Available online at www.sciencedirect.com
ScienceDirect
Information Processing Letters 106 (2008) 81–86
Information Processing Letters
www.elsevier.com/locate/ipl

Two fixed-parameter algorithms for Vertex Covering by Paths on Trees ☆

Jiong Guo *, Rolf Niedermeier, Johannes Uhlmann

Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany

Received 26 April 2007; received in revised form 12 October 2007; accepted 17 October 2007

Available online 23 October 2007

Communicated by F.Y.L. Chin

Abstract

VERTEX COVERING BY PATHS ON TREES with applications in machine translation is the task to cover all vertices of a tree $T = (V, E)$ by choosing a minimum-weight subset of given paths in the tree. The problem is NP-hard and has recently been solved by an exact algorithm running in $O(4^C \cdot |V|^2)$ time, where C denotes the maximum number of paths covering a tree vertex. We improve this running time to $O(2^C \cdot C \cdot |V|)$. On the route to this, we introduce the problem TREE-LIKE WEIGHTED HITTING SET which might be of independent interest. In addition, for the unweighted case of VERTEX COVERING BY PATHS ON TREES, we present an exact algorithm using a search tree of size $O(2^k \cdot k!)$, where k denotes the number of chosen covering paths. Finally, we briefly discuss the existence of a size-$O(k^2)$ problem kernel.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Graph algorithms; Combinatorial problems; Fixed-parameter tractability; Exact algorithms

1. Introduction

The NP-hard VERTEX COVERING BY PATHS ON TREES (VCPT) has been recently identified as an interesting algorithmic problem with a direct application in machine translation [4,5]:

Input: A tree $T = (V, E)$ and a set L of paths in T, where each path is associated with a positive number as its weight.

Task: Find a minimum-weight subset $L' \subseteq L$ such that every vertex of T is covered by at least one path from L'.

Note that a path in L can uniquely be specified by its two endpoints and hence can be represented by a vertex pair. VCPT is closely related to connectivity augmentation problems in graphs. For instance, the NP-hard TREE AUGMENTATION problem asks for a minimum-size set of additional edges whose insertion into a tree makes it 2-edge-connected [1]. It is not hard to see that the unit-weight case of VCPT generalizes TREE

☆ Supported by the Deutsche Forschungsgemeinschaft (DFG), Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4, and project PABI (parameterized algorithmics for bioinformatics), NI 369/7.

* Corresponding author.
E-mail addresses: guo@minet.uni-jena.de (J. Guo), niedermr@minet.uni-jena.de (R. Niedermeier), uhlmann@minet.uni-jena.de (J. Uhlmann).

0020-0190/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2007.10.007

AUGMENTATION [5]. Indeed, one may interpret this unweighted VCPT as a "vertex version" of TREE AUGMENTATION, that is, whereas unweighted VCPT seeks to *cover* all tree vertices by a minimum number of paths, TREE AUGMENTATION seeks to cover all tree edges by a minimum number of paths.

Lin et al. [5] proved the NP-hardness of VCPT, even for the unweighted case.[1] Moreover, they provided an exact exponential-time algorithm and showed that VCPT can be approximated to a factor 2 in polynomial time. Their exact algorithm has running time $O(4^C \cdot |V|^2)$, where C is the maximum number of paths covering a tree vertex. The algorithm is based on dynamic programming. Lin et al. emphasize that exact algorithms "are desired in the machine translation application" [5, p. 75] and that assuming small values for the parameter C "maps to the machine translation application" [5, p. 81]. Thus, Lin et al.'s exact algorithm can be seen as a practically important fixed-parameter tractability result with respect to the parameter C, inviting further studies concerning fixed-parameter algorithms [6] for VCPT.

