ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.7, July 2010, pp.1515−1523
doi: 10.3724/SP.J.1001.2010.03625

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

Multicut 问题参数算法的改进*

刘运龙[1,2], 王建新[1+], 陈建二[1]

[1](中南大学 信息科学与工程学院,湖南 长沙 410083)

[2](湖南师范大学 继续教育学院,湖南 长沙 410013)

Improved Parameterized Algorithm for the Multicut Problem

LIU Yun-Long[1,2], WANG Jian-Xin[1+], CHEN Jian-Er[1]

[1](School of Information Science and Engineering, Central South University, Changsha 410083, China)

[2](School of Continuing Education, Hu'nan Normal University, Changsha 410013, China)

+ Corresponding author: E-mail: jxwang@mail.csu.edu.cn

Liu YL, Wang JX, Chen JE. Improved parameterized algorithm for the Multicut problem. *Journal of Software*, 2010,21(7):1515−1523. http://www.jos.org.cn/1000-9825/3625.htm

Abstract: The Multicut problem is for a given graph and a given collection of terminal pairs to find a vertex set of minimum size such that the two terminals in any pair are not connected after deletion of this vertex set. This problem is NP-hard. Based on the deep analysis of its structural characteristics, employing the strategy of set partition and the improved results of another related problem, this paper proposes a parameterized algorithm of running time $O^*\left(\left\lceil\sqrt{2l}\right\rceil^{2l}4^k\right)$ for the problem, in which l denotes the number of terminal pairs and k denotes the number of removed vertices. This algorithm significantly improves the previous one of running time $O^*(2^{kl}k^k4^{k^3})$.
Key words: Muliticut; Node Multicut; set partition; maximal proper partition

摘 要: Multicut 问题即在一个图上删除最少个数的顶点,使得预先给定的一组顶点对均不连通.该问题是 NP 难的.在深入分析问题结构特点的基础上,运用集合划分策略和相关问题的最新研究结果,对它提出了一种时间复杂度为 $O^*\left(\left\lceil\sqrt{2l}\right\rceil^{2l}4^k\right)$ 的参数化算法,其中,l 为给定的顶点对数目,k 为需删除的顶点个数.该算法明显改进了当前时间复杂度为 $O^*(2^{kl}k^k4^{k^3})$的最好算法.
关键词: Muliticut;Node Multicut;集合划分;极大恰当划分
中图法分类号: TP301 **文献标识码**: A

Multicut 问题即给定一个无向图 G 和其中 l 个顶点对$(s_i,t_i)(1\leqslant i\leqslant l,l\geqslant 3)$,目标是从图 G 中删除最少的顶点

* Supported by the National Natural Science Foundation of China under Grant Nos.60433020, 60773111 (国家自然科学基金); the National Basic Research Program of China under Grant No.2008CB317107 (国家重点基础研究发展计划(973)); the Program for New Century Excellent Talents in University of China under Grant No.NCET-05-0683 (新世纪优秀人才支持计划)

Received 2008-09-12; Accepted 2009-03-31

(或者边),使得任意给定顶点对的两个顶点 s_i 和 t_i 之间均没有路径[1].该问题是图论中一个最基本的优化问题,是研究计算机通信网络可靠性和鲁棒性的一种重要模型,在聚类、超大规模集成电路设计和网络设计等方面有着广泛的实际应用[2,3].Multicut 问题是 NP-难和 Max SNP-难的[1],人们不仅运用近似算法对它进行了大量的研究[1,4,5],而且近年来运用参数计算理论及技术对它进行了一系列研究[6−8].由于边删除可以通过线图转化为顶点删除,本文主要研究 Multicut 问题顶点删除的情况.文献[7]对该问题给出了如下参数化定义:

定义 1(参数化 Multicut)[7].

输入:一个无向图 $G=(V,E)$,G 上 l 个顶点对$(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$和一个正整数 k;

参数:k 和 l.

问题:图 G 中是否存在一个大小最多为 k 的顶点子集 S,使得在图 $G-S$ 上,对于任意 $1\leqslant i\leqslant l$,s_i 和 t_i 之间均没有路径.其中,图 $G-S=(V-S,E-E_S)$,$E_S=\{e|e\in E$ 且 e 至少有一个端点属于 $S\}$.

该问题所求的顶点子集 S 习惯上也称为图 G 上顶点对$(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$的一个大小不超过 k 的分割集.Multicut 问题可进一步分为两种类型[1]:(1) 受限的情况,即限定分割集 S 不允许包含点集$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$中的点;(2) 非受限的情况,即分割集 S 允许包含点集$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$中的点.

Marx 在文献[7]中对参数化 Multicut 问题(含受限和非受限的两种情况)提出了一种时间复杂度为 $O^*(2^{kl}k^k4^{k^3})$ 的固定参数可解算法,这也是当前求解该问题的最好算法.Guo 等人在文献[8]中对一些特殊图(如树、间隔图、树宽度有界的图)上的参数化 Multicut 问题分别进行了研究,但这些结果对于一般图来说并不成立.同时,人们对 Multicut 问题的下述特殊情况也进行了研究.

定义 2(参数化 Node Multicut)[7]. 给定一个图 $G=(V,E)$和其中 l 个不同的点 $t_1,t_2,\ldots,t_l$,目标为判定图 G 中是否存在一个大小不超过 k 的顶点子集 S,使得在图 $G-S$ 中,点集$\{t_1,t_2,\ldots,t_l\}$中任何两个点之间均没有路径.

Marx 在文献[7]中证明了 Node Multicut 问题是固定参数可解的,并对它提出了一个时间复杂度为 $O^*(4^{k^3})$ 的参数化算法.如果将该问题中给定的 l 个不同的点扩展为 l 个互不相交的点集,则得到如下问题:

定义 3(参数化 Node Multicut 扩展)[9]. 给定一个图 $G=(V,E)$和其中 l 个互不相交的点集 $T_1,T_2,\ldots,T_l$,目标为判定图 G 上是否存在一个大小不超过 k 的顶点子集 S,使得在图 $G-S$ 中,点集 $T_1,T_2,\ldots,T_l$ 中任何两个点集之间均没有路径.其中,两个点集 T_i,T_j 之间没有路径是指 T_i 中的任何点与 T_j 中的任何点之间没有路径.

