第10卷 第8期 2010年3月
1671-1815(2010)08-1898-04

科 学 技 术 与 工 程
Science Technology and Engineering

Vol. 10 No. 8 Mar. 2010
© 2010 Sci. Tech. Engng.

一类度约束最小生成树问题的 Dijkstra 算法

袁卫东
(宝鸡职业技术学院基础部,宝鸡 721013)

摘 要 度约束最小生成树问题是网络设计和优化中的一个 NP 难题。结合该问题的特征,基于 Dijkstra 算法的基本思想,提出了一种求解网络 G 关于指定节点的最大度最小生成树的新算法。该算法在保证指定节点最大度的前提下,每次通过选取剩余边中权最小的边加入当前网络,最终得到网络 G 关于指定节点的最大度最小生成树。同时对算法的复杂度进行了分析。最后通过与其他算法的仿真比较和算例,表明了新算法的有效性。

关键词 最大度 度约束 Dijkstra 算法 最小生成树

中图法分类号 TP393.02; **文献标志码** A

最小生成树(minimum spanning tree, MST)问题是运筹学、组合优化中的一个常见的基本问题,近年来在计算机网络、通信网络、运输网络设计等方面得到了广泛的应用,并陆续提出了若干有效地求解方法[1—3]。该问题最早是 Boruvka 于 1926 年提出的,旨在寻找电力线网络最优的布局[4]。一般地,生成树中每个节点的度数可以不受限制,但在实际应用中,某些节点的度数往往会受到各种因素的影响,使这些节点的度数不能随意取值。比如在军事网络中,为增强网络的可靠性,防止节点故障引起的网络瘫痪而对网络节点的度数进行限制;或在通信网络中,为了维持网络的传播速度及负载平衡而对节点的度数加以限制等。像这种节点带有度约束的最小生成树就是一类重要的度约束最小生成树[5,6](degree-constrained minimum spanning tree, DCMST)问题。

本文基于求最小生成树 Dijkstra 算法的基本思想,在网络 G 关于指定节点 v_0 的最大度支撑树的最大度等于 v_0 在 G 中度的结论的基础上,提出了一种求解网络 G 关于指定节点 v_0 的最大度最小生成树的新算法。该算法能在有效时间里产生可接受的解,且与文献[7]中的算法相比,是一种效率更高且易于计算机实现的算法。

1 度约束最小生成树的数学模型[8]

在连通无向网络 $G=G(V,E,W)$ 中,$V=\{v_1,v_2,\cdots,v_n\}$ 是节点集,E 是边集,$W=(w_{ij})_{n\times n}$ 为网络的权矩阵,其中 $w_{ij}=w_{ji}, w_{ii}=+\infty\ (i,j=1,2,\cdots,n)$;若 $v_iv_j\in E$,则 $w_{ij}=W(v_iv_j)$,否则 $w_{ij}=+\infty$。并设各节点的度约束为 $b_i(i=1,2,\cdots,n)$,则 DCMST 问题的数学模型为:

$$\min Z = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}$$

$$\begin{cases} \sum\limits_{i=1}^{n} x_{ij} \leqslant b_i & (1) \\ \sum\limits_{i,j\in S} x_{ij} \leqslant |S|-1, \forall S\subset V, S\neq\varnothing & (2) \\ \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} x_{ij} = n-1 & (3) \\ x_{ij}\in\{0,1\}, i,j\in V \end{cases}$$

这里,变量 $x_{ij}=\begin{cases}1 & 边(i,j)在最优树中\\ 0 & 其它\end{cases}$,其中约束式(1)保证了所求得的树满足度约束条件,约束式(2)、式(3)保证了所得的是一棵生成树。

2009 年 12 月 17 日收到

第一作者简介:袁卫东(1962—),男,硕士,宝鸡职业技术学院基础部讲师,研究方向:应用数学。

上述模型中,当 $b_i \geqslant n-1$ 时,则 DCMST 问题转化为无约束情况下一般的最小生成树;而当某个 $b_i < n-1$,其余 $b_i \geqslant n-1$ 时,即为本文要讨论的单点度约束最小生成树问题。

2 相关知识[9—11]