This work makes two contributions concerning the fixed-parameter tractability of VCPT. First, we improve Lin et al.'s $O(4^C \cdot |V|^2)$ time bound to $O(2^C \cdot C \cdot |V|)$. In particular, for the practically relevant case of $C = O(1)$ this means linear instead of quadratic time. We achieve this result by reducing VCPT to the more general problem TREE-LIKE WEIGHTED HITTING SET, and by solving the latter with dynamic programming in the style of tree decomposition based dynamic programming. Our second contribution only applies to the unweighted case of VCPT. Instead of considering the parameter C, here we study the parameter k denoting the size of the solution L', that is, $k := |L'|$. We provide three polynomial-time data reduction rules and, based on these, develop a size-$O(2^k \cdot k!)$ search tree for unweighted VCPT. Finally, pointing to related work [3,9] for the TREE AUGMENTATION problem, we indicate the existence of a size-$O(k^2)$ problem kernel for unweighted VCPT.

2. Parameter "maximum number of paths covering a vertex"

Taking into account that VCPT is NP-hard, Lin et al. [5] introduced the parameter C denoting the upper bound on the number of paths covering a tree vertex. Here, we improve their $O(4^C \cdot |V|^2)$-time algorithm to an $O(2^C \cdot C \cdot |V|)$-time algorithm. Both algorithms are based on dynamic programming. To present our algorithm, we introduce a special case of the HITTING SET problem, namely the TREE-LIKE WEIGHTED HITTING SET (TWHS) problem, and show that it is a generalization of VCPT. TREE-LIKE WEIGHTED HITTING SET is defined as follows.

Input: A base set $S = \{s_1, s_2, \ldots, s_n\}$, a tree-like subset collection $\mathcal{K}$ of subsets of S, $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$, and a weight function $\omega : S \to \mathbb{Q}_{>0}$, where $\mathbb{Q}_{>0}$ is the set of positive rational numbers.

Task: Find a *hitting set* $S' \subseteq S$ of minimum weight such that $K_i \cap S' \neq \emptyset$ for all $1 \leqslant i \leqslant m$.

Tree-like means that the subsets in $\mathcal{K}$ can be organized in a so-called *subset-tree* $\mathcal{T}$ such that there exists a one-to-one mapping between the vertices of $\mathcal{T}$ and the subsets in a way that for every element $s \in S$ the vertices corresponding to the subsets containing s induce a subtree of $\mathcal{T}$. The subset-tree and the corresponding mapping can be computed in linear time for a given tree-like subset collection [8]. Thus, we may assume that we are given a subset-tree and the mapping for the tree-like subset collection under consideration.

Given a VCPT-instance (T, L), define $\text{vpaths}(v) := \{p \in L \mid v \text{ lies on } p\}$. Moreover, for an edge e let $\text{epaths}(e) := \{p \in L \mid e \text{ lies on } p\}$. Clearly, $C = \max_{v \in V} |\text{vpaths}(v)|$. More importantly, note that for the base set $S := L$ the subset collection $\mathcal{K} := \{\text{vpaths}(v) \mid v \in V\}$ is tree-like—the subset-tree $\mathcal{T}$ and the VCPT input instance tree T are isomorphic. VCPT forms the special case of TWHS where every element in S only maps to a path and not to a subtree of $\mathcal{T}$.

Theorem 1. TREE-LIKE WEIGHTED HITTING SET *can be solved in* $O(2^C \cdot C \cdot m)$ *time, where* $C := \max_{K \in \mathcal{K}} |K|$.

Proof. Let $\mathcal{T} = (I, F)$ be the subset-tree for the tree-like subset collection $\mathcal{K}$. We use dynamic programming to find an optimal solution for TWHS. To this end, we construct a coloring of the elements of the base set with colors 0 or 1 such that for every subset at least one of the elements that it contains is colored 1, and the sum of the weights of the elements that is colored 1 is minimized. Let $\mathcal{T}$ be rooted at an arbitrary vertex $r \in I$. We use the following notation. For a vertex $v \in I$ let $\mathcal{T}_v = (I_v, F_v)$ denote the subtree of $\mathcal{T}$ rooted at vertex v. If v is a child of u in the rooted tree, then let $\text{parent}(v) := u$. Define $K(v)$ to be the subset in $\mathcal{K}$ associated with $v \in I$ and define $\mathcal{K}_v$ to be the set of subsets corresponding to

[1] Note that the NP-hardness heavily relies on the fact that the path set L in the problem definition is part of the input.

Download English Version:

https://daneshyari.com/en/article/427982

Download Persian Version:

https://daneshyari.com/article/427982

Daneshyari.com