第 46 卷　第 1 期　　武汉大学学报(自然科学版)　　Vol. 46　No. 1
2000 年 2 月　　J. Wuhan Univ. (Nat. Sci. Ed.)　　Feb. 2000, 037～040

文章编号：0253-9888(2000)01-0037-04

用节点子树剪枝构造网络最短路径

郭成城，晏蒲柳，夏德麟
(武汉大学 电子信息学学院，武汉 430072)

摘　要：从网络联结图的邻接矩阵出发，提出在 Internet 网络环境下直接构造网络最短主树的一种方法——节点子树剪枝法. 在无约束条件和有约束条件(如转接数、传输链路带宽限制情况下)，给出 Internet 最短主树算法. 该算法用于计算 Internet 环境下可扩展的 IP 路由表具有较高效率.

关　键　词：节点子树剪枝法；路由选择；互联网
中图分类号：TP 393.1　　**文献标识码**：A

Internet 业务每 9 个月增加一倍，这一状态或许还要持续很长时间. 为适应这种需要 IP 路由器必须快速、有效、有较强的扩展能力，应能根据网络当前拓扑，快速最优地制定当前 IP 路由表，即快速确定节点之间的最佳路由与备用路由，特别是在转接次数和传输链路带宽等限制条件下完成这一工作. 而网络最短路径问题正是其中的基本问题.

最短路径问题是网络优化的一个传统问题，也是当前 Internet 环境下，有扩充能力的 IP 路由表计算中的重要问题. 在某些情况下，为了确定连通无向网络图中一棵满足某些特殊条件的主树，除了把联结图的所有主树都求出来，很难找到更有效的办法. 故对于这一类问题目前多采用穷举搜索联结图主树的方法(或改进的变形方法)，然后再按约束条件筛选，以求得满足某种限制条件的最短径主树[1]. 虽然这种方法从理论上说能够求得问题的最优解，但计算量太大. 对于 n 端的全连通网而论，其主树数为 n^{n-2}，当 n 增大时，这个数值将急剧增加. 故从计算复杂度而言，它已属于 NP 问题.

为了克服上述算法的缺点，本文从网络联结图的邻接矩阵 C 的特性出发，提出一种由 C 矩阵的行(或列)矢量表示子树集(它可能是联结图的主树，视所给定的网络图的结构而定)，直接构造网络图的最短径主树算法，以求得问题的最优解.

1　基本原理与算法

网络图 $G(V,E,D)$ 的最短径问题，实质上是求 G 的最短主树 T，而该主树的边集 E_t 是 G 的边集 E 的一个子集，并且该子集覆盖 G 的所有端，而形成一无环的联结子图，亦即

$$\begin{aligned} &T \subset G, 且\ V_t = V, E_t \subset E, \\ &D_t \subset D\{d_{ij} | i,j = 1,2,\cdots,n\} \end{aligned} \tag{1}$$

故由(1)式可知：在无约束条件情况下，求网络的最短路径问题可视为从网络图的边集 E 中选取一子集 E_t，使其满足最短径(即 $\min D_t$)主树的条件. 显然，该子集的边元素的个数 $b_n = n-1$(n 为端节点数)，并且这些边元素的径长 d_{ij} 是 E 中元素的较小者(即满足连接性条件下的一组边元素). 对于无约束条件的最短径问题，在保证联结图的连接性条件(即对每棵节点树的较长树枝进行剪除)，最后对经剪枝后的节点子树集求并，即可以得到 G 的最短径主树. 下面给出这种算法的具体步骤.

1.1　无约束条件的情况

为了便于说明该算法步骤，现令 t_j($j=1,2,\cdots,n$)表示第 j 棵节点子树，而用 V_{t_j} 表示第 j 棵节点子树端点的集合，m 表示边集 E 的边元素总数，n 为 G 的端点总数，于是该算法可描述为：

收稿日期：1999-05-28
基金项目：国家自然科学基金重大科研项目(69896240)；国家 863 计划资助项目(863-306-ZT05-2)
作者简介：郭成城(1961-)，男，讲师，现从事计算机网络与信息处理.

