A Linear-Time Algorithm for the Terminal Path Cover Problem in Trees*

Ruo-Wei Hung† and Chih-An Fang

Department of Computer Science and Information Engineering,
Chaoyang University of Technology,
Wufong, Taichung 41349, Taiwan

September 21, 2007

Abstract

Let $G = (V, E)$ be a graph with vertex set V and edge set E and let $\mathcal{T}$ be a subset of V. A *terminal path cover* $\mathcal{PC}$ of G with respect to $\mathcal{T}$ is a set of pairwise vertex-disjoint paths of G which cover the vertices of G such that all vertices in $\mathcal{T}$ are end vertices of paths in $\mathcal{PC}$. The *terminal path cover problem* is to find a terminal path cover of G of minimum cardinality with respect to $\mathcal{T}$. The path cover problem is a special case of the terminal path cover problem with $\mathcal{T}$ be empty. In this paper, we show that the terminal path cover problem on trees can be solved in linear time.

Keywords: graph algorithms; path cover; terminal path cover; trees

1 Introduction

All graphs considered in this paper are finite and undirected, without loops or multiple edges. Let $G = (V, E)$ denote a graph with vertex set V and edge set E. Throughout this paper, let n and m denote the numbers of vertices and edges of graph G, respectively. The vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, respectively. A *path cover* of a graph G is a collection of vertex-disjoint paths $P_1 = (V_1, E_1), \cdots, P_k = (V_k, E_k)$ in G whose union is $V(G)$, where V_i and E_i, for $1 \leq i \leq k$, are the vertex and edge sets of path P_i, respectively, i.e., $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\bigcup_{1\leq i\leq k} V_i = V(G)$. The *path cover problem* is to find a path cover of a graph G of minimum cardinality, called the *path cover number* of G. It is evident that the path cover problem for general graphs is NP-complete since finding a path cover, consisting of a single path, corresponds directly to the Hamiltonian path problem [9]. The Hamiltonian path problem on some special classes of graphs, including bipartite graphs [15], split graphs [10], chordal bipartite graphs [19], strong chordal split graphs [19], undirected path graphs [2], and directed path graphs [20], has been shown to be NP-complete. Hence the path cover problem on these above classes of graphs and their superclasses of graphs is also NP-complete. However, it admits polynomial time algorithms when the input is restricted to be in some special classes of graphs, including trees [18], block graphs [25, 26], inter-

*This research was partly supported by the National Science Council of Taiwan, R.O.C. under grant no. NSC96-2221-E-324-024.

†Corresponding author's e-mail: rwhung@cyut.edu.tw

val graphs [1, 4], circular-arc graphs [12], cographs [6, 7, 14, 16], bipartite distance-hereditary graphs [28], distance-hereditary graphs [13], bipartite permutation graphs [23], and cocomparability graphs [8]. The path cover problem has many practical applications. For example, in order to establish ring protocol [24], a computer network may be augmented by some auxiliary edges so as to make it Hamiltonian [11, 17]. It is easily verified that the maximum number of additional edges needed to make a network Hamiltonian is identical to the path cover number of the network. Other notable applications of the path cover problem are VLSI designing, code optimization [3], mapping parallel programs to parallel architectures [18, 22], arranging a group of persons to dinner [5], and program testing [21].

In this paper, we would like to study the variant of the path cover problem, called the *terminal path cover problem*. It is deserved to be mentioned that we are the first researchers proposing the terminal path cover problem and then investigate its complexity on some graph classes. Let p be a simple path of a graph G and let $\mathcal{T}$ be a subset of $V(G)$. The first and last vertices visited by p are called the *path-start* and *path-end* of p, respectively. Both of them are *end vertices* of p. Note that p may contain only one vertex and in this case the path-start and path-end of p are the same vertex. A *terminal path cover* of a graph G with respect to $\mathcal{T}$ is a path cover of G such that all vertices in $\mathcal{T}$ are end vertices of paths in the path cover. A minimum terminal path cover of G with respect to $\mathcal{T}$ is a terminal path cover of G of minimum cardinality. Denote by $\pi(G, \mathcal{T})$ the cardinality of a minimum terminal path cover of G with respect to $\mathcal{T}$. Given a graph G and a subset $\mathcal{T}$ of $V(G)$, the *terminal path cover problem* is to find a terminal path cover of G of size $\pi(G, \mathcal{T})$. We call $\mathcal{T}$ the *terminal set* of G, the vertices in $\mathcal{T}$ the *terminals*, and the other vertices *free vertices*. The path cover problem is a special case of the terminal path cover problem with $\mathcal{T} = \emptyset$. By definition, the terminal path cover problem on graph class $\mathcal{C}$ is NP-complete when the path cover problem on $\mathcal{C}$ is NP-complete. But, it may not be absolutely true that the terminal path cover problem on graph class $\mathcal{C}$ with $\mathcal{T} \neq \emptyset$ is polynomial time solvable when the path cover problem on $\mathcal{C}$ is polynomially solvable.

