Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

A Linear-Time Algorithm for the Terminal Path Cover Problem in Block Graphs

Ruo-Wei Hung*†

Abstract—**In this paper, we study a variant of the path cover problem, namely, the terminal path cover problem. Given a graph G and a subset $\mathcal{T}$ of vertices of G, a terminal path cover of G with respect to $\mathcal{T}$ is a set of pairwise vertex-disjoint paths $\mathcal{PC}$ that covers the vertices of G such that the vertices of $\mathcal{T}$ are all endpoints of the paths in $\mathcal{PC}$. The terminal path cover problem is to find a terminal path cover of G of minimum cardinality; note that, if $\mathcal{T}$ is empty, the stated problem coincides with the classical path cover problem. We show that the terminal path cover problem can be solved in linear time on the class of block graphs. More precisely, we first establish a tree structural representation for the class of block graphs. Then, based on the tree structure, we present an algorithm which, for a block graph G on n vertices and m edges, computes a minimum terminal path cover of G in linear time, that is, in $O(n+m)$ time. The proposed algorithm is simple and only requires linear space.**

Keywords: graph algorithms, path cover, terminal path cover, block graphs, linear-time algorithms

1 Introduction

1.1 Framework–Motivation

A well studied problem with numerous practical applications in graph theory is to find a minimum number of vertex-disjoint paths of a graph G that covers the vertices of G. This problem, also known as the *path cover problem*, finds application in the fields of VLSI design, code optimization [4], mapping parallel programs to parallel architectures [14, 18], making a network have a Hamiltonian cycle [9, 21], and program testing [17]. 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, that is, the problem of deciding whether a graph has a Hamiltonian path [8].

The Hamiltonian path problem on some special classes of graphs, including bipartite graphs [13], chordal bipartite graphs [15], undirected path graphs [3], and directed path graphs [16], has been shown to be NP-complete. Hence, the path cover problem on these above classes of graphs is also NP-complete. However, the path cover problem admits polynomial time algorithms to solve when the input is restricted to be in some classes of graphs, including trees [14], block graphs [22, 23], interval graphs [2, 5], circular-arc graphs [11], distance-hereditary graphs [10], and cocomparability graphs [7].

Consider an application of the path cover problem that mapping a parallel program into a parallel architecture. The parallel program is divided into some units. The relations among program units can be represented as a graph, where program units are represented as vertices, and two vertices are adjacent if their represented units are relevant. Then, the program units are mapped into the processors of the parallel architecture. The capabilities of the parallel architecture can be increased by adding some auxiliary links among the processors. The minimum set of edges needed to augment the parallel architecture so that it can accommodate the parallel program is determined by a minimum path cover of the graph representing the parallel program. However, some program units of the parallel program may run first. Hence, some program units must be the endpoints of paths in a path cover.

Motivated by the above issue we state a variant of the path cover problem, namely, the *terminal path cover problem*, which generalizes the path cover problem. Let G be a graph and let $\mathcal{T}$ be a subset of vertices of G. A *terminal path cover* of G with respect to $\mathcal{T}$ is a path cover of G such that the vertices of $\mathcal{T}$ are all endpoints of the 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. The *terminal path cover problem* is to find a minimum terminal path cover of G. We denote the cardinality of a minimum terminal path cover of G with respect to $\mathcal{T}$ by $\pi(G, \mathcal{T})$, called the *minimum terminal path cover number*. We call $\mathcal{T}$ the *terminal set* of G, the vertices in $\mathcal{T}$ the *terminals*, and the other vertices *free vertices*. Note that the path cover problem is a special case of the terminal path cover problem with $\mathcal{T} = \emptyset$.

We show that the terminal path cover problem is linear solvable on the class of block graphs. We now introduce block graphs as follows. Let G be a connected graph. A

*Department of Computer Science and Information Engineering, Chaoyang University of Technology, No. 168, Jifong E. Rd., Wufong Township, Taichung Country 41349, Taiwan
†Email:rwhung@cyut.edu.tw

vertex v in G is called a *cut vertex* if the removal of v from G increases the number of connected components. A *block* is a maximal connected subgraph without a cut vertex. The intersection of two distinct blocks contains at most one vertex, and a vertex is a cut vertex if and only if it is the intersection of two or more blocks. Consequently, a graph with one or more cut vertices has at least two blocks. A connected graph is a *block graph* if every block in it is a clique (complete graph). Note that the line graph of a tree is a block graph, but the reverse is not true, i.e., the class of block graphs is the super-class of trees.

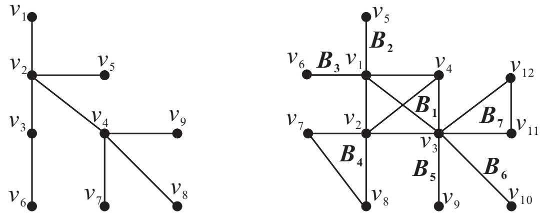

Figure 1: (a) A tree, and (b) a block graph.