最近,Chen 等人对 Node Multicut 扩展问题进行了深入研究,并在文献[9]中对它提出了一种时间复杂度为 $O^*(4^k)$的固定参数可解算法.

引理 1[9]. 受限的和非受限的 Node Multicut 扩展问题都可在时间 $O(4^kkn^3)$内求解.

本文主要研究定义 1 中的参数化 Multicut 问题(后面简称 Multicut 问题).本文首先根据 Multicut 问题实例的结构特点找到 Multicut 问题和 Node Multicut 扩展问题之间的内在联系,接着运用集合划分策略将判定 Multicut 问题的一个实例是否有解等价地转化为在多个 Node Multicut 扩展问题实例中判定是否至少存在一个实例有解,然后运用文献[9]中求解 Node Multicut 扩展问题的算法对 Multicut 问题提出了一种时间复杂度为 $O^*\left(\left\lceil\sqrt{2l}\right\rceil^{2l}4^k\right)$ 的高效算法(符号$\lceil x\rceil$表示不小于 x 的最小整数,反之,$\lfloor x\rfloor$表示不大于 x 的最大整数).

本文根据$\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\}$中不同点对之间是否存在相同点以进一步将问题进行分类.如果给定的点对$\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\}$中任意两个点对之间都不存在相同的顶点,则称其为严格的 Multicut 问题;反之,称为非严格的 Multicut 问题.下面,我们对这两类情况分别进行讨论.

1 严格 Multicut 问题参数算法的改进

通过观察分析,可以找到 Multicut 问题和 Node Multicut 扩展问题之间的内在联系,将判定 Multicut 问题的一个实例是否有解等价地转化为在多个 Node Multicut 扩展问题实例中判定是否至少存在一个实例有解.由此必须解决两个关键问题:(1) 一个 Multicut 问题实例如何等价地转化为需要判定的多个 Node Multicut 扩展问题实例?(2) 为了提高算法的效率,如何使需要判定的 Node Multicut 扩展问题实例的个数尽可能地少?下面分别讨

论这两个问题.

1.1 Multicut问题与Node Multicut扩展问题之间的联系

为便于分析 Multicut 问题与 Node Multicut 扩展问题之间的联系,首先引述集合划分的概念.设 A 为一个给定的非空集合,如果一个超集合 $P=\{R_1,R_2,\ldots,R_m\}$同时满足下述 3 个条件:

(1) $R_i\subseteq A$ 且 $R_i\neq\varnothing(1\leqslant i\leqslant m)$;

(2) $R_1\cup R_2\cup\ldots\cup R_m=A$;

(3) $R_i\cap R_j=\varnothing$ $(1\leqslant i\leqslant m,1\leqslant j\leqslant m,i\neq j)$,

则称 P 为集合 A 的一个划分.

设$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$为 Multicut 问题的一个实例,并设 S 是该实例的一个分割集.对于集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$来说,在图 $G-S$ 上不同顶点对中的点之间可能存在路径.如果将 C 中相互之间存在路径的顶点合并起来,得到的一些子集就正好构成了集合 C 的一个划分.同时,根据 Multicut 问题的定义可知,同一点对$(s_i,t_i)(1\leqslant i\leqslant l)$中的两个点 s_i 和 t_i 不可能同时属于该划分中的任意一个子集,满足这样一个条件的划分称为$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一个恰当划分.设$\{R_1,R_2,\ldots,R_m\}$为 C 的任意一个恰当划分,则由$\{R_1,R_2,\ldots,R_m\}$可以唯一构造 Node Multicut 扩展问题的一个实例,不妨用$(G,\{R_1,R_2,\ldots,R_m\},k)$来表示.设 $Q=\{P|P$ 为 C 的一种恰当划分$\}$,进一步分析可以得出,Multicut 问题与 Node Multicut 扩展问题之间存在下述联系:

定理 1. $(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集的充分必要条件是 Q 中至少存在 C 的一种恰当划分 $P_0=\{T_1,T_2,\ldots,T_m\}(2\leqslant m\leqslant 2l)$,使得$(G,\{T_1,T_2,\ldots,T_m\},k)$存在一个大小不超过 k 的分割集.

证明:首先证明其充分性.假设 Q 中存在 C 的某一种恰当划分 $P_0=\{T_1,T_2,\ldots,T_m\}(2\leqslant m\leqslant 2l)$,使得实例$(G,\{T_1,T_2,\ldots,T_m\},k)$存在一个大小不超过 k 的分割集 S.根据 Node Multicut 扩展问题的定义可知,在图 $G-S$ 上,集合$\{T_1,T_2,\ldots,T_m\}$中的任意两个不同子集之间都不存在路径.又由$\{T_1,T_2,\ldots,T_m\}$是集合 C 的一种恰当划分可知,任意一个顶点对$(s_i,t_i)(1\leqslant i\leqslant l)$中的两个点 s_i,t_i 一定不在同一个子集 $T_i(1\leqslant i\leqslant m)$中.因此,$s_i$ 和 t_i 之间一定不存在路径.即 S 也是$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$的一个分割集.