The terminal path cover problem can be applied to the applications of the path cover problem in which some vertices corresponding to the applications are restricted to be terminals. For example, the terminal path cover problem can be applied to the terminal VLSI layout problem in which electronic components are represented as vertices and some vertices must be terminals, each terminal path cover corresponds to a terminal VLSI layout. Then, the problem is to find a minimum terminal VLSI layout corresponding to a minimum terminal path cover. It is also applicable in mapping parallel programs to parallel architectures as follows: Consider a mapping between program units and network of processors, where some program units must be executed before other program units. The capabilities of a given network of processors can be increased by adding some auxiliary links among the processors. The minimum set of edges needed to augment a line-like network so that it can accommodate a parallel program with some program units be run first is determined by a minimum terminal path cover of the graph representation of the program.

In this paper, we will solve the terminal path cover problem on trees in $O(n)$ time. Moran and Wolfstahl solved the path cover problem on trees in linear time [18]. Yan et al. solved the k-path cover problem, which is a variant of the path cover problem, on trees in linear time [27]. Our basic idea for solving the terminal path cover problem on trees is sketched as follows: Given a rooted tree T and a terminal set $\mathcal{T}$ in T. We traverse the nodes in T bottom-up. Thus, while visiting a node, its children were already visited. Let v be a node in T. Denote by T_v the subtree of T rooted at v and $\mathcal{T}_v = \mathcal{T} \cap T_v$. Initially, we set $\pi(T_v, \mathcal{T}_v) = 1$ for each leaf v of T. Suppose that it is about to process internal node v with $v_1, v_2, \cdots, v_\kappa$ being its children in T. Then, we compute $\pi(T_v, \mathcal{T}_v)$ by using $\pi(T_{v_i}, \mathcal{T}_{v_i})$'s, $1 \leq i \leq \kappa$, no matter whether v is a terminal. By traversing the nodes of T once, $\pi(T, \mathcal{T})$ is computed.

2 Preliminaries

Let p be a simple path in graph G. Denote the set of vertices visited by p by $V(p)$. Let v be a vertex in G and V' be a subset of $V(G)$. Denote $G-v$ by deleting v and edges incident to v from G and denote by $G - V'$ the graph obtained from G by deleting all vertices of V' and edges incident to any vertex of V'. For two sets X and Y, let $X - Y$ denote the set of elements of X that are not in Y. For two vertex-disjoint paths $p_1 = u_1 u_2 \cdots u_{|p_1|}$ and $p_2 = v_1 v_2 \cdots v_{|p_2|}$ of G such that the path-end of p_1 and the path-start of p_2 are adjacent, let $p_1 \to p_2$ denote the path $u_1 u_2 \cdots u_{|p_1|} v_1 v_2 \cdots v_{|p_2|}$ which is said to be the *concatenation* of p_1 and p_2.

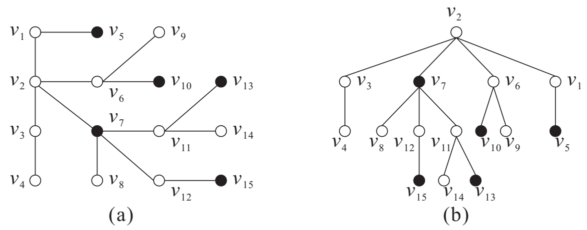

Fig. 1: (a) A tree, and (b) a rooted tree of (a) with root v_2, where the terminals are drawn by filled circles.

For a tree, we can arbitrarily pick a node in it as root and then obatin a rooted tree. For instance, Fig. 1(a) is a tree and Fig. 1(b) is its rooted tree with root v_2. In the rest of the paper, we assume that the input tree is a rooted tree. Let T be a rooted tree, v be a node in T, and let $\mathcal{T}$ be the terminal set of T. The subtree of T rooted at node v is denoted by T_v and let $\mathcal{T}_v = T_v \cap \mathcal{T}$.

Given a rooted tree T and a terminal set $\mathcal{T}$, we present an $O(n)$-time algorithm to solve the terminal path cover problem. Let v be a vertex of T. For simplicity, denote $\mathcal{T} - \{v\}$ by $\mathcal{T} - v$ and $\mathcal{T} \cup \{v\}$ by $\mathcal{T} + v$ in the rest of the paper. We establish some basic lemmas to be used in the next section.