1.2 Contribution and Related Works

In this paper, we study the the complexity status of the terminal path cover problem on the class of block graph, and show that this problem can be solved in linear time when the input is a block graph. More precisely, we establish a tree structure of a block graph. Based on the structure, we traverse its nodes bottom-up and then compute the related minimum terminal path cover numbers. After completing the traversal, the size of a minimum terminal path cover of a block graph G with n vertices and m edges is obtained. The proposed algorithm runs in time linear in the size of the input graph G, that is, in $O(n+m)$ time, and requires linear space. To the best of our knowledge, this is the first linear-time algorithm for solving the terminal path cover problem on the class of block graphs.

Previous related works are summarized below. Moran and Wolfstahl solved the path cover problem on trees [14]. Skikant et al. proposed a linear-time algorithm for the path cover problem on block graphs [20]. However, as pointed out by Yan and Chang [23], their algorithm does not work for all block graphs. Then, Yan and Chang gave another linear-time algorithm to solve the path cover problem on block graphs [23]. Although Wong [22] pointed out Yan and Chang's linear-time algorithm in [23] for the path cover problem on block graphs is not correct, Chang [6] corrected one typo at the algorithm in [23] and stated that the algorithm in [22] is much more complicated and is not clear to be implemented in linear time. Thus the linear-time algorithm proposed in [23] is correct and is more simple. Note that the path cover problem is a special case of the terminal path cover problem. In this paper, we will expand Yan and Chang's result to solve the terminal path cover problem on block graphs in linear time.

1.3 Road Map

The paper is organized as follows. In Section 2, we establish the notation and related terminology, and we present background results. In Section 3, we present our linear-time algorithm for the terminal path cover problem on block graphs. Finally, in Section 4 we conclude the paper and discuss possible future extensions.

2 Theoretical Framework

We consider finite undirected graphs without loops or multiple edges. Let $G=(V,E)$ be a graph with terminal set $\mathcal{T}$ and let p be a simple path in G. We denote the vertex set and edge set of G by $V(G)$ and $E(G)$, respectively. The set of vertices visited by p is denoted by $V(p)$. Let v be a vertex in G and V' be a subset of $V(G)$. We 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 simplicity, we denote $\mathcal{T}-\{v\}$ by $\mathcal{T}-v$, and denote $\mathcal{T}\cup\{v\}$ by $\mathcal{T}+v$. Let G' be a subgraph of G and let $\mathcal{PC}$ be a terminal path cover of G. We denote $\mathcal{T}\cap V(G')$ by $\mathcal{T}_{G'}$ and denote the restriction of $\mathcal{PC}$ to G' by $\mathcal{PC}_{G'}$. Then, $\mathcal{PC}_{G'}$ is a terminal path cover of G' with respect to $\mathcal{T}_{G'}$. For two vertex-disjoint paths $p_1=u_1u_2\cdots u_{|p_1|}$ and $p_2=v_1v_2\cdots v_{|p_2|}$ of G such that $u_{|p_1|}$ and v_1 are adjacent, let $p_1\rightarrow p_2$ denote the path $u_1u_2\cdots u_{|p_1|}v_1v_2\cdots v_{|p_2|}$ that is said to be the *concatenation* of p_1 and p_2.

2.1 The Block Tree

Let G be a block graph. Then, every block in G is a clique (complete graph) and every cut vertex is the intersection of two or more blocks in G. If B_i and B_j are two distinct blocks in G, then $B_i\cap B_j$ is empty or contains at most one vertex [1, 12, 19, 22]. For instance, Figure 1(b) depicts a block graph and v_1 is a cut vertex which is the intersection of blocks B_1, B_2, and B_3. On the other hand, Figure 1(a) reveals a tree. We can find that there are many similarities between them. In fact, trees are block graphs. It follows from the above observations that we can construct a tree-like hierarchy, called *block tree*, from a block graph as follows.

Definition 1. Let G be a block graph containing t blocks $B_1,B_2,\cdots,B_t$. The *representation tree* $T^*=(V^*,E^*)$ of G is constructed as follows: Create t new nodes $B_1,B_2,\cdots,B_t$ standing for these t blocks in G. Let $B_T=\{B_1,B_2,\cdots,B_t\}$ and let $V^*=V\cup B_T$. The edge set E^* of T^* is defined as $\{(v_i,B_j)|v_i\in B_j$ in G for $v_i\in V(G)$ and $t\geq j\geq 1\}$.

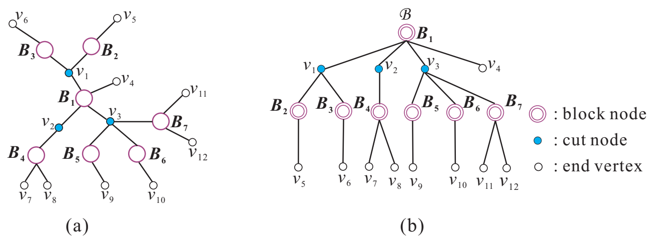

Figure 2: (a) A representation tree constructed from a block graph shown in Figure 1(b), and (b) a block tree $T_{\mathcal{B}}$ with root $\mathcal{B} = B_1$ for (a).