(i) 根据 G 的邻接矩阵 C 生成 n 个端点子树：$t_1=\{e_{1,k}|k\in V_{t_1}\}$，$t_2=\{e_{2,l}|l\in V_{t_2}\}$，…，$t_n=\{e_{n,p}|p\in V_{t_n}\}$，然后分别求子树 $t_j(j=1,2,\cdots,n)$ 的径长和 d^{t_j}，并按其大小排序生成端点子树径长和子集 $D^t=\{d^{t_j}\}$，同时令端点子树的连通度为 d_j；

(ii) 将联结图 G 的边集 E 按其元素大小依次排列，然后将其分解为两个初始子集 $\{E^c\}$(待剪边子集)和 $\{E^m\}$(待保留边子集)，显然它们元素的个数分别为 $(m-n+1)$ 和 $(n-1)$，同时设定剪除边计数器 CEN＝0；

(iii) 根据 D^t 的顺序依次提取端点子树 t_j(j 为端点号)：

若 $t_j\cap E^c\equiv t_j$，则 t_j 仅保留其中的最短边(即 $\min\{e_{ji}\}, i\in V_{t_j}$)，而其它边($d_j-1$ 条边)均被剪除(包括其它含该 d_j-1 条边的子树)，同时修改 E^c、E^m 和其对应的 d_j(将被保留的子树边放置 E^m 末尾，将 E^m 最前面的元素放置在 E^c 的尾部)，并修改计数器：

$$\mathrm{CEN}+d_j-1\Rightarrow\mathrm{CEN}$$

若 $t_j\cap E^c=\Phi$，则 t_j 的边元素均保留(不对 t_j 进行剪除树枝处理)；

若 $t_j\cap E^c\neq\Phi$，则对子树 $t_j(j=1,2,\cdots,n)$ 剪除 $t_j\cap E^c$ 的所有边元素(假定其边数为 q)，同时修改计数器 $\mathrm{CEN}+q\Rightarrow\mathrm{CEN}$；

(iv) 若 $\mathrm{CEN}\leqslant n-1$，则转(iii)，否则转(v)；

(v) 求和，生成最短主树 T，$t_1\cup t_2\cdots\cup t_n\Rightarrow T$，算法结束.

可以证明，本算法所求得的主树必为最短径主树. 因为该算法在构造主树时，是在满足主树定义的前提下，从边集 E 中选取一组最短边元素，故其径长必为最短.

现在讨论算法的复杂性. P 算法(R. C. Prim 算法)和 D 算法(E. W. Dijstra 算法)[3]从开始到终止共执行 $n-1$ 步，而每步须从 r 个 G_r 中的端与 $n-r$ 个 $G\sim G_r$ 中的端之间进行距离比较，求出较小者. 可是在运算的第 r 步中，要进行 $r(n-r)-1$ 次比较，P 算法的计算量如(2)式，为 $O(n^3)$ 量级.

$$\sum_{r=1}^{n-1}(r(n-r)-1)=\frac{1}{6}(n-1)(n+3) \quad (2)$$

P 算法中的比较是有重复的，而 D 算法对每次比较的结果进行标记，使比较次数降至 $O(n^2)$ 量级，其总计算量如下：

$$\sum_{k=1}^{n-1}3(n-k)=\frac{3}{2}n(n-1) \quad (3)$$

本算法的迭代步数由主树的边数决定. 在端点子树集 $\{t_j\}$ 中，只有 d_j 个边元素. 而在某些端点子树(d^{t_j} 大者)的边元素中，可能有多条边属于 E^C，故迭代的次数远小于 $(n-1)$. 同时，每棵端点子树只需对子集 E^C 最末尾的边元素进行比较，即可确定所判别的边元素是属于 E^C 还是属于 E^m，进而决定是否对它执行剪枝处理. 本算法的最大计算量如下：

$$(n-1)(\max(d_j)-1)=(n-1)(n-2) \quad (4)$$

(4)式是按全联结图来估算的，实际中通信网络一般不会为全连通，故计算量远小于(4)式的计算值. 可以看出本算法比 P 算法和 D 算法的效率要高.

1.2 有约束条件的情况

在许多情况下，通信网内的 n 各节点除了连通性要求外，还会提出其他要求. 例如，节点间通信时转接次数不能过多，线路上传输容量有限等. 可归结为在约束条件下求最短径主树. 现在尚无一般的有效算法. 目前有两种求解这类问题的方法，即穷举法和调整法. 由于主树个数是随端点 n 的增加呈指数增长，故它们只适用于 n(端点)和 m(边数)都不太大的情况.