接下来证明其必要性.假设实例$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集 S,在图 $G-S$ 上,虽然同一点对的两个点之间不存在路径,但不同点对中的点之间可能存在路径.于是,可以按如下办法来构造 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一个恰当划分.从 C 中的任意一点(不妨假设为 s)出发,在图 $G-S$ 上从 C 中找出所有与 s 有路径可到达的点,并设它们连同点 s 构成集合 T_1.显然,T_1 与 $C-T_1$ 之间不存在路径.接下来,类似地,从 $C-T_1$ 中的任意一点出发(不妨假设为 s'),在图 $G-S$ 上从 $C-T_1$ 中找出所有与 s'有路径可到达的点,并设它们连同点 s'构成集合 T_2.显然,$T_1,T_2,C-T_1-T_2$ 这 3 个子集两两之间都没有路径相连.这个过程继续下去,直到 C 为空,得到的子集依次为 $T_1,T_2,\ldots,T_m$.显然,$\{T_1,T_2,\ldots,T_m\}$是 C 的一个划分.又由 S 是$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$的分割集可知,任意顶点对$(s_i,t_i)(1\leqslant i\leqslant l)$的两个点 s_i 和 t_i 都不可能同时属于某一个子集 $T_i(1\leqslant i\leqslant m)$.因此,$\{T_1,T_2,\ldots,T_m\}$是 C 的一个恰当划分.从前述构造过程可以看出,在图 $G-S$ 上,$\{T_1,T_2,\ldots,T_m\}$中的任意两个子集 $T_i,T_j(i\neq j,1\leqslant i\leqslant m,1\leqslant j\leqslant m)$之间都不存在路径,因此,$S$ 是$(G,\{T_1,T_2,\ldots,T_m\},k)$的一个大小不超过 k 的分割集. □

定理 1 表明,判定 Multicut 问题任意一个实例 I 是否有解,可等价地转化为在$|Q|$个 Node Multicut 扩展问题实例中判定是否至少存在一个实例有解.显然,依次判定$|Q|$个 Node Multicut 扩展问题实例的时间开销是非常大的.下面讨论如何通过只判定其中极少一部分实例就能判定实例 I 真假性的问题.

1.2 集合$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的极大恰当划分

为便于后文表述,下面首先定义极大恰当划分和对应的隐含恰当划分两个重要概念.

定义 4. 设$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$为与顶点对$(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$对应的顶点集合,如果$\{T_1,T_2,\ldots,T_m\}$为$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一种恰当划分,并且其中任意两个子集 $T_i,T_j(i\neq j,1\leqslant i\leqslant m,1\leqslant j\leqslant\leqslant m)$的并集 $T_i\cup T_j$ 至少含有$\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\}$中某一个顶点对的两个顶点,则称$\{T_1,T_2,\ldots,T_m\}$为$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一种极大恰当划分.

从定义 4 可以看出,极大恰当划分中的各子集已经是"极大"的,相互之间不能再进行合并,否则就不构成恰

当划分.

定义 5. 设 $P=\{T_1,T_2,\ldots,T_m\}$为集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一种极大恰当划分,$F(P)=\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\}$为 C 的另一种不同于 P 的恰当划分,如果对于 P 中的任意一个子集 $T_i(1\leqslant i\leqslant m)$,$F(P)$中要么存在一个与 T_i 完全相同的子集 T_i,要么存在 $p(2\leqslant p\leqslant l)$个子集 $T_{i1},T_{i2},\ldots,T_{ip}$,使得 $T_i=\bigcup_{j=1}^{p}T_{ij}$,则称 $F(P)$为 C 的对应于极大恰当划分 P 的一种隐含恰当划分.

显然,一个极大恰当划分可能存在多个对应的隐含恰当划分.例如,设给定 4 个顶点对的集合为$\{(s_1,t_1),(s_2,t_2),(s_3,t_3),(s_4,t_4)\}$,如果设 $A_1=\{t_1,t_3,t_4\}$,$A_2=\{s_2,s_4\}$和 $A_3=\{s_1,t_2,s_3\}$,则 A_1,A_2,A_3 正好是$\{s_1,t_1,s_2,t_2,s_3,t_3,s_4,t_4\}$的一种含有 3 个子集的极大恰当划分.如果设 $A_{11}=\{t_1,t_3\}$,$A_{12}=\{t_4\}$,$A_{31}=\{s_1\}$,$A_{32}=\{t_2,s_3\}$,则恰当划分$\{A_{11},A_{12},A_2,A_3\}$和恰当划分$\{A_1,A_2,A_{31},A_{32}\}$都是与极大恰当划分$\{A_1,A_2,A_3\}$对应的隐含恰当划分.

从定义 4 和定义 5 可以看出,$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的任意一种恰当划分要么是极大恰当划分,要么为与某种极大恰当划分对应的隐含恰当划分.

设 $P=\{T_1,T_2,\ldots,T_m\}$为顶点集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一种极大恰当划分,$F(P)=\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\}$为 C 的对应于 P 的任意一种隐含恰当划分,由 P 和 $F(P)$确定的 Node Multicut 扩展问题的两个实例分别为$(G,\{T_1,T_2,\ldots,T_m\},k)$和$(G,\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\},k)$,则这两个实例之间存在下述联系:

定理 2. 如果实例$(G,\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\},k)$存在一个大小不超过 k 的分割集,则实例$(G,\{T_1,T_2,\ldots,T_m\},k)$必定存在一个大小不超过 k 的分割集.

证明:设 S 为实例$(G,\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\},k)$的一个大小不超过 k 的分割集,根据分割集的定义可知,在图 $G-S$ 上,点集 $T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m$ 两两之间不存在任何路径.假设 S 不是实例$(G,\{T_1,T_2,\ldots,T_m\},k)$的一个分割集,则在图 $G-S$ 上,$\{T_1,T_2,\ldots,T_m\}$中至少存在两个子集 $T_i,T_j(1\leqslant i\leqslant m,1\leqslant j\leqslant m,i\neq j)$,使得 T_i 中的某一点 v 和 T_j 中的某一点 u 之间存在一条路径 Y.由于 $F(P)=\{T_{11},T_{12},\ldots,T_{21},T_{22},\ldots,T_m\}$为 C 的对应于 $P=\{T_1,T_2,\ldots,T_m\}$的一种隐含恰当划分,可知路径 Y 的两个端点 v 和 u 也必定属于 $F(P)$中的两个不同子集.即 $F(P)$中存在两个有路径相连的子集,这就与已知条件相矛盾.因此,S 是实例$(G,\{T_1,T_2,\ldots,T_m\},k)$的一个大小不超过 k 的分割集. □

定理 2 表明,如果由 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的极大恰当划分确定的实例不存在一个大小不超过 k 的分割集,则由相应的隐含划分确定的实例必定不存在一个大小不超过 k 的分割集.因此,在判定 Node Multicut 扩展问题的实例时,只要判定由 C 的极大恰当划分确定的实例.C 的所有极大恰当划分究竟有多少种呢?接下来讨论一个重要观察.