Lemma 2.1. *Assume that G is a graph with terminal set $\mathcal{T}$ and $v \in \mathcal{T}$. Then, $\pi(G-v, \mathcal{T}-v)+1 \geq \pi(G, \mathcal{T}) \geq \pi(G - v, \mathcal{T} - v)$.*

Proof. Let $\mathcal{PC}^*$ be a minimum terminal path cover of $G - v$ with respect to $\mathcal{T} - v$. Since $\mathcal{PC}^*$ together with the path v forms a terminal path cover of G with respect to $\mathcal{T}$, $|\mathcal{PC}^*| + 1 \geq \pi(G, \mathcal{T})$

and, hence, $\pi(G-v,\mathcal{T}-v)+1 \geq \pi(G,\mathcal{T})$. On the other hand, suppose that $\mathcal{PC}$ is a minimum terminal path cover of G with respect to $\mathcal{T}$. Consider removing terminal v from $\mathcal{PC}$. What results is a set $\widetilde{\mathcal{PC}}$ of vertex-disjoint paths which is clearly a terminal path cover of $G-v$ with respect to $\mathcal{T}-v$. Since the deletion of a terminal in $\mathcal{PC}$ will decrease the number of paths by at most one and v is an end vertex of a path in $\mathcal{PC}$, we get that $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}|$ or $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}| - 1$. Since $\widetilde{\mathcal{PC}}$ is a terminal path cover of $G-v$ with respect to $\mathcal{T}-v$, $|\widetilde{\mathcal{PC}}| \geq \pi(G-v,\mathcal{T}-v)$. Thus, $|\mathcal{PC}| = \pi(G,\mathcal{T}) \geq |\widetilde{\mathcal{PC}}| \geq \pi(G-v,\mathcal{T}-v)$. Consequently, $\pi(G-v,\mathcal{T}-v)+1 \geq \pi(G,\mathcal{T}) \geq \pi(G-v,\mathcal{T}-v)$. □

Lemma 2.2. *Assume that G is a graph with terminal set $\mathcal{T}$ and v is a free vertex of G. Then, the following statements hold:*
(1) $\pi(G-v,\mathcal{T})+1 \geq \pi(G,\mathcal{T}+v)$;
(2) $\pi(G,\mathcal{T}+v) \geq \pi(G,\mathcal{T})$;
(3) $\pi(G-v,\mathcal{T})+1 \geq \pi(G,\mathcal{T}+v) \geq \pi(G-v,\mathcal{T})$;
(4) $\pi(G,\mathcal{T}) \geq \pi(G-v,\mathcal{T})-1$.

Proof. Since a minimum terminal path cover of $G-v$ with respect to $\mathcal{T}$ together with the path v forms a terminal path cover of G with respect to $\mathcal{T}+v$, $\pi(G-v,\mathcal{T})+1 \geq \pi(G,\mathcal{T}+v)$. Since a terminal path cover of G with respect to $\mathcal{T}+v$ is a terminal path cover of G with respect to $\mathcal{T}$, $\pi(G,\mathcal{T}+v) \geq \pi(G,\mathcal{T})$. On the other hand, suppose that $\mathcal{PC}$ is a minimum terminal path cover of G with respect to $\mathcal{T}+v$. Consider removing vertex v from $\mathcal{PC}$. What results is a terminal path cover $\widetilde{\mathcal{PC}}$ of $G-v$ with respect to $\mathcal{T}$. Since the deletion of a terminal in $\mathcal{PC}$ will decrease the number of paths by at most one and v is an end vertex of a path in $\mathcal{PC}$, we get that $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}|$ or $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}|-1$. Since $\widetilde{\mathcal{PC}}$ is a terminal path cover of $G-v$ with respect to $\mathcal{T}$, $|\widetilde{\mathcal{PC}}| \geq \pi(G-v,\mathcal{T})$. Thus, $|\mathcal{PC}| = \pi(G,\mathcal{T}+v) \geq |\widetilde{\mathcal{PC}}| \geq \pi(G-v,\mathcal{T})$. Combining with Statement (1), we get that $\pi(G-v,\mathcal{T})+1 \geq \pi(G,\mathcal{T}+v) \geq \pi(G-v,\mathcal{T})$.

Finally, we prove Statement (4). Let $\mathcal{PC}$ be a minimum terminal path cover of G with respect to $\mathcal{T}$. Consider removing vertex v from $\mathcal{PC}$. What results is a set $\mathcal{PC}^*$ of vertex-disjoint paths which is clearly a terminal path cover of $G-v$ with respect to $\mathcal{T}$. Since the removal of a free vertex from $\mathcal{PC}$ will increase the number of paths by at most one, we get that $|\mathcal{PC}|+1 \geq |\mathcal{PC}^*| \geq \pi(G-v,\mathcal{T})$. Thus, $\pi(G,\mathcal{T}) \geq \pi(G-v,\mathcal{T})-1$. □

