限制树宽图上的有界聚类

李曙光[1] 周 彤[2]
(山东工商学院计算机科学与技术学院 烟台 264005)[1] (山东工商学院研究生处 烟台 264005)[2]

摘 要 有界聚类问题源于IBM研究院开发的一个分布式流处理系统,即S系统。问题的输入是一个点赋权和边赋权的无向图,并指定若干个称为终端的顶点。称顶点集合的一个子集为一个子类。子类中所有顶点的权和加上该子类边界上所有边的权和称为该子类的费用。有界聚类问题是要得到所有顶点的一个聚类,要求每个子类的费用不超过给定预算B,每个子类至多包含一个终端,并使得所有子类的总费用最小。对于限制树宽图上的有界聚类问题,给出了拟多项式时间精确算法。利用取整的技巧对该算法进行修正,可在多项式时间之内得到$(1+\varepsilon)$-近似解,其中每个子类的费用不超过$(1+\varepsilon)B$,ε是任意小的正数。如果进一步要求每个子类恰好包含一个终端,则所给算法可在多项式时间之内得到$(1+\varepsilon)$-近似解,其中每个子类的费用不超过$(2+\varepsilon)B$。

关键词 近似算法,流处理,有界聚类,限制树宽图,动态规划
中图法分类号 TP301 **文献标识码** A

Bounded Clustering on Bounded Tree-width Graphs

LI Shu-guang[1] ZHOU Tong[2]
(College of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai 264005, China)[1]
(Postgraduate Office, Shandong Institute of Business and Technology, Yantai 264005, China)[2]

Abstract The bounded clustering problem is motivated by system S, a distributed stream processing system developed at IBM Research. Input to the problem is an undirected graph with vertex and edge weights and a subset of vertices called terminals. A cluster is a subset of the vertices. The cost of a cluster is defined as the total vertex weight in the cluster plus the total edge weight at the boundary of the cluster. The goal of the problem is to partition the vertices into clusters of cost at most a given budget B such that each cluster contains at most one terminal and the total cost of the clusters is minimized. For the problem on graphs of bounded tree-width, a pseudo-polynomial time exact algorithm was presented, which can be modified via rounding to yield a $(1+\varepsilon)$-approximation in polynomial time violating the given budget by a $1+\varepsilon$ factor, where $\varepsilon>0$ can be made arbitrarily small. For a variant of the problem on graphs of bounded treewidth where each cluster contains exactly one terminal, a polynomial time algorithm was presented that yields a $(1+\varepsilon)$-approximation violating the budget by a $2+\varepsilon$ factor.

Keywords Approximation algorithms, Stream processing, Bounded clustering, Bounded tree-width graphs, Dynamic programming

1 引言

流数据处理是近年来计算机研究领域的一大热点。在现实生活中存在着这样一类系统,外界连续不断地发送数据到系统中,要求系统能够快速地对其进行响应并即时输出相应的结果[1]。典型的数据流应用包括网络路由、入侵检测、传感器网络、股票分析、交通管理、移动通信、环境监测、健康状况监控、电子商务交易信息以及数字化战场等[2]。

本文研究的有界聚类问题[3]源于IBM研究院开发的一个分布式流处理系统,即S系统,该系统可用于大量数据的实时监控和分析处理[4]。系统中的应用可以抽象成一个图,其顶点代表处理节点,边代表数据流。处理节点自身有计算需求,要占用CPU时间,这可以用图中每个顶点有一个非负权表示。处理节点相互之间要传递数据,从而产生通信开销,这可以用图中每条边有一个非负权表示。若干个处理节点可以聚合为一个子类。同一子类中处理节点之间的消息传递可以转化为内部函数调用,因此所产生的通信开销可忽略不计。要把处理节点归为不同的子类,使得每个子类的CPU占用时间(该子类中所有处理节点的CPU占用时间加上该子类与其它子类之间的通信开销)不超过一个指定值,并且使得所有子类的CPU占用总时间最小化。

应用有时会有一些附加的资源约束。比如,某些处理节

到稿日期:2010-11-25 返修日期:2011-03-24 本文受国家自然科学基金(60970105),国家自然科学基金青年基金(11161035),山东省信息产业发展专项资金项目(2008X00039),山东省软科学研究计划(2010RKGA1057,2011RKGB5040)资助。
李曙光(1970—),男,博士,副教授,CCF会员,主要研究方向为图论、组合最优化等,E-mail:sgliytu@hotmail.com;周 彤(1969—),女,工程师,主要研究方向为计算机辅助技术。