定理 3. 设 C 为顶点对$(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$中 $2l$ 个不同顶点构成的集合,则 C 存在一种含有 $m(m\geqslant 2)$个子集的极大恰当划分的充分必要条件是 $m(m-1)\leqslant 2l$.

证明:首先证明其充分性.假设 m_0 为满足关系式 $m_0(m_0-1)\leqslant 2l$ 的最大整数,对于满足 $2\leqslant m\leqslant m_0$ 的任意一个整数 m,都可以构造出集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的一种含有 m 个非空子集的极大恰当划分.

首先用归纳法证明,对于满足 $2\leqslant m\leqslant m_0$ 的任意一个整数 m,集合 $C'=\{s_1,t_1,s_2,t_2,\ldots,s_q,t_q\}$必定存在一种含有 m 个子集的极大恰当划分,其中,$q=m(m-1)/2$.当 $m=2$ 时,设 $T_1=\{s_1\}$,$T_2=\{t_1\}$,则$\{T_1,T_2\}$构成$\{s_1,t_1\}$的一个极大恰当划分.当 $m\neq 2$ 时,根据归纳法,假设当 $d=m-1$ 时,点集$\{s_1,t_1,s_2,t_2,\ldots,s_p,t_p\}$存在一种含有 $m-1$ 个子集的极大恰当划分$\{T_1,T_2,\ldots,T_{m-1}\}$,其中,$p=(m-1)(m-2)/2$,$T_i(1\leqslant i\leqslant m-1)$含有 $m-2$ 个点.当 $d=m$ 时,对于顶点对$(s_1,t_1),(s_2,t_2),\ldots,(s_p,t_p),(s_{p+1},t_{p+1}),\ldots,(s_q,t_q)$,其中,$q=m(m-1)/2$,首先从点对$(s_{p+1},t_{p+1}),(s_{p+2},t_{p+2}),\ldots,(s_q,t_q)$中各取一点构成集合 T_m,不妨设 $T_m=\{s_{p+1},s_{p+2},\ldots,s_q\}$.又根据归纳假设,$\{s_1,t_1,s_2,t_2,\ldots,s_p,t_p\}$存在一种极大恰当划分$\{T_1,T_2,\ldots,T_{m-1}\}$,于是可将 $m-1$ 个点 $t_{p+1},t_{p+2},\ldots,t_q$ 分别加入到 $T_1,T_2,\ldots,T_{m-1}$ 中,得到 $m-1$ 个各含有 $m-1$ 个点的子集 $T'_1,T'_2,\ldots,T'_{m-1}$,它们连同 T_m 就得到了 m 个各含有 $m-1$ 个点的子集.同时,从该构造过程可以看出,由于 $T_1,T_2,\ldots,T_{m-1}$ 是不能相互合并的,所以 $T'_1,T'_2,\ldots,T'_{m-1}$ 也是不能相互合并的,又 $T_m=\{s_{p+1},s_{p+2},\ldots,s_q\}$取自$(s_{p+1},t_{p+1}),(s_{p+2},t_{p+2}),\ldots,(s_q,t_q)$,因而 T_m 与 $T'_1,T'_2,\ldots,T'_{m-1}$ 也不能合并,因此$\{T'_1,T'_2,\ldots,T'_{m-1},T_m\}$构成了 C'的一种含有 m 个子集的极大恰当划分.

对于集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$来说,如果 $m(m-1)=2l$,上述 C'的极大恰当划分就正好是 C 的极大恰当划分.如果 $m(m-1)<2l$,可以首先按上述过程构造 C'的一种极大恰当划分,然后将剩余点对中的点添加到该划分的 m

个子集中,并且满足同一个点对中的两个点不添加到同一个子集中,这样就得到了 C 的一种含有 m 个子集的极大恰当划分.

接下来证明其必要性.假设 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$ 存在一种含有 m 个子集的极大恰当划分 $\{T_1,T_2,\ldots,T_m\}$,并设 T_i 为 $\{T_1,T_2,\ldots,T_m\}$ 中任意一个固定的子集,T_j 为其余 $m-1$ 子集中的任意一个子集.根据极大恰当划分的定义,T_i 中必定至少存在一点与 T_j 中的对应点构成 $\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\}$ 中的顶点对.又 $\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\}$ 中任意两个点对之间不存在相同顶点,可知 T_i 中的每一个点只能与唯一一个其余子集中的点构成点对.因此,T_i 中至少含有 $m-1$ 个点.同时,由于 $\{T_1,T_2,\ldots,T_m\}$ 中子集的个数为 m,T_i 是其中的任意一个子集,并且 $\{T_1,T_2,\ldots,T_m\}$ 是 C 的一种划分,因此集合 C 中点的个数至少为 $m(m-1)$.故 $m(m-1)\leqslant 2l$. □

推论 1. 设 C 为顶点对 $(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$ 中 $2l$ 个不同顶点构成的集合,则 C 的任意一种极大恰当划分所含有子集的个数不会超过 $\lceil\sqrt{2l}\rceil$.

证明:设 P 为 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$ 的一种含有子集个数最多的极大恰当划分,并设 a 为 P 中子集的个数.根据定理 3 可知,a 为满足不等式 $a(a-1)\leqslant 2l$ 的最大正整数.解这个方程得到 $a=\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor$.接下来证明:对于任意正整数 l,$\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor\leqslant\lceil\sqrt{2l}\rceil$ 恒成立.