3 The Terminal Path Cover Problem in Trees

In this section, let v be a vertex in rooted tree T, $v_1, v_2, \cdots, v_\kappa$ be the children of v in T, and let $\mathcal{T}_i$ denote the terminal set of T_{v_i} for $1 \leq i \leq \kappa$. By definition, T_{v_i}'s are pairwise disjoint for $1 \leq i \leq \kappa$. By Lemma 2.1, if v_i is a terminal then either $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i}-v_i,\mathcal{T}_i-v_i)+1$ or $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i}-v_i,\mathcal{T}_i-v_i)$. Let $v_i \to p_i$ be a path in a minimum terminal path cover of T_{v_i} with respect to $\mathcal{T}_i$ such that $v_i \notin \mathcal{T}_i$. If $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i}-v_i,\mathcal{T}_i-v_i)+1$, then $p_i = \emptyset$; otherwise, $p_i \neq \emptyset$. By Lemma 2.2, if v_i is a free vertex and $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i},\mathcal{T}_i+v_i)$ then either $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i}-v_i,\mathcal{T}_i)+1$ or $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i}-v_i,\mathcal{T}_i)$. Suppose that v_i is a free vertex. $\pi(T_{v_i},\mathcal{T}_i) = \pi(T_{v_i},\mathcal{T}_i+v_i)$ implies that $v_i \to p_i$ is a path in a minimum terminal path cover of

T_{v_i} with respect to $\mathcal{T}_i$, where p_i may be empty. If $\pi(T_{v_i}, \mathcal{T}_i) = \pi(T_{v_i} - v_i, \mathcal{T}_i) + 1$, then $p_i = \emptyset$; otherwise, $p_i \neq \emptyset$.

We next define the subsets I_t, J_t, I_f of $\{v_1, v_2, \cdots, v_\kappa\}$ as follows:

$I_t = \{i | 1 \leq i \leq \kappa, v_i \in \mathcal{T}_i, \pi(T_{v_i}, \mathcal{T}_i) = \pi(T_{v_i} - v_i, \mathcal{T}_i - v_i) + 1\}$,

$J_t = \{i | 1 \leq i \leq \kappa, v_i \in \mathcal{T}_i, \pi(T_{v_i}, \mathcal{T}_i) = \pi(T_{v_i} - v_i, \mathcal{T}_i - v_i)\}$,

$I_f = \{i | 1 \leq i \leq \kappa, v_i \notin \mathcal{T}_i, \pi(T_{v_i}, \mathcal{T}_i) = \pi(T_{v_i}, \mathcal{T}_i + v_i)\}$.

By Lemma 2.1, $I_t \cup J_t = \{i | 1 \leq i \leq \kappa, v_i \in \mathcal{T}_i\}$. By Statement (2) of Lemma 2.2, $\{i | 1 \leq i \leq \kappa, v_i \notin \mathcal{T}_i\} - I_f = \{i | 1 \leq i \leq \kappa, v_i \notin \mathcal{T}_i, \pi(T_{v_i}, \mathcal{T}_i + v_i) \geq \pi(T_{v_i}, \mathcal{T}_i) + 1\}$. That is, if v_i is a free vertex and $i \notin I_f$ then there exists no minimum terminal path cover of T_{v_i} such that it contains a path with end vertex v_i.

Lemma 3.1. *Assume that $\alpha \notin I_t \cup J_t \cup I_f$ for $1 \leq \alpha \leq \kappa$ and that $\mathcal{P}_\alpha$ is a terminal path cover of T_{v_α} with respect to $\mathcal{T}_\alpha$ satisfying that v_α is an end vertex of one path in $\mathcal{P}_\alpha$. Then, $|\mathcal{P}_\alpha| \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha) + 1$.*

Proof. By definition, v_α is a free vertex. Consider setting v_α to become a terminal from $\mathcal{P}_\alpha$. Since v_α is an end vertex of a path in $\mathcal{P}_\alpha$, we obtain a terminal path cover $\widetilde{\mathcal{P}}_\alpha$ of T_{v_α} with respect to $\mathcal{T}_\alpha + v_\alpha$ such that $|\widetilde{\mathcal{P}}_\alpha| = |\mathcal{P}_\alpha|$. Then, $|\widetilde{\mathcal{P}}_\alpha| \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha + v_\alpha)$. Since $\alpha \notin I_f$, $\pi(T_{v_\alpha}, \mathcal{T}_\alpha + v_\alpha) \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha) + 1$ by Statement (2) of Lemma 2.2. Thus, $|\mathcal{P}_\alpha| = |\widetilde{\mathcal{P}}_\alpha| \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha + v_\alpha) \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha) + 1$. □

Lemma 3.2. *Assume that v is a free vertex with children $v_1, v_2, \cdots, v_\kappa$ in rooted tree T and that $\mathcal{T}_i$ is the terminal set of T_{v_i} for $1 \leq i \leq \kappa$. Then,*

$$\pi(T_v, \mathcal{T}_v) = \begin{cases} \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) - 1 & , \textit{if } |I_t \cup I_f| \geq 2; \\ \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) & , \textit{if } |I_t \cup I_f| = 1; \\ \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1 & , \textit{otherwise.} \end{cases}$$