上述提出的算法稍加修改，亦可用于求解有约束条件的最短径问题. 其方法是：在上述算法的第(iii)步中增加剪除树枝的选取要求，并附上给出的约束条件，其他步骤不变. 在网络中给出的约束条件有：转接数、业务量、延时和流量等，而且这些限制条件均与路径有关. 因此，需要对节点子树 t_j 端节点 j 至主站的可能路径建立一路径集 $p(j,i)$(其中 j,i 分别为子树端节点和主站节点)，然后从集合 $p(j,i)$ 中选取满足条件的边元素(其边长最短)作为保留树枝，而节点子树的其它树枝剪除. 据此，对于有约束条件(只考虑转接数与业务量)的最短径问题的算法步骤可描述如下：

(i)～(ii)步骤与上述无约束条件情况相同.

(iii) 若 $t_j\cap E^C\equiv t_j$，对 t_j 仅保留满足约束条件的最短边，即 $\min\{e_{ji}|r_{p(e_{ji})}\leqslant K, M\ p(e_{ji})\leqslant M, i\in V_{t_j}\}$，其它边 (d_j-1) 条均剪除(包括含该 d_j-1 条边的其他子树)，同时修改 E^C、E^m 和相应的若干个 d_j，并修改计数器 $\mathrm{CEN}+d_j-1\Rightarrow\mathrm{CEN}$；

若 $t_j\cap E^C=\Phi$，该子树不剪枝(至少要有一条路径满足约束条件. 如没有路径满足约束，则该问题无解，即约束条件太苛刻)；

若 $t_j\cap E^C\neq\Phi$，当 $t_j\cap E^C$ 的边元子集和在 E^m 的边元集的路径都满足约束条件或者只有 E^m 的边元满足约束条件时，则保留 E^m 的边元集，剪除 $t_j\cap E^C$

的 q 条边元,并令 CEN$+Q\Rightarrow$CEN;当只有 $t_j\cap E^C$ 的边元子集的路径满足约束条件时,保留其中最短边 $\min\{e_{ji}\mid r_{pcji}\leqslant M, i\in V_{t_j}\}$ 和在 E^m 中的边元,将 $t_j\cap E^C$ 中的其它 q 条剪除,并令 CEN$+q\Rightarrow$CEN.

(iv)~(v)步骤与上述无约束条件情况相同.

可以证明该算法求出的主树必为满足约束的最短径主树. 无约束和有约束的最短径求解问题,通过在算法中增加分支开关可以合二为一.

1.3　数字示例

例 1:给定通讯网络 G(如图 1 所示),已知所有边的权 d_{ij};(a) 求网络图 G 的最短径;(b) 求站点通信转接次数 2 的最短径.

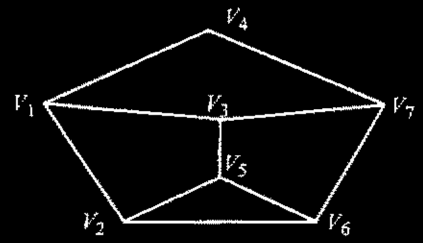


图 1　通讯网络图 G

由给定的 G 可得其权联结矩阵 C、D^t 和 E 为:

$$C=\begin{bmatrix} 0 & 0.5 & 2 & 1.5 & \infty & \infty & \infty \\ 0.5 & 0 & \infty & \infty & 1.2 & 9.2 & \infty \\ 2 & \infty & 0 & \infty & 5 & \infty & 3.1 \\ 1.5 & \infty & \infty & 0 & \infty & \infty & 4 \\ \infty & 1.2 & 5 & \infty & 0 & 6.7 & \infty \\ \infty & 9.2 & \infty & \infty & 6.7 & 0 & 15.6 \\ \infty & \infty & 3.1 & 4 & \infty & 15.6 & 0 \end{bmatrix}$$

$$D^s=\begin{bmatrix} 4 \\ 10.9 \\ 10.1 \\ 5.5 \\ 12.9 \\ 31.5 \\ 22.7 \end{bmatrix}\quad D^t=\begin{bmatrix} 31.5 \\ 22.7 \\ 12.9 \\ 10.9 \\ 10.1 \\ 5.5 \\ 4 \end{bmatrix}\quad E^t=\begin{bmatrix} 15.6 \\ 9.2 \\ 6.7 \\ 5 \\ 4 \\ 3.1 \\ 2 \\ 1.5 \\ 1.2 \\ 0.5 \end{bmatrix}$$

初始设定 $E=\{E^C,E^m\}$,其中 $E^C=\{15.6,9.2,6.7,5\}$,$E^m=\{4,3.1,2,1.5,1.2,0.5\}$,CEN$=0$. 由 C 可得节点子树 t_j 分别为:$t_1=\{0.5,2,1.5\}$,$t_2=\{0.5,1.2,9.2\}$,$t_3=\{2,5,3.1\}$,$t_4=\{1.5,4\}$,$t_5=\{1.2,5,6.7\}$,$t_6=\{9.2,6.7,15.6\}$,$t_7=\{4,3.1,15.6\}$.