对任意给定的正整数 $2l$,总存在一个正整数 m 使得 $(m-1)m\leqslant 2l<m(m+1)$.首先证明当 $(m-1)m\leqslant 2l\leqslant m^2$ 时,$\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor=\lceil\sqrt{2l}\rceil$.当 $2l=(m-1)m$ 时,显然 $\left(1+\sqrt{1+8l}\right)/2=m$.当 $2l=m^2$ 时,$\sqrt{1+8l}=\sqrt{1+4m^2}<2m+1$,即 $\left(1+\sqrt{1+8l}\right)/2<(m+1)$;又由函数 $\left(1+\sqrt{1+8l}\right)/2$ 的单调递增性可知,当 $(m-1)m\leqslant 2l\leqslant m^2$ 时,$m\leqslant\left(1+\sqrt{1+8l}\right)/2<(m+1)$,于是得到 $\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor=m$.类似地,当 $(m-1)m\leqslant 2l\leqslant m^2$ 时,$(m-1)^2<2l\leqslant m^2$,可得 $(m-1)<\sqrt{2l}\leqslant m$,于是 $\lceil\sqrt{2l}\rceil=m$.因此,$\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor=\lceil\sqrt{2l}\rceil$.同理可证,当 $m^2<2l<m(m+1)$ 时,$\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor=\lceil\sqrt{2l}\rceil-1$.故 $\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor\leqslant\lceil\sqrt{2l}\rceil$ 恒成立. □

推论 2. 设 C 为顶点对 $(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)$ 中 $2l$ 个不同顶点构成的集合,$B_m=\{P|P$ 为 C 的一种含有 m 个子集的恰当划分$\}$,$h=\lceil\sqrt{2l}\rceil$,则 $\bigcup_{m=2}^{h}B_m$ 包含了 C 的所有极大恰当划分,并且 $\bigcup_{m=2}^{h}B_m$ 中不同恰当划分的总数有上界 $\lceil\sqrt{2l}\rceil^{2l}$.

证明:显然,集合 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$ 的任意一种极大恰当划分中至少含有 2 个子集.又由推论 1 的证明可知,C 的任意一种极大恰当划分最多含有 $\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor$ 个子集.因此,C 的任一极大恰当划分所含有子集的个数 m 满足关系式 $2\leqslant m\leqslant\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor$.设 $A_m=\{P'|P'$ 为 C 的一种含有 m 个子集的极大恰当划分$\}$,$w=\left\lfloor\left(1+\sqrt{1+8l}\right)/2\right\rfloor$,则 C 的所有极大恰当划分可表示为 $\bigcup_{m=2}^{w}A_m$.又已设 $B_m=\{P|P$ 为 C 的一种含有 m 个子集的恰当划分$\}$,显然,C 的任意一种含有 m 个子集的极大恰当划分必定是 C 的一种含有 m 个子集的恰当划分.因此,$A_m\subset B_m$.相应地,$\bigcup_{m=2}^{w}A_m\subset\bigcup_{m=2}^{w}B_m$.由于已知 $h=\lceil\sqrt{2l}\rceil$,且在推论 1 中已证明 $w\leqslant h$,因此,$\bigcup_{m=2}^{w}A_m\subset\bigcup_{m=2}^{w}B_m\subseteq\bigcup_{m=2}^{h}B_m$,即 $\bigcup_{m=2}^{h}B_m$ 包含了 C 的所有极大恰当划分.

又设 $D_m=\{P''|P''$ 为 C 的一种含有 m 个子集的划分$\}$,则 $\bigcup_{m=2}^{h}B_m\subset\bigcup_{m=2}^{h}D_m$,并且 $\bigcup_{m=2}^{h}D_m$ 中不同划分的总数一定有上界 $\lceil\sqrt{2l}\rceil^{2l}$,因此 $\bigcup_{m=2}^{h}B_m$ 中不同恰当划分的种数有上界 $\lceil\sqrt{2l}\rceil^{2l}$. □

1.3 Multicut问题改进的参数化算法

下面,我们根据前述相关理论对Multicut问题提出一种改进的算法.对于任意一个给定的Multicut问题实例$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$,算法的第 1 步是确定需要判定的 Node Multicut 扩展问题的实例.由于精确求出点集 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的所有极大恰当划分极为困难,又根据推论 2 可知 C 的所有极大恰当划分包含在 $\bigcup_{m=2}^{h} B_m$ 之中,其中,$B_m=\{P|P$ 为 C 的一种含有 m 个子集的恰当划分$\}$,$h=\lceil\sqrt{2l}\rceil$,因此选择由 $\bigcup_{m=2}^{h} B_m$ 中的恰当划分所确定的 Node Multicut 扩展问题实例作为判定的对象;第 2 步是对确定的 Node Multicut 扩展问题的实例依次进行判定求解.下面首先描述如何产生集合 C 的所有含有 m 个子集的恰当划分.

