树状网络上带度约束的 k-tree core 问题

杨建芳[1]，刘建贞[1]，黄孙琴[2]
YANG Jian-fang[1]，LIU Jian-zhen[1]，HUANG Sun-qin[2]

1.杭州电子科技大学 理学院 运筹与控制研究所，杭州 310018
2.浙江长征职业技术学院 基础部，杭州 310023
1.Institute of Operational Research & Cybernetics，Hangzhou Dianzi University，Hangzhou 310018，China
2.Zhejiang Changzheng Professional & Technical College，Hangzhou 310023，China
E-mail：jianfangyang@126.com

YANG Jian-fang，LIU Jian-zhen，HUANG Sun-qin.*k*-tree core problem with degree constraint in tree network.Computer Engineering and Applications，2009，45（34）：41-43.

Abstract： According to background of actual application，the processing ability of each equipment is limited generally in the computer and correspondence networks.The article puts forward the k-tree core problem with degree constraint in a tree network，denoted as q-DTC（k）（Degree constrained Tree Core）.With the method of dynamic programming，adopting optimizing principle，it finds out local rooted cores.Then making use of greedy idea，the branch of node which is dissatisfied with degree constraint must be deleted.It takes $O(kn)$ and $O(\max\{n\log n,kn\})$ time algorithm for unweighted tree and weighted tree.
Key words： tree core problem；dynamic programming；local rooted core；greed idea

摘　要：考虑到在实际应用中，由于计算机和通信网络中一般每个设备的处理能力是有限的，在 k-tree core 问题的基础上，提出了同时带有度约束的 k-tree core 问题，即 k-tree core 中的每个节点在子树中的度不超过给定常数 q，记为 q-$DTC(k)$（Degree constrained Tree Core）。利用动态规划的方法，采用最优化原则先找出文中所定义的局部根核集，然后利用贪婪思想对不满足度限制的节点所在的分支加以删减，对无权树和赋权树得到了复杂度分别为 $O(kn)$和 $O(\max\{n\log n,kn\})$多项式时间算法，其中 n 是树的节点数。
关键词：tree core 问题；动态规划；局部根核；贪婪思想
DOI：10.3778/j.issn.1002-8331.2009.34.013　**文章编号：**1002-8331（2009）34-0041-03　**文献标识码：**A　**中图分类号：**O157.5

树状网络上的 core 和 tree core 放置问题产生于计算机网络中分布式数据库以及通信网络中高速骨干网络建设的实际应用。core 或 tree core 问题是选址问题中的一类经典问题，是指问题中的 p 个 median 组成的集合 X_p 构成一条路或者一棵树。目前有大量的文章研究了该问题，很多人在 core 或 tree core 问题上加上约束，得到该问题的新模型，尤其是对 k-tree core 问题及带半径约束的 k-tree core 问题（(k,l)-tree core 问题）已有很好的结果[1-9]。

张固在文[10]中考虑到由于计算机和通信网络中一般每个设备的处理能力是有限的，提出了带度约束的 tree core 问题（Degree constrained Tree Core）要求在树状网络中选择一棵子树，它的每个节点在子树中的度数不超过给定常数 q，记为 q-DTC，并给出了性能优越的多项式时间算法，其复杂度为 $O(n^2)$。在此基础上提出了同时带有度和叶子总数约束的 tree core 问题，记为 q-$DTC(k)$，即子树 tree core 满足叶子总数不超过常数 k，每个节点在子树的度不超过常数 q。