点可能频繁使用某种特定硬件(例如网卡),若归为一类,则会造成系统性能下降,因此,这些处理节点应分属不同的子类。可指定一个集合 D,要求每个子类至多包含 D 中的一个节点。

受到这一类应用的启发,Khandekar 等人[3]提出并研究了如下的有界聚类问题。给定有 n 个顶点的无向图 $G=(V,E)$,G 中顶点 v 的权重是 w_v,边 e 的权重是 w_e。指定终端的集合为 $D\subseteq V$。设 B 表示一个给定的非负整数预算。顶点集合的一个子集称为一个子类。对子类 $S\subseteq V$,令 $\delta(S)$ 表示恰好有一个端点在 S 中的边的集合。令 $w(S)=\sum_{v\in S}w_v$,$w(\delta(S))=\sum_{e\in\delta(S)}w_e$。子类 S 中所有顶点的权和加上 S 边界上所有边的权和称为 S 的费用,记为 $C(S)$。即:$C(S)=w(S)+w(\delta(S))$。有界聚类问题是要得到 $V(G)$ 的一个聚类 $S_1,S_2,\cdots,S_k$,要求每个子类的费用不超过 B 且至多包含一个终端,并使得两端点不在同一子类中的所有边的权和 $\frac{1}{2}\sum_{i=1}^{k}w(\delta(S_i))$ 最小。或等价地,使得所有子类的总费用 $\sum_{i=1}^{k}w(\delta(S_i))+\sum_{v\in V}w_v$ 最小。

各子类费用不超过 B 且至多包含一个终端的聚类称为可行聚类。很显然,在一个可行聚类中,子类的数目 k 不会小于终端的数目 $|D|$。可能会出现 $k>|D|$ 的情况,因为 k 并不是输入参数,而是由算法决定的。

在文献[3]中,Khandekar 等人首先研究了有界聚类问题的可行性形式,给出了一个多项式时间算法,即不考虑子类总费用的最小化,只研究是否存在可行聚类,若存在则给出一个可行聚类。这一结果可推广到目标函数为任意的对称次模函数。接着证明了有界聚类问题是 NP 难解的,并设计了一个算法,它可在多项式时间之内得到 $O(\log^2 n)$-近似解,其中任一子类的费用不超过 $O(\log n)\cdot B$。

本文研究限制树宽图上的有界聚类问题,以期得到更好的算法,目标函数是最小化所有子类的总费用。关于限制树宽图的详细介绍见文献[5,6]。我们给出了求解限制树宽图上的有界聚类问题的一个拟多项式时间精确算法。利用取整的技巧对该算法进行修正,可在多项式时间之内得到一个 $(1+\varepsilon)$-近似解,其中每个子类的费用不超过 $(1+\varepsilon)B$,ε 是任意小的正数。如果要求每个子类恰好包含一个终端,则所给算法可在多项式时间之内得到 $(1+\varepsilon)$-近似解,其中每个子类的费用不超过 $(2+\varepsilon)B$。在文献[7]中,我们给出了求解树图上有界聚类问题的拟多项式时间精确算法和多项式时间近似算法,可视为本文所给算法的特殊情形,因为树图是树宽为 1 的限制树宽图。

本文第 2 节研究限制树宽图上的有界聚类问题,给出了拟多项式时间精确算法,并说明如何得到多项式时间近似算法;第 3 节研究限制树宽图上有界聚类问题的一个变形,要求每个子类恰好包含一个终端,给出了多项式时间近似算法;最后指出了可供进一步研究的工作。

2 限制树宽图上的有界聚类

图 $G=(V,E)$ 的树分解是指有序对 $(\{X_i\mid i\in I\},T=(I,F))$,其中分解树 T 的每个结点 $i\in I$ 对应一个子集 $X_i\subseteq V$,称 X_i 为结点 i 的袋子。树 T 和它的袋子集合 $\{X_i\mid i\in I\}$ 必须满足如下 3 个条件:

图 G 的每个顶点至少属于某个袋子,即 $\bigcup_{i\in I}X_i=V$;

对于图 G 的每条边,至少有某个袋子包含该边的两个端点,即对于所有的 $(v,w)\in E$,存在 $i\in I$ 使得 $v,w\in X_i$;