图 1 详细描述了产生 C 的所有含有 m 个子集的恰当划分的算法.为了表述简便,算法中用符号 $P[i]$表示划分 P 中的第 $i(1\leqslant i\leqslant m)$个子集,用 Q 表示存储各种不同划分的集合.算法采用递归技术,首先对两种使递归结束的特殊情况进行处理,然后将一般情况分为 3 种子情况分别进行处理,处理步骤都是首先通过递归过程对集合$C'=\{s_1,t_1,s_2,t_2,\ldots,s_{l-1},t_{l-1}\}$进行恰当划分,然后在 C'的各种恰当划分中配置点对(s_l,t_l)中的两个点.具体来说,算法的第 3.2 步是首先产生 C'的所有含有 $m-2$ 个子集的恰当划分,然后将子集$\{s_l\},\{t_l\}$配置到每种划分中;算法的第 3.3 步是首先产生 C'的所有含有 $m-1$ 个子集的恰当划分,然后将点 s_l 和子集$\{t_l\}$(或者点 t_l 和子集$\{s_l\}$)配置到每种划分中;算法的第 3.4 步是首先产生 C'的所有含有 m 个子集的恰当划分,然后将点 s_l,t_l 配置到每种划分中.显然,每种配置都相应地得到 C 的一种含有 m 个子集的恰当划分.

```
Algorithm PARTITION({s1,t1,s2,t2,…,sl,tl},m)
1.  if m=2l then {P={{s1},{t1},{s2},{t2},…,{sl},{tl}}; Q={P}; return Q }
2.  if m>2l then {Q=∅; return Q }
3.  if 2≤m<2l then
  3.1  Q=∅;
  3.2  if m≥4 then
         Q1=PARTITION({s1,t1,s2,t2,…,sl−1,tl−1},m−2);
         for each partition P in Q1 do {P[m−1]={sl}; P[m]={tl}; Q=Q∪{P}}
  3.3  if m≥3 then
         Q2=PARTITION({s1,t1,s2,t2,…,sl−1,tl−1},m−1);
          for each partition P in Q2 do
             for i=1 to m−1 do
                { R=P; H=P;
                  R[i]=R[i]∪{sl}; R[m]={tl}; Q=Q∪{R};
                  H[i]=H[i]∪{tl}; H[m]={sl}; Q=Q∪{H}}
  3.4  if m≥2 then
         Q3=PARTITION({s1,t1,s2,t2,…,sl−1,tl−1},m);
         if Q3≠∅ then
           for each partition P in Q3 do
               for i=1 to m−1 do
                 for j=i+1 to m do
                      {R=P; H=P;
                       R[i]=R[i]∪{sl}; R[j]=R[j]∪{tl}; Q=Q∪{R};
                       H[i]=H[i]∪{tl}; H[j]=H[j]∪{sl}; Q=Q∪{H}}
  3.5  return Q.
```

Fig.1 An algorithm for outputting all proper m-partitions of $\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$

图 1 生成集合$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的所有含有 m 个子集的恰当划分算法

定理 4. 算法 PARTITION 能够正确生成$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$的所有含有 m 个子集的恰当划分.

证明:设 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$.如果要将 C 划分为 $m=2l$ 个子集,显然其划分方法是唯一确定的,算法的第 1 步正确地返回了这种划分.如果要将 C 划分为 $m>2l$ 个子集,显然是不可能的,因此算法第 2 步返回空集.对于一般情况 $2\leqslant m<2l$,设 $C'=\{s_1,t_1,s_2,t_2,\ldots,s_{l-1},t_{l-1}\}$,由分析可知,$C$ 的含有 m 个子集的恰当划分可在 C'的恰当划分的基础上通过以下 3 种不同途径来产生:

(1) 当 $m\geqslant4$ 时,对于 C'的任意一种含有 $m-2$ 个子集的恰当划分 P_1,只要将子集$\{s_l\}$和$\{t_l\}$同时添加到 P_1 中,

就可以得到 C 的一种含有 m 个子集的恰当划分.因此,算法的第 3.2 步首先通过递归过程产生 C' 的所有含有 $m-2$ 个子集的恰当划分,然后在每种恰当划分中添加子集 $\{s_l\}$ 和 $\{t_l\}$;

(2) 当 $m\geqslant 3$ 时,对于 C' 的任意一种含有 $m-1$ 个子集的恰当划分 P_2,只要将顶点 s_l(或 t_l)插入到 P_2 中的任意一个子集中,然后合并子集 $\{t_l\}$(或者 $\{s_l\}$),即可得到 C 的一种含有 m 个子集的恰当划分.因此,算法的第 3.3 步首先通过递归过程产生 C' 的所有含有 $m-1$ 个子集的恰当划分,然后对每种恰当划分进行点和子集的相关处理;

(3) 当 $m\geqslant 2$ 时,对于 C' 的任意一种含有 m 个子集的恰当划分 P_3,只要将顶点 s_l 和 t_l 分别插入到 P_3 的任意两个不同子集中,就得到 C 的一种含有 m 个子集的恰当划分.因此,算法的第 3.4 步首先通过递归过程产生 C' 的所有含有 m 个子集的恰当划分,然后在每种恰当划分的不同子集中添加点 s_l 和 t_l.

显然,上述 3 种情况是互不相容的.根据加法原理,C 的所有含有 m 个子集的恰当划分等于这 3 种情况的并集.同时,由于 C' 的任何一种非恰当划分或者任何一种子集个数少于 $m-2$ 的恰当划分都不可能通过处理点对 (s_l,t_l) 得到 C 的一种含有 m 个子集的恰当划分,因此上述 3 种不同情况包含了 C 的所有含有 m 个子集的恰当划分.故算法 PARTITION 能够正确生成 $\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$ 的所有含有 m 个子集的恰当划分. □

至此,可以得到求解 Multicut 问题的一个完整的参数化算法,其具体描述如图 2 所示.其中,语句 6.2 中的 $NMC(G',U[j],k')$ 表示文献[9]中判定 Node Multicut 扩展问题实例 $(G',U[j],k')$ 是否存在一个大小不超过 k' 的分割集的算法,$U[j]$ 表示 $\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$ 的一种恰当划分.

Algorithm ***NPMC***$(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$
Input: an instance $(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$ of the parameterized Multicut problem ($l\geqslant 3$)
Output: a separator of size bounded by k for $(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$ or report "No".
1. $Q=\varnothing$; $w=\lceil\sqrt{2l}\rceil$;
2. **For** i=2 to w **Do** $Q=Q\cup$***PARTITION***$(\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\},i)$;
3. $z=|Q|$; //z denotes the number of partitions in Q
4. $U[1..z]=Q$; // set the partitions in Q to the array U
5. j=1; H="No";
6. **while** $j\leqslant z$ and H="No" **do**
 6.1 $\{G'=G$; $k'=k$;
 6.2 H=***NMC***$(G',U[j],k')$;
 6.3 $j=j+1\}$
7. **return** H.

Fig.2 An improved parameterized algorithm for the Multicut problem
图 2 Multicut 问题的一种改进的参数算法

定理 5. 算法 NPMC 可以在时间 $O^*\left(\lceil\sqrt{2l}\rceil^{2l}4^k\right)$ 内求解严格的 Multicut 问题.