Clearly, T^* is a tree. For instance, given a block graph G shown in Figure 1(b), the representation tree T^* of G is shown in Figure 2(a). While picking an arbitrary block node $\mathcal{B}$ of T^* as the root, we get a rooted tree with root $\mathcal{B}$. This rooted tree, denoted by $T_{\mathcal{B}} = (V^*, E^*)$, is called the *block tree* corresponding to block graph G. Figure 2(b) depicts the block tree of the block graph shown in Figure 1(b). Note that rooting a representation tree suggests a natural way to decompose the computation. On the other hand, given a block graph G the block tree can be constructed in $O(|V(G)|+|E(G)|)$ time by the depth first search [1].

We call an element of V^* a node of block tree $T_{\mathcal{B}}$ in general. The element of V^* is called a *block node* of $T_{\mathcal{B}}$ if it is in B_T; that is, it is not a vertex of $V(G)$. A node is called an *end vertex* in $T_{\mathcal{B}}$ if it is in $V(G)$ but it is not a cut vertex in G. The remains are called *cut nodes*. Figure 2(b) also reveals the types of nodes in $T_{\mathcal{B}}$.

Let G be a block graph and let $T_{\mathcal{B}}$ be its corresponding block tree with root $\mathcal{B}$. The subtree of $T_{\mathcal{B}}$ rooted at node ω is denoted by T_ω, where ω is either a cut node or a block node in $T_{\mathcal{B}}$. Let G_ω denote the subgraph of G induced by the set of vertices of $V(G)$ which are nodes in the subtree T_ω of $T_{\mathcal{B}}$. For instance, $G_{B_4} = (\{v_7, v_8\}, \{(v_7, v_8)\})$ and $G_{v_1} = (\{v_1, v_5, v_6\}, \{(v_1, v_5), (v_1, v_6)\})$ in Figure 2.

2.2 The Background Results

In this subsection, we establish some basic lemmas that are used in proving our main results. Let G be a graph. Since removing a terminal from a terminal path cover of G will decrease the number of paths by at most one, the following lemma and corollary can be easily verified.

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

Corollary 2. *Assume that G is a graph with terminal set $\mathcal{T}$ and $v \in \mathcal{T}$. Then, $\pi(G, \mathcal{T}) > \pi(G - v, \mathcal{T} - v)$ if and only if $\pi(G, \mathcal{T}) = \pi(G - v, \mathcal{T} - v) + 1$.*

For a graph G with terminal set $\mathcal{T}$ and a free vertex v of G, we have the following lemma and corollary.

Lemma 3. *Assume that G is a graph with terminal set $\mathcal{T}$ and $v \in V(G) - \mathcal{T}$. 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) $\pi(G - v, \mathcal{T}) + 1 \geq \pi(G, \mathcal{T} + v) \geq \pi(G - v, \mathcal{T})$.

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 minimum 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 endpoint 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 obtain that $\pi(G - v, \mathcal{T}) + 1 \geq \pi(G, \mathcal{T} + v) \geq \pi(G - v, \mathcal{T})$. □

Corollary 4. *Assume that G is a graph with terminal set $\mathcal{T}$ and $v \in V(G) - \mathcal{T}$. Then, $\pi(G, \mathcal{T}) > \pi(G - v, \mathcal{T})$ if and only if $\pi(G, \mathcal{T}) = \pi(G, \mathcal{T} + v) = \pi(G - v, \mathcal{T}) + 1$.*

3 The Terminal Path Cover Problem in Block Graphs

We next present a linear-time algorithm to solve the terminal path cover problem on block graphs. In the rest of the paper, let G denote a block graph with terminal set $\mathcal{T}$ and let $T_{\mathcal{B}}$ represent its block tree with root $\mathcal{B}$. We will traverse the nodes of $T_{\mathcal{B}}$ in a bottom-up manner. Then, the traversed node may be either a block node or a cut node.

3.1 Block Nodes

Suppose that B is a block node with children $v_1, v_2, \cdots, v_c$ in $T_{\mathcal{B}}$. By definition, $v_1, v_2, \cdots, v_c$ form a clique and $(G_{v_i} - v_i)$'s are pairwise disjoint for $c \geq i \geq 1$. By Lemma 1, if $v_i \in \mathcal{T}$ then either $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i) + 1$ or $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i)$. By Lemma 3, if $v_i \notin \mathcal{T}$ and $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i)$ then either $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}}) + 1$ or $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}})$. Note that v_i is a free vertex and $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i)$ implies that v_i is an endpoint of a path in a minimum terminal path cover of G_{v_i} with respect to $\mathcal{T}_{G_{v_i}}$. We then define the following notation:

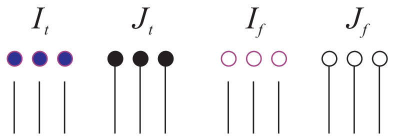

Figure 3: The sketch map of I_t, J_t, I_f, J_f, where terminals are drawn by filled circle, a line represents a path, and for $v_i \in I_t \cup I_f$, v_i is a path in a minimum terminal path cover of G_{v_i}.