设 i_1,i_2 和 i_3 是树 T 的 3 个结点,如果 i_2 在 T 中从 i_1 到 i_3 的路上,则 $X_{i_1}\cap X_{i_3}\subseteq X_{i_2}$。

树分解的宽度定义为 $\max\{|X_i|:i\in I\}-1$。图 G 的树宽就是图 G 的任何树分解中最小的宽度。若图 G 的树宽限制为某个常数,则称图 G 为限制树宽图。

本节研究限制树宽图上的有界聚类问题,将给出拟多项式时间精确算法和多项式时间近似算法。

设图 G 是有 n 个顶点的限制树宽图,其树宽为常数 b。可在线性时间之内求出图 G 的宽度为 b 的树分解[8]。为便于叙述和分析算法,我们进一步在线性时间之内将该树分解转化为一个宽度为 b 的良好的树分解,其中分解树 T 的结点数不超过 $4n$[9]。

在一个良好的树分解中,T 是一棵根树,并且每个结点 $i\in I$ 属于如下 4 类之一:

叶子结点:结点 i 是树 T 的一片叶子并且 $|X_i|=1$;

引入结点:结点 i 有一个孩子 j 并且存在顶点 $v\in V$ 使得 $X_i=X_j\cup\{v\}$;

遗忘结点:结点 i 有一个孩子 j 并且存在顶点 $v\in V$ 使得 $X_j=X_i\cup\{v\}$;

联合结点:结点 i 有两个孩子 j_1 和 j_2 并且 $X_i=X_{j_1}=X_{j_2}$。

令 T_i 表示 T 的以结点 i 为树根的子树。令 G_i 表示图 G 中对应于 T_i 中的结点的所有顶点的导出子图,则 T_i 是 G_i 的分解树。

算法将沿着树 T 按自底向上的方式依次处理每个结点。设当前正在处理的树 T 的结点为 i。如果事先知道图 G 上有界聚类问题的最优聚类为 $S_1,S_2,\cdots,S_k$,则可记 $S_l\cap X_i=H_l,l=1,2,\cdots,k$。不失一般性,可设 $H_1,\cdots,H_q$ 为非空集合,而 $H_{q+1},\cdots,H_k$ 为空集,$q\leqslant k$ 且 $q\in[1,\cdots,b+1]$。显然,$H_1,\cdots,H_q$ 构成 X_i 的一个聚类。当然,事先并不知道图 G 上有界聚类问题的最优聚类,但是可以枚举出 X_i 所有可能的聚类,其数目不超过常数 $(b+1)^{b+1}$。在这些可能的聚类中,一定有一个恰好是 $H_1,\cdots,H_q$。

考虑如前所述的图 G 上有界聚类问题的最优聚类 $S_1,S_2,\cdots,S_k$。这一聚类在图 G_i 上的限制可视为是对 X_i 的聚类 $H_1,\cdots,H_q$ 的一个扩充。设该 $V(G_i)$ 的聚类中包含 H_l 的子类为 $H_l^i,l=1,2,\cdots,q$。令 B_l 表示 H_l^i 的费用(在 G_i 上的限制)的一个上界,$B_l\in\{1,2,\cdots,B\}$。令二元变量 t_l 表示如下含义:$t_l=0$ 时,H_l^i 不允许包含终端,$t_l=1$ 时没有这一限制(此时 H_l^i 可包含终端,也可不包含终端)。

结点 i 的动态规划表单 $C(i,\{H_1,\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ 储存着 $V(G_i)$ 的满足 $H_1,\cdots,H_q,B_1,\cdots,B_q$ 和 $t_1,\cdots,t_q$ 所规定的所有约束条件的最优聚类的费用。显然,结点 i 的不同动态规划表单至多有 $2^{b+1}\cdot(b+1)^{b+1}\cdot B^{b+1}$ 个。在给出结点 i 的所有后代的动态规划表单后,可计算结点 i 的动态规划表单。更明确地讲,我们将利用结点 i 的所有孩子的动态规划表单来计算结点 i 的动态规划表单。分情形计算

如下。

如果 i 是 T 的一个叶子结点，则有 $|X_i|=1$，可设 $X_i=\{v\}$。分为两种情形：

若顶点 v 不是一个终端(即 $v\notin D$)，则

$$C(i,\{v\},B_1,0)=C(i,\{v\},B_1,1)=\begin{cases}+\infty, & B_1<w_v\\ w_v, & B_1\geqslant w_v\end{cases}$$