1　模型

设 $T=(V,E)$是一棵具有 n 个节点的无权树，$P_{u,v}$ 表示连接任意两点u,v之间的唯一路径，任意两点u,v之间的距离为 $P_{u,v}$ 路上边的条数，记为 $d(u,v)$。给定子树 $S\subset T$，定义点 u 到 S 的距离为：

$$d(u,S)=\min_{v\in V(S)}d(u,v)\tag{1}$$

T 中所有节点到 S 的距离和为：

$$D(S)=\sum_{v\in V}d(v,S)\tag{2}$$

若 $S=\{u\}$时，则

$$D(S)=D(u)=\sum_{v\in V}d(v,u)\tag{3}$$

令 $\deg(v)$表示节点 v 在树 T 中的度数，$\deg_S(v)$表示节点 v

基金项目：浙江省自然科学基金（the Natural Science Foundation of Zhejiang Province of China under Grant No.y606026）；杭州电子科技大学科学研究基金（the Research Science Foundation of Hangzhou Dianzi University，No.KYF091507018）。
作者简介：杨建芳（1978-），女，讲师，研究方向：组合优化、网络优化；刘建贞（1979-），女，讲师，研究方向：非线性规划、网络优化；黄孙琴（1981-），女，助教，研究方向：组合优化、网络优化。
收稿日期：2008-07-08　**修回日期：**2008-10-14

在子树 S 中的度数,$m(S)$表示子树 S 的叶子总数。对于给定的正整数 $q(q\geqslant 2)$,$k(k\geqslant 2)$,q-$DTC(k)$问题是在树 T 中找一棵子树 S^*,使得

$$D(S^*)=\min_{S\subseteq T}\{D(S)|m(S)\leqslant k \text{ 且 } \deg_S(v)\leqslant q,\forall v\in V(S)\} \quad (4)$$

若 $m(S^*)=2$,则称 S^* 为树 T 的 core,是树中的一条路。

该问题的算法是基于文献[7]关于 k-tree core 的算法,先找出树 T 的 core,其所需要的时间为 $O(n)$,再从树 T 的 core 出发,逐步添加 $i(i\leqslant k-2)$条路,同时满足度约束。

2 无权树上 q-$DTC(k)$问题

2.1 相关知识

设 S 是树 T 的一棵子树,若 S 与给定的路 P 相交于节点 v,即满足 $P\cap S=\{v\}$,则定义:

$$DS(v,P)=D(S)-D(S\cup P) \quad (5)$$

表示子树 S 在 v 点处,增加路 P 所节省的费用。

对于树 T 的某一子树 S 和节点 $v\notin V(S)$,设 u 是与 v 相邻的节点,满足 $d(u,S)=d(v,S)-1$。若路 P 满足 $DS(u,P)=\max_{w\in V_S(v)}\{DS(u,P_{u,w})\}$(其中 $V_S(v)=\{x|P_{v,x}\cap S=\emptyset$ and $d(x,S)\geqslant d(v,S)\}$),则把 P 称为子树 S 在 v 点处的局部根核(local rooted core),记为 $LRC(v,S)$。对于给定的子树 $S\subset T$,如果存在一条边 (u,v),满足 $u\in V(S)$且 $v\notin V(S)$,则称 v 与 S 相邻,用 $N(S)$表示树 T 中所有与 S 相邻的节点构成的集合。由 $LRC(v,S)$的定义及文[8]中的分析,则有以下定理。

定理 1[7] 设 S 是树 T 的一个子树,设 v 满足 $DS(LRC(v,S))=\max_{w\notin V(S)}\{DS(LRC(w,S))\}$,则 $v\in N(S)$。

在构造 k-tree core 问题解的过程中需要用到下述定理。

定理 2[3] 对于任意 k-tree core $S\neq T$,则存在一个$(k+1)$-tree core S',满足 $S\subset S'$。

利用该定理可以从 k-tree core 出发得到一个 $(k+1)$-tree core,结合定理 1 有以下推论。

推论 1[7] 对于任意 k-tree core $S\neq T$,设 P 是子树 S 在 v 点处的局部根核(local rooted core),并且满足 $DS(LRC(v,S))=\max_{w\in N(S)}\{DS(LRC(w,S))\}$,则 $S\cup P$ 是$(k+1)$-tree core。

由推论 1 知,如果已知 k-tree core S,求$(k+1)$-tree core 只需要从 S 出发找到一条路径 P 满足 $DS(LRC(v,S))=\max_{w\in N(S)}\{DS(LRC(w,S))\}$即可。