Definition 2. Let B is a block node with children $v_1, v_2, \cdots, v_c$ in block tree $T_{\mathcal{B}}$. Define
$I_t = \{v_i \in \mathcal{T} | c \geq i \geq 1, \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i) + 1\}$,
$J_t = \{v_i \in \mathcal{T} | c \geq i \geq 1, \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i)\}$,
$I_f = \{v_i \notin \mathcal{T} | c \geq i \geq 1, \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}}) + 1\}$,
$J_f = \{v_i \notin \mathcal{T} | c \geq i \geq 1, \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i) = \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}})\}$.

By Lemma 1, $I_t \cup J_t = \{v_i | c \geq i \geq 1, v_i \in \mathcal{T}\}$. By Lemma 3, $I_f \cup J_f = \{v_i | c \geq i \geq 1, v_i \notin \mathcal{T}, \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) = \pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i)\}$. Figure 3 gives a brief illustration of I_t, J_t, I_f, J_f. We then have the following lemma for computing $\pi(G_B, \mathcal{T}_{G_B})$.

Lemma 5. *Assume that B is a block node with children $v_1, v_2, \cdots, v_c$ in block tree $T_{\mathcal{B}}$. Then,*

$$\pi(G_B, \mathcal{T}_{G_B}) = \begin{cases} \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| + 1 \\ \qquad , \text{ if } I_f \neq \emptyset \text{ and } I_t \cup J_f = \emptyset; \\ \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lfloor \frac{|I_t| + |J_f|}{2} \rfloor \\ \qquad , \text{ otherwise.} \end{cases}$$

Proof. For each $c \geq i \geq 1$, let $\mathcal{PC}_i$ be a minimum terminal path cover of G_{v_i} with respect to $\mathcal{T}_{G_{v_i}}$. For $v_i \in I_t$ (resp. $v_i \in J_t$, $v_i \in I_f$, $v_i \in J_f$), we may assume that v_i (resp. $v_i \to p_i$, v_i, $v_i \to p_i$) is a path in $\mathcal{PC}_i$ with $v_i \in \mathcal{T}_{G_{v_i}}$ (resp. $v_i \in \mathcal{T}_{G_{v_i}}$ and $p_i \neq \emptyset$, $v_i \notin \mathcal{T}_{G_{v_i}}$, $v_i \notin \mathcal{T}_{G_{v_i}}$ and $p_i \neq \emptyset$). For the case of $I_f \neq \emptyset$ and $I_t \cup J_f = \emptyset$, let $\mathcal{PC} = \cup_{c \geq i \geq 1} \mathcal{PC}_i - \cup_{v_i \in I_f} \{v_i\} \cup \{p\}$, where p is a path visiting all v_i's of I_f. Then, $\mathcal{PC}$ is a terminal path cover of G_B with respect to $\mathcal{T}_{G_B}$. Hence, $\pi(G_B, \mathcal{T}_{G_B}) \leq \sum_{i=1}^{c} \pi(G_i, \mathcal{T}_{G_{v_i}}) - |I_f| + 1$. For the other cases, let $\mathcal{P} = \cup_{v_i \in I_t} \{v_i\} \cup_{v_i \in J_f} \{v_i \to p_i\}$ and let $\mathcal{PC} = \cup_{c \geq i \geq 1} \mathcal{PC}_i - \cup_{v_i \in I_f} \{v_i\} - \mathcal{P} \cup \{p_1, p_2, \cdots, p_{\lceil \frac{|I_t| + |J_f|}{2} \rceil}\}$, where each p_i except p_1 is the concatenation of two paths in $\mathcal{P}$ and p_1 is formed by the remaining one or two paths in $\mathcal{P}$ together with all v_i's for $v_i \in I_f$. Then, $\mathcal{PC}$ is a terminal path cover of G_B with respect to $\mathcal{T}_{G_B}$. Hence, $\pi(G_B, \mathcal{T}_{G_B}) \leq \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lfloor \frac{|I_t| + |J_f|}{2} \rfloor$.

On the other hand, suppose that $\mathcal{PC}$ is a minimum terminal path cover of G_B with respect to $\mathcal{T}_{G_B}$. A path in $\mathcal{PC}$ is called *mixed* if it contains vertices in at least two different G_{v_i}'s. We may assume that $\mathcal{PC}$ is chosen to contain the fewest vertices in all mixed paths. Any mixed path r is of the form $r' \to r'' \to r'''$, where r' or r''' is either $\emptyset$, v_i in $\mathcal{T}_{G_{v_i}}$, or a nontrivial path in some G_{v_i} with v_i as an endpoint, and r'' is either $\emptyset$ or a sequence of some v_i's. It follows that the deletion of any vertex x in r'' is still a path, which we denote by $r - x$. Let
$I' = \{v_i | c \geq i \geq 1$, v_i is the only vertex of G_{v_i} that is in some mixed path r and $v_i \notin \mathcal{T}_{G_{v_i}}\}$, and
$J' = \{v_i | c \geq i \geq 1$, G_{v_i} contains a nonempty r' or r''' of a mixed path $r\}$.
It is easy to see that $I' \cap J' = \emptyset$. We claim that $I' \subseteq I_f$ and $J' \subseteq I_f \cup I_t \cup J_f$.