若顶点 v 是一个终端，则

$C(i,\{v\},B_1,0)=+\infty$

$$C(i,\{v\},B_1,1)=\begin{cases}+\infty, & B_1<w_v\\ w_v, & B_1\geqslant w_v\end{cases}$$

如果 i 是 T 的一个引入结点，则 i 有一个孩子 j 并且存在顶点 $v\in V$ 使得 $X_i=X_j\cup\{v\}$。图 G_i 可看作是由图 G_j 添加顶点 v 和若干条从 v 到 X_j 中的顶点的边构成。顶点 v 和 $V(G_j)-X_j$ 中的所有顶点都不相邻。

设 X_j 的一个聚类是 $H_1,\cdots,H_q$，$q\leqslant b$。将顶点 v 放入该聚类的某个子类中或单独作为一个子类，就得到 X_i 的一个聚类，故 X_i 的聚类形如 $H_1,\cdots,H_p\cup\{v\},\cdots,H_{q+1}$，意味着顶点 v 被放入第 p 个子类，$p\in\{1,2,\cdots,q+1\}$。若顶点 v 单独作为一个子类，则 $H_{q+1}=\{v\}$，否则 $H_{q+1}=\phi$。

记 $C_l(v)=\sum_{u\in H_l}w_{vu}$，$l\in\{1,2,\cdots,q+1\}$ 且 $l\neq p$。记 $C_p(v)=w_v+\sum_{l\neq p}C_l(v)$。

若顶点 v 不是一个终端，则

$$C(i,\{H_1,\cdots,H_p\cup\{v\},\cdots,H_{q+1}\},\{B_1,\cdots B_p,\cdots,B_{q+1}\},\{t_1,\cdots 0,\cdots,t_{q+1}\})=\sum_{l=1}^{q+1}C_l(v)+C(j,\{H_1,\cdots,H_p,\cdots,H_q\},\{B_1-C_1(v),\cdots,B_p-C_p(v),\cdots,B_q-C_q(v)\},\{t_1,\cdots 0,\cdots,t_q\})$$

$$C(i,\{H_1,\cdots,H_p\cup\{v\},\cdots,H_{q+1}\},\{B_1,\cdots,B_p,\cdots,B_{q+1}\},\{t_1,\cdots 1,\cdots,t_{q+1}\})=\sum_{l=1}^{q+1}C_l(v)+C(j,\{H_1,\cdots,H_p,\cdots,H_q\},\{B_1-C_1(v),\cdots,B_p-C_p(v),\cdots,B_q-C_q(v)\},\{t_1,\cdots 1,\cdots,t_q\})$$

若顶点 v 是一个终端，则

$$C(i,\{H_1,\cdots,H_p\cup\{v\},\cdots,H_{q+1}\},\{B_1,\cdots B_p,\cdots,B_{q+1}\},\{t_1,\cdots 0,\cdots,t_{q+1}\})=+\infty$$

$$C(i,\{H_1,\cdots,H_p\cup\{v\},\cdots,H_{q+1}\},\{B_1,\cdots B_p,\cdots,B_{q+1}\},\{t_1,\cdots 1,\cdots,t_{q+1}\})=\sum_{l=1}^{q+1}C_l(v)+C(j,\{H_1,\cdots,H_p,\cdots,H_q\},\{B_1-C_1(v),\cdots,B_p-C_p(v),\cdots,B_q-C_q(v)\},\{t_1,\cdots 0,\cdots,t_q\})$$

如果 i 是 T 的一个遗忘结点，则 i 有一个孩子 j 并且存在顶点 $v\in V$ 使得 $X_j=X_i\cup\{v\}$。此时 G_i 和 G_j 是同一个图。

在 X_j 的一个聚类中去掉顶点 v，就得到了 X_i 的一个聚类，设为 $H_1,\cdots,H_q$，$q\leqslant b$。因此，结点 i 的动态规划表单 $C(i,\{H_1,\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ 储存着如下 $q+2B$ 个值中的最小者。

$C(j,\{H_1,\cdots H_p\cup\{v\},\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ $(p=1,2,\cdots,q)$