A.Shioura 和 Uno 在文[7]中给出了对于给定的 S,所有 $v\in N(S)$,可在线性时间求解 $LRC(v,S)$。

2.2 q-$DTC(k)$问题的一个算法

在树 T 中,对于子树 S 和节点 $u\in V(S)$,定义:

$$N(u)=\{v|P_{u,v}\cap S=\{u\} \text{ and adjacent to } u\} \quad (6)$$

由 $N(S)$的定义知,有 $N(S)=\cup_{u\in V(S)}N(u)$,用 LS 表示所有局部根核(local rooted core)所构成的集合。下面给出求解 q-$DTC(k)$问题的动态规划算法,在算法中先找出树 T 的 $core(C)$,自树 T 的 $core(C)$开始,找出所有与之相邻的节点 v 所对应 $LRC(v,C)$,令 $S:=\cup_{v\in N(C)}LRC(v,C)\cup C$,得到一棵新的子树 S,然后对子树 S 重复上述过程,找出所有局部根核(local rooted core),即 LS,用 $L_j(j=1,2,\cdots,m-2)$表示 LS 中的元素(其中 m 表示树 T 中的叶子的总个数)。

算法 1 Find LS

Input: a tree T

Output: LS

begin $S:=C$, $LS:=\emptyset$

  while $m(S)<m$, do

    for all $v\in N(S)$, do

      find $LRC(v,S)$

      $LS:=\cup_{v\in N(S)}LRC(v,S)\cup LS$

      $S:=S\cup LS$

      compute $DS(L_j)$, $\forall L_j\in LS$

    end do

  end while

output LS

end

类似于文献[7]中的分析,可知算法 1 在 $O(n)$时间内求得局部根核集 LS。

现在考虑在集合 LS 中,采取贪婪的思想,从 $core(C)$出发,按照某种“利益”最大化(即 $DS(L_j)$最大化)的方式逐步选取 LS 中的元素,在选择过程中,同时考虑度的约束。

事实上,q-$DTC(k)$问题不一定能找到叶子总数恰好为 k 的解。

如图 1 所示:取 $q=3$,$k=5$ 时,如果要满足度的限制,子树的叶子数达不到 5,所以在设计以下算法的过程中,不只是以树的总叶子数的约束 k 作为终止条件。由定理 1 知,肯定存在节点 $v\in N(S)$,满足 $DS(LRC(v,S))=\max\{DS(L_j)|L_j\in LS\}$,因此在算法设计中只需逐步考虑与 S 相邻的节点。

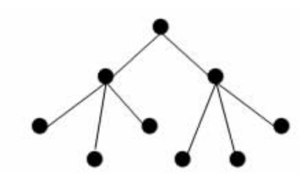

图 1 树网络

算法如下:

算法 2 Find q-$DTC(k)$

Input: a tree T, degree constraint q and leaf constraint k.

Output: a subtree S^* of T, and $D(S^*)$

Step 1: Find a core C

Step 2: Find LS

Step 3:

