Journal of Computer and System Sciences 78 (2012) 1637–1650

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Multicut in trees viewed through the eyes of vertex cover ☆

Jianer Chen [a,1], Jia-Hao Fan [a], Iyad Kanj [b,*,2], Yang Liu [c], Fenghui Zhang [d]

[a] Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA
[b] DePaul University, Chicago, IL 60604, USA
[c] Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA
[d] Google Kirkland, 747 6th Street South, Kirkland, WA 98033, USA

ARTICLE INFO

Article history:
Received 9 September 2011
Received in revised form 23 February 2012
Accepted 2 March 2012
Available online 6 March 2012

Keywords:
Parameterized complexity
Kernelization
Parameterized algorithms
Multicut

ABSTRACT

We take a new look at the multicut problem in trees, denoted MULTICUT ON TREES henceforth, through the eyes of the VERTEX COVER problem. This connection, together with other techniques that we develop, allows us to give an upper bound of $O(k^3)$ on the kernel size for MULTICUT ON TREES, significantly improving the $O(k^6)$ upper bound given by Bousquet et al. We exploit this connection further to present a parameterized algorithm for MULTICUT ON TREES that runs in time $O^*(\rho^k)$, where $\rho = (\sqrt{5}+1)/2 \approx 1.618$. This improves the previous (time) upper bound of $O^*(2^k)$, given by Guo and Niedermeier, for the problem.

1. Introduction

In the MULTICUT ON TREES problem we are given a tree T, a set of requests $R \subseteq V(T) \times V(T)$ between pairs of vertices in T, and a nonnegative integer k, and we are asked to decide if we can remove at most k edges from the tree to disconnect all the requests in R (i.e., every path in the tree that corresponds to a request in R contains at least one of the removed edges).

The MULTICUT ON TREES problem has applications in networking [5]. The problem is known to be NP-hard, and its optimization version can be approximated to within ratio 2 [9]. We consider the MULTICUT ON TREES problem from the parameterized complexity perspective. We mention that the parameterized complexity of several graph separation problems, including variants of the MULTICUT ON TREES problem, was studied with respect to different parameters by Marx in [11]. Guo and Niedermeier [10] showed that the MULTICUT ON TREES problem is fixed-parameter tractable by giving an $O^*(2^k)$ time algorithm for the problem. (The asymptotic notation $O^*(f(k))$ denotes time complexity of the form $f(k) \cdot p(n)$, where $p(n)$ is a polynomial in the input length n.) They also showed that MULTICUT ON TREES has an exponential-size kernel. Recently, Bousquet, Daligault, Thomassé, and Yeo, improved the upper bound on the kernel size for MULTICUT ON TREES to $O(k^6)$ [3].

In this paper we take a new look at MULTICUT ON TREES through the eyes of the VERTEX COVER problem. This connection allows us to give an upper bound of $O(k^3)$ on the kernel size for MULTICUT ON TREES, significantly improving the previous $O(k^6)$ upper bound given by Bousquet et al. [3]. We exploit this connection further to give a parameterized algorithm

☆ A preliminary version of this paper appears in the proceedings of WADS 2011, Lecture Notes in Comput. Sci., vol. 6844, 2011, pp. 219–230.

* Corresponding author.
E-mail addresses: chen@cs.tamu.edu (J. Chen), grantfan@cs.tamu.edu (J.-H. Fan), ikanj@cs.depaul.edu (I. Kanj), yliu@cs.panam.edu (Y. Liu), fhzhang@gmail.com (F. Zhang).

[1] This work was supported in part by the USA National Science Foundation under the grants CCF-0830455 and CCF-0917288.
[2] This work was supported in part by a DePaul University Competitive Research Grant.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2012.03.001

for MULTICUT ON TREES that runs in $O^*(\rho^k)$ time, where $\rho = (\sqrt{5}+1)/2 \approx 1.618$ (golden ratio) is the positive root of the polynomial $x^2 - x - 1$, thus improving the $O^*(2^k)$ time algorithm, given by Guo and Niedermeier [10].

To obtain the $O(k^3)$ upper bound on the kernel size, we introduce a novel approach that relies on a grouping of the vertices in the tree. This grouping allows us to derive tighter upper bounds on the number of vertices in the tree after applying some new reduction rules that we introduce in this paper. Some of the reduction rules we apply exploit a connection between VERTEX COVER and MULTICUT ON TREES that is observed in this paper. The novel approach can be summarized as follows. We first group the vertices in the tree into $O(k)$ groups. We then introduce an ordering that orders the leaves in a group with respect to every other group. This ordering allows us to introduce a set of reduction rules that limits the number of leaves in a group that have requests to the vertices in another group. At the core of this set of reduction rule is a rule that utilizes the crown kernelization algorithm for VERTEX COVER [1]. All the above allows us to upper bound the number of leaves in the reduced instance by $O(k^2)$, improving the $O(k^4)$ upper bound on the number of leaves obtained in [3]. Finally, we show that the size of the reduced instance is at most the number of leaves in the reduced instance multiplied by a linear factor of k, thus yielding an upper bound of $O(k^3)$ on the size of the kernel.