Next, we will prove the above two claims. Suppose that $v_i \in I' - I_f$. Let v_i be the only vertex of G_{v_i} that is in some mixed path r. By definition, $v_i \notin \mathcal{T}_{G_{v_i}}$. Consider the terminal path cover $\widetilde{\mathcal{PC}} = \mathcal{PC} - \mathcal{PC}_{G_{v_i} - v_i} - \{r\} \cup \mathcal{PC}_i \cup \{r - v_i\}$ of G_B. Then, we get that $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}| - |\mathcal{PC}_{G_{v_i} - v_i}| + |\mathcal{PC}_i| \leq |\mathcal{PC}| - \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}}) + \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) \leq |\mathcal{PC}|$ since $v_i \notin I_f$ and $v_i \notin \mathcal{T}_{G_{v_i}}$ implies $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) \leq \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}})$ by Corollary 4. Consequently, $\widetilde{\mathcal{PC}}$ is another minimum terminal path cover of G_B with fewer vertices in all mixed paths than $\mathcal{PC}$, a contradiction. Hence, $I' \subseteq I_f$.

Suppose that $v_i \in J' - (I_f \cup I_t \cup J_f)$. Without loss of generality, assume that G_{v_i} contains a nonempty r'_i for some mixed path $r_i = r'_i \to r''_i \to r'''_i$. Consider that r'_i contains $v_i \in \mathcal{T}_{G_{v_i}}$ only. Let $\widetilde{\mathcal{PC}} = \mathcal{PC} - \{r_i\} - \mathcal{PC}_{G_{v_i} - v_i} \cup \mathcal{PC}_i \cup \{r_i - v_i\}$. Then, $\widetilde{\mathcal{PC}}$ is a terminal path cover of G_B with respect to $\mathcal{T}_{G_B}$ and $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}| - |\mathcal{PC}_{G_{v_i} - v_i}| + |\mathcal{PC}_i| \leq |\mathcal{PC}| - \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i) + \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) \leq |\mathcal{PC}|$ since $v_i \notin I_t$ and $v_i \in \mathcal{T}_{G_{v_i}}$ implies that $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) \leq \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}} - v_i)$ by Corollary 2. Consequently, $\widetilde{\mathcal{PC}}$ is another minimum terminal path cover of G_B with fewer vertices than $\mathcal{PC}$ in all mixed paths, a contradiction. Now, consider that $r'_i = \widetilde{r}_i \to v_i$ such that $\widetilde{r}_i \neq \emptyset$ and $v_i \notin \mathcal{T}_{G_{v_i}}$. Consider the terminal path cover $\widetilde{\mathcal{PC}} = \mathcal{PC} - \{r_i\} - \mathcal{PC}_{G_{v_i} - V(r'_i)} \cup \mathcal{PC}_i \cup \{r''_i \to r'''_i\}$ of G_B, where $\mathcal{PC}_{G_{v_i} - V(r'_i)} \cup \{r'_i\}$ forms a terminal path cover of G_{v_i} with respect to $\mathcal{T}_{G_{v_i}} + v_i$. Then, $|\widetilde{\mathcal{PC}}| = |\mathcal{PC}| - |\mathcal{PC}_{G_{v_i} - V(r'_i)}| + |\mathcal{PC}_i| \leq |\mathcal{PC}| - (\pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i) - 1) + \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) \leq |\mathcal{PC}|$ since $v_i \notin I_f \cup J_f$ and $v_i \notin \mathcal{T}_{G_{v_i}}$ implies that $\pi(G_{v_i}, \mathcal{T}_{G_{v_i}} + v_i) > \pi(G_{v_i}, \mathcal{T}_{G_{v_i}})$ by Statement (2) of Lemma 3. Consequently, $\widetilde{\mathcal{PC}}$ is another minimum terminal path cover of G_B with fewer vertices than $\mathcal{PC}$ in all mixed paths, a contradiction. Thus, $J' \subseteq I_f \cup I_t \cup J_f$.