$C(j,\{H_1\cdots,H_q,\{v\}\},\{B_1,\cdots,B_q,B_{q+1}\},\{t_1,\cdots,t_q,0\})$ $(B_{q+1}=1,\cdots,B)$

$C(j,\{H_1\cdots,H_q,\{v\}\},\{B_1,\cdots,B_q,B_{q+1}\},\{t_1,\cdots,t_q,1\})$ $(B_{q+1}=1,\cdots,B)$

注意到顶点 v 在后续计算中不再出现，并且后 $2B$ 个值中的最小者是

$C(j,\{H_1\cdots,H_q,\{v\}\},\{B_1,\cdots,B_q,B\},\{t_1,\cdots,t_q,1\})$

因此，结点 i 的动态规划表单 $C(i,\{H_1,\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ 储存着如下 $q+1$ 个值中的最小者。

$C(j,\{H_1,\cdots,H_p\cup\{v\},\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ $(p=1,2,\cdots,q)$

$C(j,\{H_1\cdots,H_q,\{v\}\},\{B_1,\cdots,B_q,B\},\{t_1,\cdots,t_q,1\})$

如果 i 是 T 的一个联合结点，则 i 有两个孩子 j_1 和 j_2 并且 $X_i=X_{j_1}=X_{j_2}$。图 G_i 可看作是 G_{j_1} 和 G_{j_2} 的并图。

考虑 X_i 的一个聚类 $H_1,\cdots,H_q$，$q\leqslant b+1$。用 H_l^i 表示 $V(G_i)$ 的聚类中包含 H_l 的子类，$l=1,2,\cdots,q$。H_l^i 的费用在 G_i 上的限制 B_l，由以下两部分构成

$$C_l=\sum_{v\in H_l}w_v+\sum_{v\in H_l}\sum_{z\in X_i\setminus H_l}w_{vz}$$

$$B_l-C_l=\sum_{v\in H_l}\sum_{z\in V(G_i)\setminus H_l^i\setminus X_i}w_{vz}+\sum_{v\in H_l^i\setminus H_l}w_v+\sum_{v\in H_l^i\setminus H_l}\sum_{z\in V(G_i)\setminus H_l^i}w_{vz}$$

只要知道了 $H_1,\cdots,H_q,C_l$ 马上就能计算出来。

注意到 X_i 的聚类 $H_1,\cdots,H_q$ 既是 X_{j_1} 的聚类，也是 X_{j_2} 的聚类。分别用 H_l^{j1} 和 H_l^{j2} 表示 $V(G_{j_1})$ 和 $V(G_{j_2})$ 的聚类中包含 H_l 的子类，$l=1,2,\cdots,q$。令

$$B_{1l}=\sum_{v\in H_l}\sum_{z\in V(G_{j_1})\setminus H_l^{j1}\setminus X_{j_1}}w_{vz}+\sum_{v\in H_l^{j1}\setminus H_l}w_v+\sum_{v\in H_l^{j1}\setminus H_l}\sum_{z\in V(G_{j_1})\setminus H_l^{j1}}w_{vz}$$

$$B_{2l}=\sum_{v\in H_l}\sum_{z\in V(G_{j_2})\setminus H_l^{j2}\setminus X_{j_2}}w_{vz}+\sum_{v\in H_l^{j2}\setminus H_l}w_v+\sum_{v\in H_l^{j2}\setminus H_l}\sum_{z\in V(G_{j_2})\setminus H_l^{j2}}w_{vz}$$

则有：$B_{1l}+B_{2l}=B_l-C_l$。因此，对每一个可能的 B_l-C_l 的值，可以在至多 B 次之内猜出最优的 B_{1l} 和 B_{2l}。我们有：$B_{1l}+C_l=B_l^{j1}$ 和 $B_{2l}+C_l=B_l^{j2}$，这里 B_l^{j1} 和 B_l^{j2} 分别表示 H_l^i 的费用在 G_{j_1} 和 G_{j_2} 上的限制。还有：$B_l^{j1}+B_l^{j2}-C_l=B_l$，$l=1,2,\cdots,q$。

计算结点 i 的动态规划表单 $C(i,\{H_1,\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})$ 的基本公式如下

$$C(i,\{H_1,\cdots,H_q\},\{B_1,\cdots,B_q\},\{t_1,\cdots,t_q\})=C(j_1,\{H_1,\cdots,H_q\},\{B_1^{j1},\cdots,B_q^{j1}\},\{t_1^{j1},\cdots,t_q^{j1}\})+C(j_2,\{H_1,\cdots,H_q\},\{B_1^{j2},\cdots,B_q^{j2}\},\{t_1^{j2},\cdots,t_q^{j2}\})-\sum_{l=1}^{q}C_l$$