To obtain the $O^*(((\sqrt{5}+1)/2)^k)$ time algorithm, we first establish new structural connections between MULTICUT ON TREES and VERTEX COVER that allow us to simplify the instance of MULTICUT ON TREES. We then exploit the simplified structure of the resulting instance to present a simple search-tree algorithm for MULTICUT ON TREES that runs in time $O^*(((\sqrt{5}+1)/2)^k)$. We note that, even though some connection between MULTICUT ON TREES and VERTEX COVER was observed in [9,10], this connection was not developed or utilized in kernelization algorithms, nor in parameterized algorithms for MULTICUT ON TREES.

We mention that, very recently, the multicut problem on general graphs was shown to be fixed-parameter tractable independently by Bousquet, Daligault, and Thomassé [2], and by Marx and Razgon [12], thus answering an outstanding open problem in parameterized complexity theory.

2. Preliminaries

We assume familiarity with basic graph theory and parameterized complexity notation and terminology. For more information, we refer the reader to [6,8,13,14].

2.1. Graphs, forests, trees, and caterpillars

For a graph H we denote by $V(H)$ and $E(H)$ the set of vertices and edges of H, respectively; $n(H) = |V(H)|$ and $e(H) = |E(H)|$ are the number of vertices and edges in H. For a set of vertices $S \subseteq V(H)$, we denote by $H[S]$ the subgraph of H induced by the vertices in S. For a vertex $v \in H$, $H - v$ denotes $H[V(H) \setminus \{v\}]$, and for a subset of vertices $S \subseteq V(H)$, $H - S$ denotes $H[V(H) \setminus S]$. By *removing* a subgraph H' of H we mean removing $V(H')$ from H to obtain $H - V(H')$. Two vertices u and v in H are said to be *adjacent* or *neighbors* if $uv \in E(H)$. For two vertices $u, v \in V(H)$, we denote by $H - uv$ the graph $(V(H), E(H) \setminus \{uv\})$, and by $H + uv$ the *simple* graph $(V(H), E(H) \cup \{uv\})$. By *removing* an edge uv from H we mean setting $H = H - uv$. For a subset of edges $E' \subseteq E(H)$, we denote by $H - E'$ the graph $(V(H), E(H) \setminus E')$. For a vertex $v \in H$, $N(v)$ denotes the set of neighbors of v in H. The *degree* of a vertex v in H, denoted $deg_H(v)$, is $|N(v)|$. The *degree* of H, denoted $\Delta(H)$, is $\Delta(H) = \max\{deg_H(v)\colon\ v \in H\}$. The *length* of a path in a graph H is the number of edges in it. A *matching* in a graph is a set of edges such that no two edges in the set share an endpoint. A *vertex cover* for a graph H is a set of vertices such that each edge in H is incident to at least one vertex in this set. A vertex cover for H is *minimum* if its cardinality is minimum among all vertex covers of H; we denote by $\tau(H)$ the cardinality/size of a minimum vertex cover of H.

A *tree* is a connected acyclic graph. A *leaf* in a tree is a vertex of degree at most 1. A nonleaf vertex in a tree is called an *internal* vertex. The *internal degree* of a vertex v in a tree is the number of nonleaf vertices in $N(v)$. For two vertices u and v, the *distance* between u and v in T, denoted $dist_T(u, v)$, is the length of the unique path between u and v in T. A leaf x in a tree is said to be *attached* to vertex u if u is the unique neighbor of x in the tree. A *caterpillar* is a tree consisting of a path with leaves attached to the vertices on the path. A *forest* is a collection of disjoint trees.

Let T be a tree with root r. For a vertex $u \neq r$ in $V(T)$, we denote by $\pi(u)$ the parent of u in T. A *sibling* of u is a child $v \neq u$ of $\pi(u)$ (if exists), an *uncle* of u is a sibling of $\pi(u)$, and a *cousin* of u is a child of an uncle of u. A vertex v is a *nephew* of a vertex u if u is an uncle of v. For a vertex $u \in V(T)$, T_u denotes the subtree of T rooted at u. The *children* of a vertex u in $V(T)$, denoted *children*(u), are the vertices in $N(u)$ if $u = r$, and in $N(u) - \pi(u)$ if $u \neq r$. A vertex u is a *grandparent* of a vertex v if $\pi(v)$ is a child of u. A vertex v is a *grandchild* of a vertex u if u is a grandparent of v.

2.2. Parameterized complexity

A *parameterized problem* is a set of instances of the form (x, k), where $x \in \Sigma^*$ for a finite alphabet set Σ, and k is a non-negative integer called the *parameter*. A parameterized problem Q is *fixed parameter tractable*, or simply FPT, if there exists an algorithm that on input (x, k) decides if (x, k) is a yes-instance of Q in time $f(k)n^{O(1)}$, where f is a computable function independent of $n = |x|$. A parameterized problem Q is *kernelizable* if there exists a polynomial-time reduction that maps an instance (x, k) of Q to another instance (x', k') of Q such that: (1) $|x'| \leqslant g(k)$ for some computable function g,