Now, suppose $\mathcal{PC}$ has κ mixed paths. Then,
$\pi(G_B, \mathcal{T}_{G_B}) = |\mathcal{PC}| = \sum_{v_i \notin I' \cup J'} |\mathcal{PC}_i| + \sum_{v_i \in I'} |\mathcal{PC}_{G_{v_i} - v_i}| +$

$\sum_{v_i \in J' \cap I_t} |\mathcal{PC}_{G_{v_i}-v_i}| + \sum_{v_i \in J' \cap (I_f \cup J_f)} |\mathcal{PC}_{G_{v_i}-V(r'_i)}| + \kappa \geq \sum_{v_i \notin I' \cup J'} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) + \sum_{v_i \in I'} \pi(G_{v_i} - v_i, \mathcal{T}_{G_{v_i}}) + \sum_{v_i \in J' \cap I_t} (\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - 1) + \sum_{v_i \in J' \cap (I_f \cup J_f)} (\pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - 1) + \kappa = \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I'| - |J'| + \kappa$.
For the case of $I_f \neq \emptyset$ and $I_t \cup J_f = \emptyset$, $I' \cup J' \subseteq I_f$. If $\kappa = 0$, then $I' = J' = \emptyset$, and, hence, $-|I'| - |J'| + \kappa = 0 \geq -|I_f| + 1$. If $\kappa \geq 1$, then $-|I'| - |J'| + \kappa \geq -|I_f| + 1$. Thus, $\pi(G_B, \mathcal{T}_{G_B}) \geq \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I'| - |J'| + \kappa \geq \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| + 1$. Consider the other cases. Since each mixed path contains vertices in at most two G_{v_i}'s with $v_i \in J'$, we get that $\kappa \geq \lceil \frac{|J'|}{2} \rceil$. Since $J' \subseteq I_f \cup I_t \cup J_f$, we have that $|J'| = |J' \cap I_f| + |J' \cap (I_t \cup J_f)|$, and, hence, $\lfloor \frac{|J'|}{2} \rfloor = \lfloor \frac{|J' \cap I_f| + |J' \cap (I_t \cup J_f)|}{2} \rfloor \leq |J' \cap I_f| + \lfloor \frac{|J' \cap (I_t \cup J_f)|}{2} \rfloor$. Then, $-|I'| - |J'| + \kappa \geq -|I'| - \lfloor \frac{|J'|}{2} \rfloor \geq -|I'| - |J' \cap I_f| - \lfloor \frac{|J' \cap (I_t \cup J_f)|}{2} \rfloor \geq -|I_f| - \lfloor \frac{|I_t| + |J_f|}{2} \rfloor$. Thus, $\pi(G_B, \mathcal{T}_{G_B}) \geq \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I'| - |J'| + \kappa \geq \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lfloor \frac{|I_t| + |J_f|}{2} \rfloor$. □

3.2 Cut Nodes

We then consider the cut nodes in block tree $T_{\mathcal{B}}$. Let v be a cut node with children block nodes $B_1, B_2, \cdots, B_b$ in $T_{\mathcal{B}}$. By Lemma 5 and Definition 2, we should calculate $\pi(G_v, \mathcal{T}_{G_v})$ and $\pi(G_v - v, \mathcal{T}_{G_v} - v)$ if $v \in \mathcal{T}$; and compute $\pi(G_v, \mathcal{T}_{G_v})$, $\pi(G_v, \mathcal{T}_{G_v} + v)$, $\pi(G_v - v, \mathcal{T}_{G_v})$ otherwise. For $v \in \mathcal{T}$, we compute $\pi(G_v, \mathcal{T}_{G_v})$ and $\pi(G_v - v, \mathcal{T}_{G_v} - v)$ by Lemma 10 and Lemma 6, respectively. For $v \notin \mathcal{T}$, we compute $\pi(G_v, \mathcal{T}_{G_v})$, $\pi(G_v, \mathcal{T}_{G_v} + v)$, and $\pi(G_v - v, \mathcal{T}_{G_v})$ by Lemma 11, Lemma 12, and Lemma 6, respectively. Since $G_{B_1}, G_{B_2}, \cdots, G_{B_b}$ are pairwise disjoint, the following lemma is obvious:

Lemma 6. *Assume that v is a cut node with children $B_1, B_2, \cdots, B_b$ in block tree $T_{\mathcal{B}}$. If $v \in \mathcal{T}$, then $\pi(G_v - v, \mathcal{T}_{G_v} - v) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}})$; otherwise, $\pi(G_v - v, \mathcal{T}_{G_v}) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}})$.*

Let B be a child of cut node v in $\mathcal{T}_{\mathcal{B}}$, and let $v_1, v_2, \cdots, v_c$ be the children of B in $\mathcal{T}_{\mathcal{B}}$. Let $G_B + v$ be the graph with vertex set $V(G_B) \cup \{v\}$ and edge set $E(G_B) \cup \{(v, v_i) | c \geq i \geq 1\}$. Recall that I_t, J_t, I_f, J_f are subsets of children of B that are defined in Definition 2. For graph $G_B + v$, we can construct its block tree T_B^v from T_B by setting v to be the child of B such that T_B^v is rooted at B, B has children $v, v_1, v_2, \cdots, v_c$, and v has no child. Then, the following two lemmas can be easily verified from Lemma 5.