如前所述，结点 i 的不同动态规划表单至多有 $2^{b+1}\cdot(b+1)^{b+1}\cdot B^{b+1}$ 个。根据上式，每个动态规划表单的值可通过取至多 $2^{2b+2}\cdot B^{b+1}$ 个值中的最小者来得到。为简化计算，需要作如下几点说明

若某个 H_l 包含至少两个终端，则结点 i 的所有动态规划表单的值均为 $+\infty$。

若 H_l 包含一个终端，则凡是出现 $t_l=0$ 的所有动态规划表单的值均为 $+\infty$，而 $t_l=1$ 只能拆分为 $t_l^{j1}=t_l^{j2}=1$，$l=1,2,\cdots,q$。

若 H_l 不包含终端，则 $t_l=0$ 只能拆分为 $t_l^{j1}=t_l^{j2}=0$，而 $t_l=1$ 只能拆分为 $t_l^{j1}=0$ 且 $t_l^{j2}=1$ 或者 $t_l^{j1}=1$ 且 $t_l^{j2}=0$。

设结点 r 是 T 的树根。在计算出所有其他结点的动态规划表单之后，可计算结点 r 的动态规划表单。注意到 $G_r=(V,E)=G$，因此结点 r 的动态规划表单所有值的最小者就是图 G 上有界聚类问题的最优值。

由于 T 有 $O(n)$ 个结点，故整个算法的时间复杂性为 $O(n \cdot 2^{3b+3} \cdot (b+1)^{b+1} \cdot B^{2b+2})$。

为了得到最优聚类，可应用标准动态规划的附加簿记(additional book-keeping)方法，在给出每个动态规划表单值的同时，储存相应的部分聚类。算法的时间复杂性不变。

综上所述，得到如下定理。

定理 1 存在求解限制树宽图上有界聚类问题的拟多项式时间精确算法，其运行时间为 $O(n \cdot 2^{3b+3} \cdot (b+1)^{b+1} \cdot B^{2b+2})$。

接下来说明如何对这一拟多项式算法进行修正，得到多项式时间近似算法。

令 $\alpha=\frac{\lceil 3n/\varepsilon \rceil}{B}$。置每个顶点 $v\in V$ 的新权 $w'_v=\lfloor \alpha \cdot w_v \rfloor$。置每条边 $e\in E$ 的新权 $w'_e=\lfloor \alpha \cdot w_e \rfloor$。令 $B'=\alpha B=\lceil 3n/\varepsilon \rceil$。显然，若在取整以前的问题中有一个预算为 B 的可行聚类(每个子类的费用不超过 B)，则在新问题(取整以后的问题)中有一个预算为 B' 的可行聚类(每个子类的费用不超过 B')。

对于新问题，我们调用如前所述的算法，其运行时间为 $B'=\lceil 3n/\varepsilon \rceil$ 和 n 的多项式函数，得到一个预算为 B' 的最优聚类。设它的目标值为 OPT'，我们有 $OPT' \leqslant \alpha \cdot OPT$。显然，该聚类也是取整以前的问题的一个聚类。注意到对于 $v\in V$ 和 $e\in E$ 分别有 $w_v \leqslant \frac{w'_v}{\alpha}+\frac{1}{\alpha}$ 和 $w_e \leqslant \frac{w'_e}{\alpha}+\frac{1}{\alpha}$。对于取整以前的问题，所得聚类的目标值至多是

$$\frac{OPT'}{\alpha}+\frac{3n}{\alpha} \leqslant OPT+\varepsilon B \leqslant (1+\varepsilon) \cdot OPT$$

该聚类中每个子类的费用至多是

$$\frac{B'}{\alpha}+\frac{3n}{\alpha} \leqslant B+\varepsilon B=(1+\varepsilon)B$$

这样得到如下定理。

定理 2 存在求解限制树宽图上有界聚类问题的近似算法，该算法可在 $O(\frac{n^{2b+3}}{\varepsilon^{2b+2}} \cdot 2^{3b+3} \cdot 3^{2b+2} \cdot (b+1)^{b+1})$ 时间之内得到 $(1+\varepsilon)$-近似解，其中每个子类的费用不超过 $(1+\varepsilon)B$，其中 ε 是任意小的正数。