证明:对于 Multicut 问题的任意一个实例 $(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$,设 $C=\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$,由推论 2 可知,$\bigcup_{m=2}^{h}B_m$ 包含了 C 的所有极大恰当划分,其中,$B_m=\{P|P$ 为 C 的一种含有 m 个子集的恰当划分$\}$,$h=\lceil\sqrt{2l}\rceil$.因此,算法 NPMC 的第 2 步通过算法 $PARTITION(C,m)$ 产生了集合 C 的所有子集个数不超过 $\lceil\sqrt{2l}\rceil$ 的恰当划分.由前面的分析可知,这些恰当划分决定了需要判定的 Node Multicut 扩展问题的实例总数.算法 NPMC 的第 6.1 步~第 6.3 步依次对所有需要判定的 Node Multicut 扩展问题实例进行判定.根据文献[9],算法 $NMC(G',U[j],k')$ 能够正确判定实例 $(G',U[j],k')$ 是否存在一个大小不超过 $k'(k'=k)$ 的分割集.又由定理 1 和定理 2 可知,如果 Multicut 问题实例 $(G,\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$ 存在一个大小不超过 k 的分割集,则在算法的第 6 步循环结束时一定可以找到.否则,算法返回相关报告.因此,算法 NPMC 是正确的.进一步地,由引理 1 可知,算法 NPMC 对于求解受限的和非受限的 Multicut 问题都是正确的.

接下来分析算法 NPMC 的时间复杂度.根据推论 2 可知,$\bigcup_{m=2}^{h}B_m$ 中含有恰当划分的总数有上界 $\lceil\sqrt{2l}\rceil^{2l}$,每种划分中的顶点个数为 $2l$,因此算法第 2 步产生所有恰当划分需要的时间为 $O\left(\lceil\sqrt{2l}\rceil^{2l}2l\right)$.算法第 6 步最多循环

$\lceil\sqrt{2l}\rceil^{2l}$次,根据引理 1 可知,每次循环的时间为$O(4^k kn^3)$,可得第 6 步循环语句的时间为$O\left(\lceil\sqrt{2l}\rceil^{2l}4^k kn^3\right)$.因此,算法 NPMC 的时间复杂度为$O\left(\lceil\sqrt{2l}\rceil^{2l}2l\right)+O\left(\lceil\sqrt{2l}\rceil^{2l}4^k kn^3\right)=O\left(\lceil\sqrt{2l}\rceil^{2l}4^k kn^3\right)=O^*\left(\lceil\sqrt{2l}\rceil^{2l}4^k\right)$. □

2 非严格 Multicut 问题参数算法的改进

前面讨论的 Multicut 问题要求给定的顶点对相互之间没有相同点,本节讨论顶点对之间存在相同点的情况.对于给定的 l 个顶点对,不失一般性,设点 s 在其中 $q(q\leqslant l)$个点对中重复出现,即顶点对有形式$(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)$.由于处理策略将涉及给定顶点对的变化,因此下面分受限和非受限两种情况加以讨论.

(1) 分割集不允许包含给定顶点对中的点.

这时采取直接添加一些点和边的办法,将给定有重复点的顶点对转化为相互之间没有重复点的顶点对.具体添加过程为:设点 s 在图 G 上的所有邻居点的集合为 B,在图上增加 $q-1$ 个不同的点 $s_1,s_2,\ldots,s_{q-1}$,并且对于每一个 $s_p,1\leqslant p\leqslant q-1$,在点 s_p 和 B 中的每一个点之间都添加一条边.在此基础上,点 s_1 与 t_1,s_2 与 $t_2,\ldots,s_{q-1}$ 与 t_{q-1} 组成新的顶点对,$s_q=s$ 唯一地与 t_q 组成点对.

如果设$(G,\{(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)\},k)$为 Multicut 问题的一个实例,$(G',\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$为增加点和边以后对应的 Multicut 问题实例,则它们之间存在下述联系:

引理 2. $(G,\{(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集的充分必要条件为$(G',\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集.

证明:首先证明其充分性.假设实例$(G',\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集 S,且 S 不含有$\{s_1,t_1,s_2,t_2,\ldots,s_l,t_l\}$中的点,则在图 $G'-S$ 上,s_1 与 t_1,s_2 与 $t_2,\ldots,s_l$ 与 t_l 之间均不存在路径.下面只要证明在图 $G-S$ 上,点 s 与点 $t_1,t_2,\ldots,t_{q-1}$ 之间都不存在路径.反设 s 与其中的一个点 $t_p(1\leqslant p\leqslant q-1)$之间存在一条路径 Y_1,则路径 Y_1 一定经过 s 的某个邻居节点 g_1.又由前述添加操作可知,在图 G'上,点 g_1 也是 $s_1,s_2,\ldots,s_{q-1}$ 的邻居节点,即点 g_1 必定与点 s_p 相连.于是在图 $G'-S$ 上,点 s_p 与 t_p 之间存在一条经过点 g_1 的路径,这与假设矛盾.因此在图 $G-S$ 上,点 s 与点 $t_1,t_2,\ldots,t_{q-1}$ 之间都不存在路径.又由于在图 G 上增加点和边的操作不涉及到顶点对$(s_q,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)$,因此在图 $G-S$ 上,s_q 与 t_q,s_{q+1} 与 $t_{q+1},\ldots,s_l$ 与 t_l 之间也不存在路径,故 S 也是实例$(G,\{(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)\},k)$的一个大小不超过 k 的分割集.

类似地,可以证明其必要性.假设实例$(G,\{(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)\},k)$存在一个大小不超过 k 的分割集 S,则在图 $G-S$ 上,s 与$\{t_1,t_2,\ldots,t_q\},s_{q+1}$ 与 t_{q+1},s_{q+2} 与 $t_{q+2},\ldots,s_l$ 与 t_l 之间不存在路径.接下来只要证明在图 $G'-S$ 上,s_1 与 t_1,s_2 与 $t_2,\ldots,s_{q-1}$ 与 t_{q-1} 之间不存在路径即可.反设其中一个顶点对 s_p 与 $t_p(1\leqslant p\leqslant q-1)$之间存在一条路径 Y_2,显然,路径 Y_2 必定经过 s_p 的一个邻居节点 g_2;又 s_p 的邻居节点也是点 s 的邻居节点,因此在图 $G-S$ 上,点 s 与点 t_p 之间存在一条经过点 g_2 的路径,这与假设相矛盾.故 S 也是实例$(G',\{(s_1,t_1),(s_2,t_2),\ldots,(s_l,t_l)\},k)$的一个大小不超过 k 的分割集. □

(2) 分割集允许含有给定顶点对中的点.