2　结果与分析

按上述算法步骤可求得问题的解,其迭代过程如下表 1 所示. 同理,可求得有约束条件(限制转接次数)下的最短径主树,其迭代过程如表 2 所示.

表 1 中 CEN$=4=m-n+1$,迭代结束对 t_j 求得 $T=\{t_1\}\cup\{t_2\}\cup\cdots\cup\{t_7\}$,主树长 $\sum d=15$,其中 T 就是在无约束条件下的最短矩主树,如图 2 所示. 表 2 中 CEN$=4=m-n+1$,迭代结束对 t_j 求得 $T=\{t_1\}\cup\{t_2\}\cup\cdots\cup\{t_7\}$,主树长 $\sum d=17.5$,最大转接数 $r=1$,其中 T 为有约束条件下的最短径主树如图 3 所示,$K_{\max}=2$.

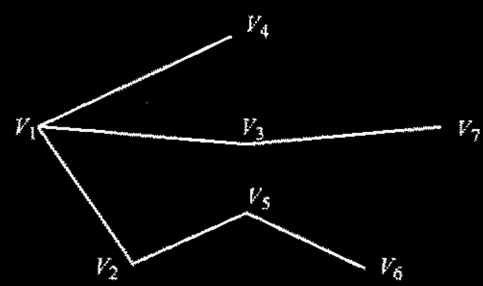


图 2　无约束条件下的最短径主树

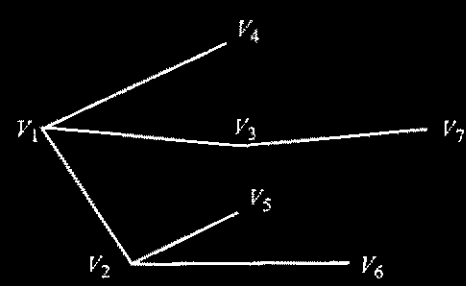


图 3　有约束条件下的最短径主树

本文研究的 Internet IP 路由表计算中网络最短径求解的节点子树剪枝算法,与现有的此类问题的算法比较有如下特点:(i) 原理简单,计算量小,效率高. 能有效地求解相应问题的最优解(在问题最有优解的情况下);(ii) 该方法具有求解的统一性. 能对无约束的最短径问题和有约束的最短径问题用同一种计算流程进行求解;(iii) 算法流程简单,易于编程实现.

通讯网的最短径问题也可用 0/1 规划来描述和求解. 但传统的 0/1 规划算法难于对这类问题求解,而必须采用特殊的搜索方法[2~5]来对问题求解,特别是当问题涉及的维数较高时计算量很大,这也提出节点子树剪枝法的原因.