3 有界聚类问题的一个变形

本节研究限制树宽图上有界聚类问题的一个变形，要求每个子类恰好包含一个终端。

首先利用上一节给出的算法得到限制树宽图上有界聚类问题的一个 $(1+\frac{\varepsilon}{2})$-近似解(允许有的子类不含终端)，其中每个子类的费用不超过 $(1+\frac{\varepsilon}{2})B$。我们现在描述如何利用它来得到变形问题的一个 $(1+\varepsilon)$-近似解，其中每个子类的费用不超过 $(2+\varepsilon)B$。

若变形问题存在可行解，则它的目标值(该聚类中所有子类的总费用)至多为 $|D| \cdot B$，$|D|$ 表示终端的数目。这意味着我们刚才所得限制树宽图上有界聚类问题的 $(1+\frac{\varepsilon}{2})$-近似解的目标值至多为 $(1+\frac{\varepsilon}{2}) \cdot |D| \cdot B$，因为变形问题的最优解一定是限制树宽图上有界聚类问题的可行解。又由于该 $(1+\frac{\varepsilon}{2})$-近似解中每个子类的费用不超过 $(1+\frac{\varepsilon}{2})B$，我们可以将该解中的所有子类分成 $|D|$ 组，使得每组恰好包含一个终端并且费用至多为 $2 \cdot (1+\frac{\varepsilon}{2})B=(2+\varepsilon)B$。每组作为一个子类，这些组便构成了变形问题的一个 $(1+\varepsilon)$-近似解，其中每个子类的费用不超过 $(2+\varepsilon)B$。

这样我们便得到：

定理 3 存在求解限制树宽图上有界聚类变形问题的近似算法，该算法的运行时间为 $O(\frac{n^{2b+3}}{\varepsilon^{2b+2}} \cdot 2^{3b+3} \cdot 3^{2b+2} \cdot (b+1)^{b+1})$，可得到 $(1+\varepsilon)$-近似解，其中每个子类的费用不超过 $(2+\varepsilon)B$，ε 是任意小的正数。

结束语 本文研究了限制树宽图上的有界聚类问题和该问题的一个变形，给出了拟多项式时间精确算法或多项式时间近似算法。可尝试寻找限制树宽图上有界聚类问题和它的变形问题的真正意义上的 $(1+\varepsilon)$-近似算法(所得聚类中每个子类的费用均不超过 B)，也可以继续研究其他图上的这一问题，给出好的算法。

参 考 文 献

[1] Babeock B, Babu S, Datar M, et al. Models and issues in data stream systems[C]// Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems(PODS'02). New York: ACM Press, 2002: 1-16

[2] 金澈清，钱卫宁，周傲英. 流数据分析与管理综述[J]. 软件学报，2004, 15(8): 1172-1181

[3] Khandekar R, Hildrum K, Parekh S, et al. Bounded size graph clustering with applications to stream processing[C]// Proceedings of the Foundations of Software Technology and Theoretical Computer Science(FSTTCS'09). Kanpur, India, 2009: 275-286

[4] System S stream computing system[OL]. http://www-01.ibm.com/software/data/infosphere/streams

[5] Bodlaender H L, Koster A M C A. Combinatorial optimization on graphs of bounded treewidth[J]. The Computer Journal, 2008, 51(3): 255-269

[6] Kleinberg J, Tardos E. Algorithm design[M]. Boston: Addison-Wesley, 2005

[7] Li Shu-guang, Zhu Da-ming, Xin Xiao. Bounded size graph clustering on trees[C]// Proceedings of 2010 International Forum on Information Technology and Applications(IFITA'10). Kunming, China, 2010: 95-98

[8] Bodlaender H L. A linear time algorithm for finding tree-decompositions of small treewidth[J]. SIAM Journal on Computing, 1996, 25(6): 1305-1317

[9] Kloks T. Treewidth: Computations and approximations [C]// Lecture Notes in Computer Science, vol. 842. Berlin: Springer-Verlag, 1994

[10] ZHANG Liang, Byeons-Seob You, Ge Jun-wei, et al. A graph-based sliding windong multi-join over data stream[J]. Jourmnal of Chongqing University of Posts and Telecommunications: Natural Science, 2007, 19(3): 362-366