Lemma 7. *Assume that v is a cut node, B is a child of v, and that $v_1, v_2, \cdots, v_c$ are children of B in $T_{\mathcal{B}}$. If $v \in \mathcal{T}$, then $\pi(G_B + v, \mathcal{T}_{G_B} + v) = \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lceil \frac{|I_t| + |J_f|}{2} \rceil + 1$.*

Lemma 8. *Assume that v is a cut node, B is a child of v, and that $v_1, v_2, \cdots, v_c$ are children of B in $T_{\mathcal{B}}$. If $v \notin \mathcal{T}$, then*

$$\pi(G_B + v, \mathcal{T}_{G_B}) = \begin{cases} \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| + 1 & \text{, if } I_t \cup J_f = \emptyset; \\ \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lfloor \frac{|I_t| + |J_f|}{2} \rfloor & \text{, otherwise.} \end{cases}$$

Assume that $v \notin \mathcal{T}$ is a cut node with child B in $T_{\mathcal{B}}$. By setting v to be a terminal, we can calculate $\pi(G_B + v, \mathcal{T}_{G_B} + v)$ by Lemma 7. Thus the following lemma immediately holds:

Lemma 9. *Assume that v is a cut node, B is a child of v, and that $v_1, v_2, \cdots, v_c$ are children of B in $T_{\mathcal{B}}$. If $v \notin \mathcal{T}$, then $\pi(G_B + v, \mathcal{T}_{G_B} + v) = \sum_{i=1}^{c} \pi(G_{v_i}, \mathcal{T}_{G_{v_i}}) - |I_f| - \lceil \frac{|I_t| + |J_f|}{2} \rceil + 1$.*

Let v be a cut node with child node B in $T_{\mathcal{B}}$. We observe that (1) if v is a terminal then it must be an endpoint of a path in a minimum terminal path cover of G_v with respect to $\mathcal{T}_{G_v}$, and (2) if v is a free vertex and $\pi(G_B + v, \mathcal{T}_{G_B}) = \pi(G_B + v, \mathcal{T}_{G_B} + v)$ then v is an endpoint of a path in a minimum terminal path cover of $G_B + v$ with respect to $\mathcal{T}_{G_B}$. Using Lemmas 7– 9, we have the following two lemmas. Due to the space limitation, the proofs of these two lemmas are omitted.

Lemma 10. *Assume that $v \in \mathcal{T}$ is a cut node with children $B_1, B_2, \cdots, B_b$ in block tree $T_{\mathcal{B}}$. If there exists B_λ for $b \geq \lambda \geq 1$ such that $\pi(G_{B_\lambda} + v, \mathcal{T}_{G_{B_\lambda}} + v) = \pi(G_{B_\lambda}, \mathcal{T}_{G_{B_\lambda}})$, then $\pi(G_v, \mathcal{T}_{G_v}) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{B_i})$; otherwise, $\pi(G_v, \mathcal{T}_{G_v}) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{B_i}) + 1$.*

Lemma 11. *Assume that $v \notin \mathcal{T}$ is a cut node with children $B_1, B_2, \cdots, B_b$ in block tree $T_{\mathcal{B}}$. Let $\tau = |\{B_i | b \geq i \geq 1, \pi(G_{B_i} + v, \mathcal{T}_{G_{B_i}}) = \pi(G_{B_i} + v, \mathcal{T}_{G_{B_i}} + v) = \pi(G_{B_i}, \mathcal{T}_{G_{B_i}})\}|$ and let $\eta = |\{B_i | b \geq i \geq 1, \pi(G_{B_i} + v, \mathcal{T}_{G_{B_i}}) = \pi(G_{B_i}, \mathcal{T}_{G_{B_i}})\}|$. Then,*

$$\pi(G_v, \mathcal{T}_{G_v}) = \begin{cases} \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}}) - 1 & \text{, if } \tau \geq 2; \\ \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}}) & \text{, if } \tau \leq 1 \text{ and } \eta \geq 1; \\ \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}}) + 1 & \text{, otherwise.} \end{cases}$$

The following lemma can be easily obtained from Lemma 10 by setting a free vertex v to become a terminal.

Lemma 12. *Assume that $v \notin \mathcal{T}$ is a cut node with children $B_1, B_2, \cdots, B_b$ in block tree $T_{\mathcal{B}}$. If there exists B_λ for $b \geq \lambda \geq 1$ such that $\pi(G_{B_\lambda} + v, \mathcal{T}_{G_{B_\lambda}} + v) = \pi(G_{B_\lambda}, \mathcal{T}_{G_{B_\lambda}})$, then $\pi(G_v, \mathcal{T}_{G_v} + v) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}})$; otherwise, $\pi(G_v, \mathcal{T}_{G_v} + v) = \sum_{i=1}^{b} \pi(G_{B_i}, \mathcal{T}_{G_{B_i}}) + 1$.*