begin $S^*:=C$, $m(S^*)=2$, $D(S^*)=D(C)$, $LS:=LS$

  while $m(S^*)<k$ and $LS\neq\emptyset$ do

    find $v_i\in N(S^*)$ with

      $DS(LRC(v_i,S^*))=\max\{DS(L_j)|L_j\in LS\}$

      and $\exists u\in N(S^*)$ with $v_i\in N(u)\subset N(S^*)$

    if $\deg_{S^*}(u)+1\leqslant q$ then

        $D(S^*):=D(S^*)-DS(LRC(v_i,S^*))$

        $S^*:S^*\cup LRC(v_i,S^*)$

        $LS:LS-LRC(v_i,S^*)$

    else

        $D(S^*):=D(S^*)$

        $S^*:=S^*$

        $LS:=LS-\{L_j|L_j\subseteq T_{S^*}^{v_i}\}$

    end if

  end while

output a subtree S^* of T, and $D(S^*)$

end

定理3 算法2得到$q\text{-}DTC(k)$问题的最优解。

证明 当$S^*=C$显然是度不超过q，叶子约束为$k=2$时的最优解。

假设S是$q\text{-}DTC(k-1)$最优解，如果对于S中的每一个节点u都已达到度的约束，即$\deg_S(u)=q$，$\forall u\in S$，则S也是$q\text{-}DTC(k)$问题的最优解。否则的话考虑$N(S)$中点的节省费用，由定理1知，肯定存在节点$v\in N(S)$，满足$DS(LRC(v,S))=\max\{DS(L_j)|L_j\in LS\}$，不妨设节点$v_i\in N(S)$满足上述条件，并且存在$u\in V(S)$，使得$v_i\in N(u)\subset N(S)$。由$DS(v,P)=D(S)-D(S\cup P)$知，显然有

$$D(S\cup LRC(v_i,S))\leqslant D(S\cup LRC(v_j,S)),j\neq i$$

如果$\deg_S(u)+1\leqslant q$，由推论1知$S^*=S\cup LRC(v_i,S)$为$q\text{-}DTC(k)$问题的最优解。

如果$\deg_S(u)+1>q$，则节点u在T中的度$\deg_S(u)>q$，不妨设u在T_S中的子节点为$w_1,w_2,\cdots,w_{\deg(u)-1}$，并按$w_i$所在的局部根核的节省费用从大到小排序，则$S$包含了子节点$w_1,w_2,\cdots,w_{q-1}$所在的分支。显然$w_i(q-1<i\leqslant\deg(u)-1)$所在的分支被删减掉要优于$w_i(i=1,2,\cdots,q-1)$所在的分支被删减，所以$q\text{-}DTC(k)$问题的最优解$S^*$也包含了$w_1,w_2,\cdots,w_{q-1}$所在的分支。

S中的其他节点u重复上述讨论，可以推出算法2得到$q\text{-}DTC(k)$问题的最优解。

定理4 可在$O(kn)$时间得到无权树上$q\text{-}DTC(k)$问题的最优解。

证明 由文献[1]知，求解树T的core需要时间为$O(n)$，由算法1分析知寻找LS需要时间是$O(n)$。显然算法2可在$O(kn)$内完成，所以可在$O(kn)$时间内求解$q\text{-}DTC(k)$问题。

3 赋权树上的q-$DTC(k)$问题

讨论在赋权树上的$q\text{-}DTC(k)$问题。设$T=(V,E)$是一棵具有n个节点的树，即对于每一个节点$v\in V$，具有非负权重$r(v)$，对于每一条边为$e\in E$，有非负长度$d(e)$。$P_{u,v}$表示连接任意两点u,v之间的唯一路径，任意两点u,v之间的距离为$P_{u,v}$路上边的长度和，记为$d(u,v)=\sum_{e\in P_{u,v}}d(e)$。给定子树$S\subset T$，定义点$u$到$S$的距离为：

$$d(u,S)=\min_{v\in V(S)}d(u,v) \tag{7}$$

T中所有节点到S的距离加权和为：

$$D(S)=\sum_{v\in V}r(v)d(v,S) \tag{8}$$

不同于无权树上的$q\text{-}DTC(k)$问题，在计算和寻找LS中元素时需要考虑树T中边和节点的权重，而存储边和节点权重所需要的时间是$O(\log n)$，因此算法2实际需要时间为$O(n\log n)$，所以按2.2节中的分析，赋权树上的$q\text{-}DTC(k)$问题可在$O(\max\{n\log n,kn\})$时间内求解。

参考文献：

[1] Morgan C A，Slater J P.A linear algorithm for a core of a tree[J].J Algorithms，1980，1：247-258.

[2] Becker R I.Inductive algorithms on finite trees[J].Questiones Math，1990，13：165-181.

[3] Peng S，Stephens A B，Yesha Y.Algorithms for a core and k-tree core of a tree[J].J Algorithms，1993，15：143-159.

[4] Minieka E，Patel N H.On finding the core of a tree with a specified length[J].J Algorithms，1983，4：345-352.