Proof. For each $1 \leq i \leq \kappa$, let $\mathcal{P}_i$ be a minimum terminal path cover of T_{v_i} with respect to $\mathcal{T}_i$. Consider the case of $|I_t \cup I_f| \geq 2$. Let α and β be in $I_t \cup I_f$ such that $v_\alpha \to p_\alpha \in \mathcal{P}_\alpha$ and $v_\beta \to p_\beta \in \mathcal{P}_\beta$, where p_α and p_β may be empty. Then, $\bigcup_{1 \leq i \leq \kappa} \mathcal{P}_i - \{v_\alpha \to p_\alpha, v_\beta \to p_\beta\} \cup \{p_\alpha \to v_\alpha \to v \to v_\beta \to p_\beta\}$ forms a terminal path cover of T_v of size $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) - 1$ with respect to $\mathcal{T}_v$. Hence, $\pi(T_v, \mathcal{T}_v) \leq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) - 1$. By Statement (4) of Lemma 2.2, $\pi(T_v, \mathcal{T}_v) \geq \pi(T_v - v, \mathcal{T}_v) - 1 = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) - 1$. Thus, $\pi(T_v, \mathcal{T}_v) = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) - 1$ if $|I_t \cup I_f| \geq 2$. In the following, assume that $|I_t \cup I_f| < 2$.

Let $\mathcal{PC}$ be a minimum terminal path cover of T_v with respect to $\mathcal{T}_v$. A path in $\mathcal{PC}$ is called *mixed* if it contains vertices in two different T_{v_i}'s. It is easy to see that there is at most one mixed path in $\mathcal{PC}$. If a mixed path p exists in $\mathcal{PC}$, let p be of the form $p_x \to v \to p_y$, where the path-end of p_x is v_x and the path-start of p_y is v_y for $1 \leq x, y \leq \kappa$. Consider the following two cases:

Case 1: $|I_t \cup I_f| = 1$. Let α be in $I_t \cup I_f$ such that $v_\alpha \to p_\alpha \in \mathcal{P}_\alpha$, where p_α may be empty. Then, $\bigcup_{1 \leq i \leq \kappa} \mathcal{P}_i - \{v_\alpha \to p_\alpha\} \cup \{v \to v_\alpha \to p_\alpha\}$ forms a terminal path cover of T_v with respect to $\mathcal{T}_v$. Hence, $\pi(T_v, \mathcal{T}_v) \leq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$. Next, we will prove that $\pi(T_v, \mathcal{T}_v) \geq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$.

First consider that there exists a mixed path $p = p_x \to v \to p_y$ in $\mathcal{PC}$. Since $|I_t \cup I_f| = 1$, there exists at least one of x, y such that it is not in $I_t \cup I_f$. Without loss of generality, assume that $x \notin I_t \cup I_f$. Then, $x \in J_t$ or ($v_x \notin \mathcal{T}_x$ and $x \notin I_f$). Let $\widehat{\mathcal{P}}_x$ and $\widehat{\mathcal{P}}_y$ be the restrictions of $\mathcal{PC}$ to $T_{v_x} - V(p_x)$ and $T_{v_y} - V(p_y)$, respectively. We then prove the following claim:

Claim (1): if $x \notin I_t \cup I_f$, then $|\widehat{\mathcal{P}}_x| \geq \pi(T_{v_x}, \mathcal{T}_x)$. Suppose $x \in J_t$. Then, $v_x \in \mathcal{T}_x$ and p_x only contains v_x. Clearly, $\widehat{\mathcal{P}}_x$ is a terminal path cover of $T_{v_x} - v_x$ with respect to $\mathcal{T}_x - v_x$. Thus, $|\widehat{\mathcal{P}}_x| \geq \pi(T_{v_x} - v_x, \mathcal{T}_x - v_x)$. By definition of J_t, $\pi(T_{v_x}, \mathcal{T}_x) = \pi(T_{v_x} - v_x, \mathcal{T}_x - v_x)$. Hence, $|\widehat{\mathcal{P}}_x| \geq \pi(T_{v_x}, \mathcal{T}_x)$. Suppose $v_x \notin \mathcal{T}_x$ and $x \notin I_f$. Then, $\widehat{\mathcal{P}}_x \cup \{p_x\}$ forms a terminal path cover of T_{v_x} with respect to $\mathcal{T}_x$ such that v_x is an end vertex of path p_x. By Lemma 3.1, $|\widehat{\mathcal{P}}_x| + 1 \geq \pi(T_{v_x}, \mathcal{T}_x) + 1$ and, hence, $|\widehat{\mathcal{P}}_x| \geq \pi(T_{v_x}, \mathcal{T}_x)$. In either case, $|\widehat{\mathcal{P}}_x| \geq \pi(T_{v_x}, \mathcal{T}_x)$. Thus, Claim (1) is proved.