对于这种情形,可以首先比较 $l-q+1$ 和 k 的大小.当 $l-q+1\leqslant k$ 时,显然问题实例是一个真实例,因为集合$\{s,s_{q+1},\ldots,s_l\}$就是该实例的一个分割集.当 $l-q+1>k$ 时,可以首先通过添加一些点将 Multicut 问题不受限的实例转化为受限的实例.具体添加过程为:在图 G 上对点集$\{t_1,t_2,\ldots,t_q,s_q,s_{q+1},t_{q+1},\ldots,s_l,t_l\}$(其中,$s_q=s$)来说,对应于每一个点 $s_i(q\leqslant i\leqslant l)$添加一个点 s_i',并在 s_i' 和 s_i 之间添加一条边,对应于每一个点 $t_i(1\leqslant i\leqslant l)$添加一个点 t_i',并在 t_i' 和 t_i 之间添加一条边.求解不受限的实例$(G,\{(s,t_1),(s,t_2),\ldots,(s,t_q),(s_{q+1},t_{q+1}),\ldots,(s_l,t_l)\},k)$等价于求解受限的实例$(G',\{(s',t_1'),(s',t_2'),\ldots,(s',t_q'),(s_{q+1}',t_{q+1}'),\ldots,(s_l',t_l')\},k)$[1].接下来再运用前述情况(1)中的技术来处理顶点对中的重复点.

总之,对于非严格的 Multicut 问题,可以得到如下结论:

定理 6. 非严格的 Multicut 问题可以在时间 $O^*\left(\lceil\sqrt{2l}\rceil^{2l}4^k\right)$内求解.

3 结 论

本文首先深入分析了 Multicut 问题本身的结构特点,找到了 Multicut 问题和 Node Multicut 扩展问题之间的内在联系,然后运用集合划分策略和 Node Multicut 扩展问题的研究结果对 Multicut 问题提出了一个时间复杂度为 $O^*\left(\left\lceil\sqrt{2l}\right\rceil^{2l}4^k\right)$ 的固定参数可解算法.在运用集合划分策略时,其中的关键技术是将相关划分分为极大恰当划分和与之对应的隐含恰当划分两种类型,证明了问题的解可以在由极大恰当划分确定的实例中找到.这一技术显著地减少了需要判定的 Node Multicut 扩展问题实例的总数,明显地降低了算法的时间复杂度.

本文研究的 Multicut 问题是以给定顶点对的数目 l 和待删除的顶点数 k 两个值为参数,对于只以待删除的顶点数 k 为参数的 Multicut 问题是否为固定参数可解的目前还没有定论[10],我们将对此作进一步的研究.

References:

[1] Calinescu G, Fernandes CG, Reed B. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. Journal of Algorithms, 2003,48(2):333−359. [doi: 10.1016/S0196-6774(03)00073-7]

[2] Kortsarts Y, Kortsarz G, Nutov Z. Greedy approximation algorithms for directed multicuts. Networks, 2005,45(4):214−217. [doi: 10.1002/net.20066]

[3] Even G, Naor S, Schieber B, Rao S. Divide-and-Conquer approximation algorithms via spreading metrics. Journal of the ACM, 2000,47(4):585−616. [doi: 10.1145/347476.347478]

[4] Garg N, Vazirani V, Yannakakis M. Approximate max-flow min-(multi) cut theorems and their applications. SIAM Journal on Computing, 1996,25(2):235−251. [doi: 10.1137/S0097539793243016]

[5] Zhang P. Approximation algorithms for generalized Multicut in trees. Journal of Computer Research and Development, 2008,45(7): 1195−1202 (in Chinese with English abstract).

[6] Guo J, Niedermeier R. Fixed-Parameter tractability and data reduction for multicut in trees. Network, 2005,46(3):124−135. [doi: 10.1002/net.20081]

[7] Marx D. Parameterized graph separation problems. In: Downey R, Fellows M, Dehne F, eds. Proc. of the 1st Workshop on Parameterized and Exact Computation. LNCS 3162, Berlin: Springer-Verlag, 2004. 71−82.

[8] Guo J, Hüffner F, Kenar E, Niedermeier R, Uhlmann J. Complexity and exact algorithms for Multicut. In: Wiedermann J, Tel G, Pokorny J, Bielikova M, Stuller J, eds. Proc. of the SOFSEM 2006: Theory and Practice of Computer Science. LNCS 3831, Berlin: Springer-Verlag, 2006. 303−312.

[9] Chen J, Liu Y, Lu S. An improved parameterized algorithm for the minimum node multiway cut problem. In: Dehne F, Sack JR, Zeh N, eds. Proc. of the 10th Workshop on Algorithms and Data. LNCS 4619, Berlin: Springer-Verlag, 2007. 495−506.

[10] Demaine ED, Gutin G, Marx D, Stege U. Open problems from dagstuhl seminar 07281: Structure theory and FPT algorithmics for graphs, digraphs and hypergraphs. 2007. http://www.dagstuhl.de/about-dagstuhl/searchbox/

附中文参考文献:

[5] 张鹏.树上推广的 Multicut 问题的近似算法.计算机研究与发展,2008,45(7):1195−1202.

刘运龙(1971－),男,湖南衡阳人,博士,讲师,主要研究领域为参数计算,计算机算法及理论.

陈建二(1954－),男,博士,教授,博士生导师,主要研究领域为生物信息学,计算机理论,计算复杂性及优化,计算机网络优化算法,计算机图形理论与算法.

王建新(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机优化算法,生物信息学,网络优化理论.