[5] Peng S，Lo W.Efficient algorithms for finding a core of a tree with a specified length[J].J Algorithms，1996，20：445-458.

[6] Beckera R I，Chang Y I.Finding the l-core of a tree[J].Discrete Applied Mathematics，2002，118：25-42.

[7] Shioura A，Uno T.A linear time algorithm for finding a k-tree core[J].J Algorithms，1997，23：281-290.

[8] Becker R I，Lari I，Storchi G，et al.Efficient algorithms for finding the (k,l)-core of tree networks[J].Networks，2002，40(4)：208-251.

[9] Wang B F，Peng S.Efficient algorithms for a constrained k-tree core problem in a tree network[J].Journal of Algorithms，2006，59：107-124.

[10] 张固.若干网络定位问题的算法研究[D].杭州：杭州电子科技大学，2004.

（上接31页）

最后，证明M上$(\in,\in\vee q)$-fuzzy关联滤子的扩张定理：

定理13 设μ是M的$(\in,\in\vee q)$-fuzzy关联滤子，ρ是M的$(\in,\in\vee q)$-fuzzy滤子。如果$\forall x\in M$有$\mu(x)\leqslant\rho(x)$且$\mu(1)=\rho(1)$，则ρ也是M的$(\in,\in\vee q)$-fuzzy关联滤子。

证明 设$x,y\in M$，令$u=x\to(x\to y)$，则由u-(3)和u-(6)可得

$$x\to(x\to(u\to y))=u\to(x\to(x\to y))=1$$

因为μ是M的$(\in,\in\vee q)$-fuzzy关联滤子，所以由定理9(2)和$\mu(1)=\rho(1)$得：

$$\mu(x\to(u\to y))\geqslant\min\{\mu(x\to(x\to(u\to y))),0.5\}=\min\{\mu(1),0.5\}=\min\{\rho(1),0.5\}$$

又$\forall x\in M$有$\mu(x)\leqslant\rho(x)$，故$\rho(x\to(u\to y))\geqslant\mu(x\to(u\to y))\geqslant\min\{\rho(1),0.5\}$。于是由$\rho$是$M$的$(\in,\in\vee q)$-fuzzy滤子便得

$$\begin{aligned}\rho(x\to y)\geqslant&\min\{\rho(u),\rho(u\to(x\to y)),0.5\}=\\&\min\{\rho(u),\rho(x\to(u\to y)),0.5\}\geqslant\\&\min\{\rho(u),\min\{\rho(1),0.5\},0.5\}=\\&\min\{\rho(u),\rho(1),0.5\}=\\&\min\{\rho(u),0.5\}=\min\{\rho(x\to(x\to y)),0.5\}\end{aligned}$$

所以由定理9(2)知ρ是M的$(\in,\in\vee q)$-fuzzy关联滤子。

参考文献：

[1] Zadeh L A.Fuzzy sets[J].Information Control，1965，8：338-353.

[2] Xu Y，Ruan D，Qin K Y，et al.Lattice-valued logic[M].Berlin：Springer，2004.

[3] Hajek P.Metamathematics of fuzzy logic[M].Dordrecht：Kluwer Academic Publishers，1998.

[4] 刘春辉，徐罗山.Fuzzy蕴涵代数的MP滤子[J].模糊系统与数学，2009，23(2)：1-6.

[5] 王国俊.非经典数理逻辑与近似推理[M].北京：科学出版社，2003.

[6] 何华灿.泛逻辑学原理[M].北京：科学出版社，2001.

[7] 罗敏霞，何华灿.一种泛逻辑代数系统[J].计算机科学与应用，2005，41(14)：21-22.

[8] 肖云萍，邹庭荣.泛逻辑学中UB代数系统的滤子与商代数[J].计算机工程与应用，2007，43(35)：90-92.

[9] 肖云萍，邹庭荣.泛逻辑学UB代数系统的fuzzy滤子[J].计算机科学与探索，2008，2(2)：212-216.