Since $\widehat{\mathcal{P}}_y \cup \{p_y\}$ is a terminal path cover of T_{v_y} with respect to $\mathcal{T}_y$, $|\widehat{\mathcal{P}}_y| + 1 \geq \pi(T_{v_y}, \mathcal{T}_y)$. Then, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) = \sum_{i=1; i \neq x,y}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + (\pi(T_{v_x}, \mathcal{T}_x) + \pi(T_{v_y}, \mathcal{T}_y)) \leq |\mathcal{PC} - \{p\} - \widehat{\mathcal{P}}_x - \widehat{\mathcal{P}}_y| + (|\widehat{\mathcal{P}}_x| + |\widehat{\mathcal{P}}_y| + 1) = |\mathcal{PC}|$.

Next consider that there exists no mixed path in $\mathcal{PC}$. Let $p_v = v \to p_\alpha$ be a path in $\mathcal{PC}$, where p_α may be empty and the path-start of p_α is v_α if $p_\alpha \neq \emptyset$. If $p_\alpha = \emptyset$, then $|\mathcal{PC}| = \pi(T_v, \mathcal{T}_v) \geq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$ and it contradicts that $\pi(T_v, \mathcal{T}_v) \leq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$. Thus, $p_\alpha \neq \emptyset$. Let $\widehat{\mathcal{P}}_\alpha$ be the restriction of $\mathcal{PC}$ to $T_{v_\alpha} - V(p_\alpha)$. Since $\widehat{\mathcal{P}}_\alpha \cup \{p_\alpha\}$ forms a terminal path cover of T_{v_α} with respect to $\mathcal{T}_\alpha$, $|\widehat{\mathcal{P}}_\alpha| + 1 \geq \pi(T_{v_\alpha}, \mathcal{T}_\alpha)$. Thus, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) = \sum_{i=1; i \neq \alpha}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + \pi(T_{v_\alpha}, \mathcal{T}_\alpha) \leq |\mathcal{PC} - \{p_\alpha\} - \widehat{\mathcal{P}}_\alpha| + |\widehat{\mathcal{P}}_\alpha| + 1 = |\mathcal{PC}|$.

Case 2: $|I_t \cup I_f| = 0$. In this case, $\bigcup_{1 \leq i \leq \kappa} \mathcal{P}_i \cup \{v\}$ forms a terminal path cover of T_v of size $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$ with respect to $\mathcal{T}_v$. Hence, $\pi(T_v, \mathcal{T}_v) \leq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$. Next, we will prove that $\pi(T_v, \mathcal{T}_v) \geq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$.

First consider that there exists a mixed path $p = p_x \to v \to p_y$ in $\mathcal{PC}$. Let $\widehat{\mathcal{P}}_x$ and $\widehat{\mathcal{P}}_y$ be the restrictions of $\mathcal{PC}$ to $T_{v_x} - V(p_x)$ and $T_{v_y} - V(p_y)$, respectively. Since $|I_t \cup I_f| = 0$, neither x nor y is in $I_t \cup I_f$. By Claim (1) in Case 1, $\pi(T_{v_x}, \mathcal{T}_x) \leq |\widehat{\mathcal{P}}_x|$ and $\pi(T_{v_y}, \mathcal{T}_y) \leq |\widehat{\mathcal{P}}_y|$. Then, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) = \sum_{i=1; i \neq x,y}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + (\pi(T_{v_x}, \mathcal{T}_x) + \pi(T_{v_y}, \mathcal{T}_y)) \leq |\mathcal{PC} - \{p\} - \widehat{\mathcal{P}}_x - \widehat{\mathcal{P}}_y| + (|\widehat{\mathcal{P}}_x| + |\widehat{\mathcal{P}}_y|) = |\mathcal{PC}| - 1$. Thus, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1 \leq |\mathcal{PC}| = \pi(T_v, \mathcal{T}_v)$.

Next consider that there exists no mixed path in $\mathcal{PC}$. Let $p_v = v \to p_\alpha$ be a path in $\mathcal{PC}$, where p_α may be empty and the path-start of p_α is v_α if $p_\alpha \neq \emptyset$. If $p_\alpha = \emptyset$, then $|\mathcal{PC}| \geq \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$. Suppose $p_\alpha \neq \emptyset$ below. Let $\widehat{\mathcal{P}}_\alpha$ be a restriction of $\mathcal{PC}$ to $T_{v_\alpha} - V(p_\alpha)$. Since $|I_t \cup I_f| = 0$, $\alpha \notin I_t \cup I_f$. By Claim (1) in Case 1, $\pi(T_{v_\alpha}, \mathcal{T}_\alpha) \leq |\widehat{\mathcal{P}}_\alpha|$. Then, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) = \sum_{i=1; i \neq \alpha}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + \pi(T_{v_\alpha}, \mathcal{T}_\alpha) \leq |\mathcal{PC} - \{p_v\} - \widehat{\mathcal{P}}_\alpha| + |\widehat{\mathcal{P}}_\alpha| = |\mathcal{PC}| - 1$. Thus, $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1 \leq |\mathcal{PC}| = \pi(T_v, \mathcal{T}_v)$. □