定义 1 设 $G(V,E,W)$ 是一个简单的连通无向网络,$v_0 \in V$ 是特别指定的一个节点,k 为给定的一个正整数。如果 T 是 G 的一个支撑树且 $d_T(v_0)=k$,则称 T 为 G 的 k 度限制支撑树。

定义 2 设 $G(V,E,W)$ 是一个简单的连通无向网络,$v_0 \in V$ 是网络中指定的一个节点,在这个网络的所有关于节点 v_0 的最大度限制支撑树中权最小的那些支撑树称为网络 $G(V,E,W)$ 关于节点 v_0 的最大度最小生成树。

定理 1[7] 设 $G(V,E,W)$ 是一个简单的连通无向网络,$v_0 \in V$ 是 G 中指定的一个节点,则网络 G 关于节点 v_0 的最大度支撑树 T' 一定存在,且$(i=1,\cdots,m)$(其中 $d_{T'}(v_0)$ 表示节点 v_0 在 T' 中的度,$d_G(v_0)$ 表示节点 v_0 在 G 中的度)。

定理 2 设 T 是连通无向网络 $G(V,E,W)$ 的 k 度限制支撑树,则 T 是 G 的最小 k 度限制树当且仅当下面三个条件同时成立:

(1) 对于 G 中任何两条与 v_0 关联的边所产生的 T 的可行交换都是不可改进的,即 T 通过这类可行变换得到的新的支撑树 T' 满足 $w(T') \geqslant w(T)$。

(2) 对于 G 中任何两条不与 v_0 关联的边所产生的 T 的可行交换都是不可改进的。

(3) 对于 T 的任何两个可行交换$(+e_1,-f_1)$和$(+e_2,-f_2)$,若 e_1,f_2 与 v_0 关联,e_2,f_1 不与 v_0 关联,则有 $w(f_1)+w(f_2) \leqslant w(e_1)+w(e_2)$。

3 新算法

3.1 算法思想

新算法是在保证指定节点 v_0 最大度的前提下,采用求最小生成树的 Dijkstra 算法的基本思想,每次通过选取剩余边中权最小的边及该边的相应节点,加入当前网络,直至把与 v_0 不相邻的节点全部加入当前网络,则得到原网络关于指定节点 v_0 的最大度最小生成树。

3.2 算法步骤

Step0:求 $E_G(v_0)$,$V_G(v_0)$. 记 $E_G(v_0)$ 表示 G 中与节点 v_0 关联的边集,$V_G(v_0)$ 表示 G 中与节点 v_0 相邻的点集,J 表示 $V_G(v_0)$ 中节点的下标集。

令 $E=E_G(v_0)$,$S=\{v_0\}\cup V_G(v_0)$,$R=V\backslash\{v_0\}\cup V_G(v_0)$。

$\forall v_j \in R$,令 $u_{p_j j}=\min\limits_{k\in J}\{w_{kj}\}$,$p_j \in J$;

Step1:选取 $u_{p_i i}=\min\limits_{v_j\in R} u_{p_j j}$. 若 $u_{p_i i}=\infty$,停止,G 中不存在关于 v_0 的最大度支撑树;否则,$S:=S\cup\{v_i\}$,$R:=R\backslash\{v_i\}$,$E:=E\cup\{v_{p_i}v_i\}$,转 step2;

Step2:若 $R=\varnothing$,结束,图(S,E)是 G 的关于 v_0 的最大度最小生成树;否则,

对 $\forall v_j \in R$,若 $u_{p_j j} \leqslant w_{ij}$,则 $u_{p_j j}$ 保持不变;否则,置 $p_j:=i$,且 $u_{p_j j}:=w_{ij}$,转 step1;

