策图(ADD)技术[J]. 桂林电子工业学院学报,2004, 6(3): 54 - 57.

[5] 徐周波,古天龙,赵念忠. 网络最大流问题的一种新的ADD 的求解方法[J]. 通信学报,2005, 2(2): 1 - 8.

[6] 谢凡荣. 求解网络最大流问题的一个算法[J]. 运筹与管理,2004,10(4): 37 - 42.

Algorithm and Its Implementation for the Network Maximal Flow

ZHANG Jing[1], *QIU Xue-shao*[2]

(1. Software Institute, Tongji University, Shanghai 201800, China;

2. Department of Information and Computation Science., Zhengzhou Institute of Light Industry, Zhengzhou 450002, China)

Abstract: On the algorithm of the network maximal flow, the paper provides a method of achieving it. The concrete procedure is to achieve the algorithm by using stack and structural array. First of all, an adjacency list should be established and its composition chiefly includes orientation, capacity, flux and so on. Afterwards the labeling method is adopted to find the augmenting chain according to the adjacency list. In the process, some spots are stored in the stack by means of the depth superior traverse and range superior traverse and the course way is also conserved in the array. Keep on doing this till the maximum flow and the flow of each arc are all found.

Key words: maximal flow; augmenting chain; label method; adjacency list; structural array

(编辑 张小强)