Lemma 3.3. *Assume that v is a terminal with children $v_1, v_2, \cdots, v_\kappa$ in rooted tree T and that $\mathcal{T}_i$ is the terminal set of T_{v_i} for $1 \leq i \leq \kappa$. Then,*

$$\pi(T_v, \mathcal{T}_v) = \begin{cases} \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) & \text{, if } |I_t \cup I_f| \geq 1; \\ \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1 & \text{, otherwise.} \end{cases}$$

Proof. For each $1 \leq i \leq \kappa$, let $\mathcal{P}_i$ be a minimum terminal path cover of T_{v_i} with respect to $\mathcal{T}_i$. Consider that $|I_t \cup I_f| \geq 1$. Let α be in

$I_t \cup I_f$ such that $v_\alpha \to p_\alpha \in \mathcal{P}_\alpha$, where p_α may be empty. Then, $\bigcup_{1\le i\le\kappa} \mathcal{P}_i - \{v_\alpha \to p_\alpha\} \cup \{v \to v_\alpha \to p_\alpha\}$ forms a terminal path cover of T_v of size $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$ with respect to $\mathcal{T}_v$. Hence, $\pi(T_v, \mathcal{T}_v) \le \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$. By Lemma 2.1, $\pi(T_v, \mathcal{T}_v) \ge \pi(T_v - v, \mathcal{T}_v - v) = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$. Thus, $\pi(T_v, \mathcal{T}_v) = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$ if $|I_t \cup I_f| \ge 1$. In the following, assume that $|I_t \cup I_f| = 0$. Clearly, $\bigcup_{1\le i\le\kappa} \mathcal{P}_i \cup \{v\}$ forms a terminal path cover of T_v of size $\sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$ with respect to $\mathcal{T}_v$. Thus, $\pi(T_v, \mathcal{T}_v) \le \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$. We can easily prove that $\pi(T_v, \mathcal{T}_v) \ge \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$ via arguments similar to those for proving Case 2 of Lemma 3.2 under that there exists no mixed path in a minimum terminal path cover of T_v. Thus, $\pi(T_v, \mathcal{T}_v) = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1$ if $|I_t \cup I_f| = 0$. □

It follows from Lemma 3.3 that the following lemma can be easily verified:

Lemma 3.4. *Assume that v is a free vertex with children $v_1, v_2, \cdots, v_\kappa$ in rooted tree T and that $\mathcal{T}_i$ is the terminal set of T_{v_i} for $1 \le i \le \kappa$. Then,*

$$\pi(T_v, \mathcal{T}_v + v) = \begin{cases} \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) & \text{, if } |I_t \cup I_f| \ge 1; \\ \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i) + 1, & \text{otherwise.} \end{cases}$$

By definition of I_t and J_t, we should calculate $\pi(T_v, \mathcal{T}_v)$ and $\pi(T_v - v, \mathcal{T}_v - v)$ if v is a terminal. Since T_{v_i}'s are pairwise disjoint for $1 \le i \le \kappa$, the following lemma is obvious:

Lemma 3.5. *Assume that v is a terminal with children $v_1, v_2, \cdots, v_\kappa$ in rooted tree T and that $\mathcal{T}_i$ is the terminal set of T_{v_i} for $1 \le i \le \kappa$. Then, $\pi(T_v - v, \mathcal{T}_v - v) = \sum_{i=1}^{\kappa} \pi(T_{v_i}, \mathcal{T}_i)$.*

Based on Lemmas 3.2–3.5, given a rooted tree T with terminal set $\mathcal{T}$ we present an $O(n)$-time algorithm to compute $\pi(T, \mathcal{T})$ as follows: Initially, let $\pi(T_v, \emptyset) = 1$ and $\pi(T_v, \{v\}) = 1$ for each leaf $v \notin \mathcal{T}$; and let $\pi(T_v, \{v\}) = 1$ and $\pi(T_v - v, \emptyset) = 0$ for each leaf $v \in \mathcal{T}$. Our algorithm then traverses the nodes of T in a bottom-up manner. For each internal and free node v with terminal set $\mathcal{T}_v$, it computes $\pi(T_v, \mathcal{T}_v)$ and $\pi(T_v, \mathcal{T}_v + v)$ by using Lemma 3.2 and Lemma 3.4, respectively. For each internal and terminal node v with terminal set $\mathcal{T}_v$, it computes $\pi(T_v, \mathcal{T}_v)$ and $\pi(T_v - v, \mathcal{T}_v - v)$ by using Lemma 3.3 and Lemma 3.5, respectively. If the traversed node v is the root of T, then it outputs $\pi(T_v, \mathcal{T}_v)$. After visiting each node of T, $\pi(T, \mathcal{T})$ is calculated. For instance, given a rooted tree T shown in Fig. 1(b) and a terminal set $\mathcal{T} = \{v_5, v_7, v_{10}, v_{13}, v_{15}\}$, our algorithm outputs $\pi(T, \mathcal{T}) = 5$.