3.3 复杂度分析

定理 3 新算法的最坏运算复杂度为 $O(n^2)$。

证:设简单无向网络有 n 个节点,m 条边。由算法步骤可知:执行 Step0 需判断 m 次,且对 R 中的 $n-1-d_G(v_0)$ 个节点,每个要做 $d_G(v_0)-1$ 次比较;第一次执行 step1 要做 $n-d_G(v_0)-2$ 次比较,第二次执行 step1 则做 $n-d_G(v_0)-3$ 次比较,依此类推,执行 step1 共需 $\frac{(n-d_G(v_0)-2)(n-d_G(v_0)-1)}{2}$ 次比较;而第一次执行 step2 先做 1 次判断,然后做 $n-d_G(v_0)-2$ 次比较,第二次执行 step2 也要先做 1 次判断,再做 $n-d_G(v_0)-3$ 次比较,类推可得,执行 step2 共需做 $n-d_G(v_0)$ 次判断和 $\frac{(n-d_G(v_0)-2)(n-d_G(v_0)-1)}{2}$ 次比较;整个算法共需 $m+(n-1-d_G(v_0))(d_G(v_0)-1)+\frac{(n-d_G(v_0)-2)(n-1-d_G(v_0))}{2}+n-d_G(v_0)+\frac{(n-d_G(v_0)-2)(n-d_G(v_0)-1)}{2}=(n-d_G(v_0)-1)\times(n-3)+m+n-d_G(v_0)$ 次运算。又由于 $d_G(v_0)\geqslant 1$,

且边数而弧数 $m \leqslant \frac{n(n-1)}{2}$,因此 $(n-d_G(v_0)-1) \times (n-3)+m+n-d_G(v_0) \leqslant (n-2)(n-3)+\frac{n(n-1)}{2}+n-1=\frac{3n^2}{2}-\frac{9n}{2}+5$。由此可见,新算法的最坏运算复杂度为 $O(n^2)$。

3.4 仿真比较

为检验算法的效果,我们采用 Waxman 提出的随机网络模型进行了测试。该模型产生的随机网络与真实网络比较接近。我们在 CPU 为 pentium M processor 2.00 GHz、内存 1 G 的计算机上,用新算法和文献[7]的算法计算了由该模型生成的不同规模的连通网络关于指定节点的最大度最小生成树,运行时间如表 1 所示。

表 1 两种算法的运行时间

网络规模		5	7	9	12	14	16	18	20
运行时间/s	新算法	1.713 6	3.084 2	4.712 9	6.831 8	9.582 6	13.486 5	16.145 3	20.667 8
	文献算法	3.245 1	5.022 5	9.374 7	16.974 4	27.379 1	46.042 7	59.003 3	72.359 2

从表 1 数据可以看到:随着网络规模的增大,新算法的运算速度比文献[7]的算法相比明显要高。另外,由于新算法的思想基于 Dijkstra 算法,因此它是一种更易于计算机实现的算法。

3.5 算例

A 公司在某地区设置了两级销售网络,当产品完成生产后需要通过铁路运送到该地区的某铁路货运站,为了节约成本需要先将货物分别发运到一级销售点,一级销售点再向各二级销售点发货。同时为了根据实际的市场供求,进行实时动态调节,要求各销售点之间能互相供货。现假设有一批货物运到了货运站,要求在保证货运站的货物必须配送到各一级销售点的前提下,怎样安排发货才能使总的费用最少。该市的销售网络如图 1 所示。

解 把图 1 看成一个简单连通网络 G,火车站记为 v_0,一级销售点分别记为 v_1、v_2,二级销售点分别记为 v_3、v_4、v_5,边上的权值表示两个销售点之间的运输费用,则该问题就是求网络 G 关于 v_0 的最大度约束下的最小生成树。

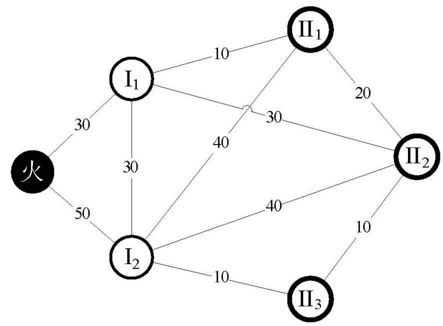

图 1 某市的销售网络

Step0: $E_G(v_0)=\{v_0v_1, v_0v_2\}$, $V_G(v_0)=\{v_1, v_2\}$, $J=\{1,2\}$,

$E=\{v_0v_1, v_0v_2\}$, $S=\{v_0, v_1, v_2\}$, $R=\{v_3, v_4, v_5\}$,

对 v_3, $u_{13}=\min\{w_{13}, w_{23}\}=\min\{10, 40\}=10$。

对 v_4, $u_{14}=\min\{w_{14}, w_{24}\}=\min\{30, 40\}=30$。

对 v_5, $u_{25}=\min\{w_{15}, w_{25}\}=\min\{\infty, 10\}=10$。

Step1: 取 $u_{13}=10$, 则 $S=\{v_0, v_1, v_2, v_3\}$, $R=\{v_4, v_5\}$, $E=\{v_0v_1, v_0v_2, v_1v_3\}$。