表1 无约束条件的迭代过程

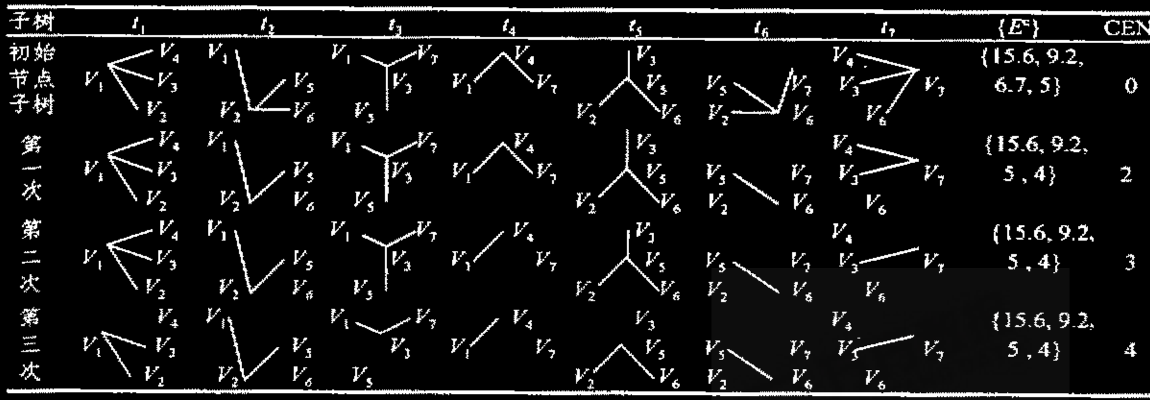
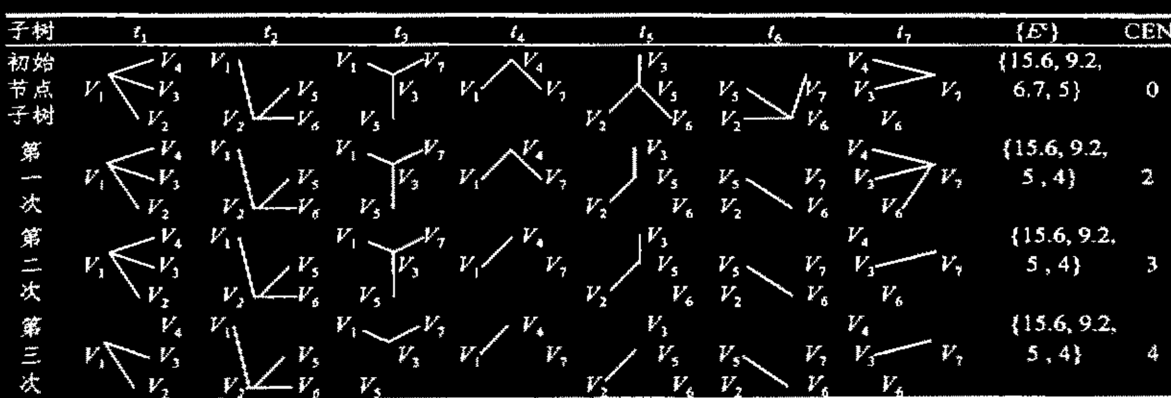


表2 有约束条件的迭代过程



参考文献：

[1] ZHUO Jiong-ban. *The Theorical Basis of Communication Network*. Beijing: The Press of Public Post and Telecom, 1991(Ch).

[2] DAI Zhao-yi. *The Theory on Information Amount in Computer Network*. Beijing: The Press of Electornic Industry, 1997(Ch).

[3] Pluris Inc, Pluris Massively Paraller Routing. http://WWW.pluris.com, 1999-05-06.

[4] XIA De-lin, ZHOU Jing-fang. A Continuous/Discrete Nonlinear Optimization Design Method, Approximation. Optimization and Computing. *Theory and Application IMACS*, 1990: 353-356.

[5] Internet Performance Measurement and Analysis Project. http://WWW.merit.edu/ipma, 1999-05-06.

Research About the Shortest Path in Network Based on the Clipping Branch Method

GUO Cheng-cheng, YAN Pu-liu, XIA De-lin

(College of Electronic Information, Wuhan University, Wuhan 430072, China)

Abstract: Using graph theory the algorithm——clipping Branch in nods Subtree, to construct the shortest maintree in a scalable network environment directly was developed. The basic principle of the algorithm is explained as well as the detailed solution of it within bounds (restrains of jump number or the bandwidth of the transmission link) and not. Compared to the convential approach, the approach is a speedup with low complex of compuation.

Key words: clippling branch; method routing decision; Internet

用节点子树剪枝构造网络最短路径

作者： 郭成城， 晏蒲柳， 夏德麟， GUO Cheng-cheng， YAN Pu-liu， XIA De-lin
作者单位： 武汉大学电子信息学学院,武汉,430072
刊名： 武汉大学学报（自然科学版） ISTIC PKU
英文刊名： JOURNAL OF WUHAN UNIVERSITY(NATURAL SCIENCE EDITION)
年，卷(期)： 2000,46(1)

参考文献(4条)

1. ZHUO Jiong-ban The Theorical Basis of Communication Network 1991

2. DAI Zhao-yi The Theory on information Amount in Computer Network 1997

3. Pluris lnc Pluris Massively Paraller Routing 1999

4. Internet Performance Measurement and Analysis Project 1999

引证文献(1条)