Since the process on each node takes constant time, our algorithm runs in $O(n)$ time. Though we only describe the algorithm to compute $\pi(T, \mathcal{T})$ for a rooted tree T with terminal set $\mathcal{T}$, it can be easily extended to find a minimum terminal path cover of T with respect to $\mathcal{T}$ in the same time bound. Hence, we conclude the following theorem.

Theorem 3.6. *Given a rooted tree T and a terminal set $\mathcal{T}$, the terminal path cover problem on T can be solved in $O(n)$ time.*

4 Concluding Remarks

The path cover problem on trees is linear solvable in [18]. However, the path cover problem is a special case of terminal path cover problem with terminal set be empty. In this paper, we solve the terminal path cover problem on trees in $O(n)$ time. It is interesting to know whether the approach used in this paper can be applied to design efficient algorithms for the terminal path cover problem on other classes of graphs, such as block graphs, Ptolemaic graphs and distance-hereditary graphs which form the super-classes of trees.

References

[1] S.R. Arikati and C.P. Rangan, Linear algorithm for optimal path cover problem on interval graphs, Inform. Process. Lett. 35 (1990) 149–153.

[2] A.A. Bertossi and M.A. Bonuccelli, Hamiltonian circuits in interval graph generalizations, Inform. Process. Lett. 23 (1986) 195–200.

[3] F.T. Boesch and J.F. Gimpel, Covering the points a digraph with point-disjoint paths and its application to code optimization, J. ACM 24 (1977) 192–198.

[4] M.S. Chang, S.L. Peng, and J.L. Liaw, Deferred-query: an efficient approach for some problems on interval graphs, Networks 34 (1999) 1–10.

[5] R.S. Chang, Jump number maximization for proper interval graphs and series-parallel graphs, Inform. Sci. 115 (1999) 103–122.

[6] G.J. Chang and D. Kuo, The $L(2,1)$-labeling problem on graphs, SIAM J. Discrete Math. 9 (1996) 309–316.

[7] D.G. Corneil, H. Lerchs, and L.K. Stewart, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.

[8] P. Damaschke, J.S. Deogun, D. Kratsch, and G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, Order 8 (1992) 383–391.

[9] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[10] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.

[11] S. Goodman and S. Hedetniemi, On the Hamiltonian completion problem, in: Proc. 1973 Capital Conf. on Graph Theory and Combinatorics, Lecture Notes in Math., Vol. 406, Springer, Berlin, 1974, pp. 262–272.

[12] R.W. Hung and M.S. Chang, Solving the path cover problem on circular-arc graphs by using an approximation algorithm, Discrete Appl. Math. 154 (2006) 76–105.

[13] R.W. Hung and M.S. Chang, Finding a minimum path cover of a distance-hereditary graph in polynomial time, to appear in: Discrete Appl. Math.

[14] H.A. Jung, On a class of posets and the corresponding comparability graphs, J. Combin. Theory Ser. B 24 (1978) 125–133.

[15] M.S. Krishnamoorthy, An NP-hard problem in bipartite graphs, SIGACT News 7 (1976) 26.

[16] R. Lin, S. Olariu, and G. Pruesse, An optimal path cover algorithm for cographs, Comput. Math. Appl. 30 (1995) 75–83.

[17] J. van Leeuwen, Graph Algorithms: Handbook of Theoretical Computer Science Vol. A, Elsevier, Amsterdam, 1990.

[18] S. Moran and Y. Wolfstahl, Optimal covering of cacti by vertex disjoint paths, Theoret. Comput. Sci. 84 (1991) 179–197.

[19] H. Müller, Hamiltonian circuits in chordal bipartite graphs, Discrete Math. 156 (1996) 291–298.

[20] G. Narasimhan, A note on the Hamiltonian circuit problem on directed path graphs, Inform. Process. Lett. 32 (1989) 167–170.

[21] S.C. Ntafos and S. Louis Hakimi, On path cover problems in digraphs and applications to program testing, IEEE Trans. Software Engrg. 5 (1979) 520–529.

[22] S. Pinter and Y. Wolfstahl, On mapping processes to processors, Internat. J. Parallel Programming 16 (1987) 1–15.

[23] R. Srikant, R. Sundaram, K.S. Singh, and C.P. Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, Theoret. Comput. Sci. 115 (1993) 351–357.

[24] A.S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, 1981.

[25] P.K. Wong, Optimal path cover problem on block graphs, Theoret. Comput. Sci. 225 (1999) 163–169.

[26] J.H. Yan and G.J. Chang, The path-partition problem in block graphs, Inform. Process. Lett. 52 (1994) 317–322.

[27] J.H. Yan, G.J. Chang, S.M. Hedetniemi, and S.T. Hedetniemi, k-path partitions in trees, Discrete Appl. Math. 78 (1997) 227–233.

[28] H.G. Yeh and G.J. Chang, Weighted connected domination and Steiner trees in distance-hereditary graphs, Discrete Appl. Math. 87 (1998) 245–253.