Step2: $R \neq \phi$, 对 v_4, $u_{14}=30 \geqslant w_{34}=20$, 置 $p_j := 3$, 则 $u_{34}=20$。

对 v_5, $u_{25}=10 \leqslant w_{35}=\infty$, 则 $u_{25}=10$ 不变。

经过三次迭代后,得到网络 G 关于 v_0 的最大度最小生成树 T_3,如图 2 所示,且 $w(T_3)=110$。

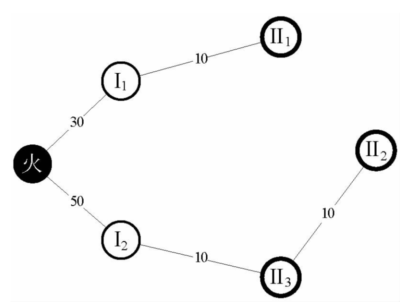

图 2 网络 G 关于 v_0 的最大度最小生成树 T_3

4 结语

度约束最小生成树问题作为网络优化领域的一个常见问题,对其求解算法的设计具有着重要的

理论意义和应用价值。本文提出了一种解决网络 G 关于指定节点的最大度最小生成树问题的有效算法,并对算法进行了分析,但它只适用于一类度约束最小生成树问题的求解。

而随着当前社会网络化进程的加快,越来越多的带复杂约束的网络优化问题呈现在我们面前,为这些问题设计好的求解算法是推进这一进程的关键方面。因此对更一般地度约束最小生成树问题,设计精确且有效的算法仍是一个值得研究的问题。

参 考 文 献

1 Wang Guangrong, Gu Naijie. An efficient parallel minimum spanning tree algorithm on message passing parallel machine. Journal of Software, 2000;11(7):889—898

2 Julstrom B A. Greedy heuristics for the bounded diameter minimum spanning tree problem. ACM Journal of Experimental Algorithmics, 2009; 14(1):1—14

3 Singh A, Gupta A K. Improved heuristics for the bounded-diameter minimum spanning tree problem. Soft Computing, 2007;11(10):911—921

4 玄光男, 程润伟. 遗传算法与工程优化. 北京: 清华大学出版社, 2004; 67:231—233

5 张春丽, 何 镕. 基于免疫-蚁群算法的度约束最小生成树算法. 计算机工程与设计,2008;29(3): 694—696,699

6 Liao Feixiong, Ma liang. Heuristic algorithm for degree-constrained minimum spanning tree. J University of Shanghai for Science and Technology, 2007;29(2): 142—144

7 马来焕. 求解最大度约束下最小生成树的新算法. 江南大学学报, 2009;8(5):1—4

8 宋海洲. 求解度约束最小生成树的单亲遗传算法. 系统工程理论与实践, 2005;4(4):61—66

9 谢 政. 网络算法与复杂性理论. 长沙:国防科技大学出版社, 2003: 33—37

10 张先迪,李正良. 图论及其应用. 北京:高等教育出版社,2005: 31—43

11 West D B. 图论导引. 李建中,骆吉洲,译. 北京:机械工业出版社,2006:63—67

Dijkstra Algorithm for a Kind of the Degree-constrained Minimum Spanning Tree Problem

YUAN Wei-dong

(Faculty of Science, Baoji Vocational Technology College, Baoji 721013, P. R. China)

[**Abstract**] As it knows to all, the degree-constrained minimum spanning tree problem is a NP difficulty in the network design and optimization. Therefore, concerning about the characteristics of this problem, a new algorithm is presented, basing on the fundamental idea of the Dijkstra algorithm. With the given node being maximum degree assured by this new algorithm, and selecting the edge with the minimum weight in remaining edges every single time, finally it comes out the minimum spanning tree of the given node under the maximum degree constraint in network G. At the same time, the complexity of the new algorithm has been analyzed. The effectiveness is proved though an simulation comparement with the other algorithm.

[**Key words**] maximum degree degree constraint Dijkstra algorithm minimum spanning tree