3.3 The Algorithm

Based on Lemma 5, Lemmas 6–9, and Lemmas 10–12, given a block tree $T_{\mathcal{B}}$ with terminal set $\mathcal{T}$, a linear algorithm for computing $\pi(G_{\mathcal{B}}, \mathcal{T})$ is sketched as follows: Initially, let $\pi(G_v, \emptyset) = 1$, $\pi(G_v, \{v\}) = 1$, and $\pi(G_v - v, \emptyset) = 0$ for each end vertex $v \notin \mathcal{T}$; and let $\pi(G_v, \{v\}) = 1$ and $\pi(G_v - v, \emptyset) = 0$ for each end vertex $v \in \mathcal{T}$. Our algorithm then traverses the nodes of $T_{\mathcal{B}}$ in a bottom-up manner. For each block node B, it computes $\pi(G_B, \mathcal{T}_{G_B})$ using Lemma 5. If B is the root of $T_{\mathcal{B}}$, then it outputs $\pi(G_B, \mathcal{T}_{G_B})$; otherwise, let v be the parent of B, if $v \notin \mathcal{T}$ then it computes $\pi(G_B + v, \mathcal{T}_{G_B})$ and $\pi(G_B + v, \mathcal{T}_{G_B} + v)$ using Lemmas 8–9, otherwise it computes $\pi(G_B + v, \mathcal{T}_B + v)$ by Lemma 7. For each cut node v, the algorithm computes $\pi(G_v, \mathcal{T}_{G_v})$ using Lemmas 10–11, $\pi(G_v - v, \mathcal{T}_{G_v} - v)$ using Lemma 6, and $\pi(G_v, \mathcal{T}_{G_v} + v)$ using Lemma 12 if $v \notin \mathcal{T}$. After visiting each node of $T_{\mathcal{B}}$ once, $\pi(G_{\mathcal{B}}, \mathcal{T})$ is calculated.

The correctness of the above algorithm follows from Lemma 5, Lemmas 6–9, and Lemmas 10–12. Now, we analyze its complexity. Let B be a block node with c children and let v be a cut node with b children in $T_{\mathcal{B}}$. Then, processing block node B and cut node v takes $O(c)$ and $O(b)$ time, respectively. Let $B_1, B_2, \cdots, B_t$ be the block nodes in $T_{\mathcal{B}}$ and let $\delta(B_i)$ denote the number of children of B_i for $t \geq i \geq 1$. Then, $\sum_{i=1}^{t} \delta(B_i) = |V(G_{\mathcal{B}})|$. Thus, processing all block nodes requires $O(|V(G_{\mathcal{B}})|)$ time. Since the number of block nodes in $T_{\mathcal{B}}$ is bounded in $O(|V(G_{\mathcal{B}})|)$ and processing all cut nodes takes $O(t)$ time, processing all cut nodes requires $O(|V(G_{\mathcal{B}})|)$ time. It follows immediately from the above analyses that $\pi(G_{\mathcal{B}}, \mathcal{T})$ can be calculated in $O(|V(G_{\mathcal{B}})|)$ time.

Though we only describe the algorithm to compute $\pi(G_{\mathcal{B}}, \mathcal{T})$ for a block graph $G_{\mathcal{B}}$ with terminal set $\mathcal{T}$, it can be easily extended to find a minimum terminal path cover of $G_{\mathcal{B}}$ in the same time bound. Hence, we conclude the following theorem.

Theorem 13. *Given a block graph $G = (V, E)$ with terminal set $\mathcal{T}$, the terminal path cover problem on G can be solved in $O(|V| + |E|)$-linear time. Moreover, if its block tree is given, then the terminal path cover problem on G can be solved in $O(|V|)$ time.*

4 Concluding Remarks

The path cover problem on block graphs is linear solvable in [23]. However, the path cover problem is a special case of terminal path cover problem with terminal set be empty. In this paper, we first construct a block tree of a block graph. Based on the block tree, we solve the terminal path cover problem on block graphs in linear 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 the other classes of graphs, such as Ptolemaic graphs and distance-hereditary graphs that are the super-classes of block graphs.

5 Acknowledgements

This research was partly supported by the National Science Council of Taiwan under grant no. NSC96-2221-E324-024. The author would like to thank anonymous referee for many useful comments and suggestions which have improved the presentation of this paper.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading, MA, 1974.

[2] S.R. Arikati, C.P. Rangan, "Linear algorithm for optimal path cover problem on interval graphs," *Inform. Process. Lett.*, vol. 35, 1990, pp. 149–153.

[3] A.A. Bertossi, M.A. Bonuccelli, "Hamiltonian circuits in interval graph generalizations," *Inform. Process. Lett.*, vol. 23, 1986, pp. 195–200.

[4] F.T. Boesch, J.F. Gimpel, "Covering the points a digraph with point-disjoint paths and its application to code optimization," *J. ACM*, vol. 24, 1977, pp. 192–198.

[5] M.S. Chang, S.L. Peng, and J.L. Liaw, "Deferred-query: an efficient approach for some problems on interval graphs," *Networks*, vol. 34, 1999, pp. 1–10.

[6] G.J. Chang, "Corrigendum to: The path-partition problem in block graphs," *Inform. Process. Lett.*, vol. 83, 2002, pp. 293–293.

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

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

[9] S. Goodman, S. Hedetniemi, "On the Hamiltonian completion problem," in: *Proc. 1973 Capital Conf.*

on Graph Theory and Combinatorics, *Lecture Notes in Math.*, Springer, Berlin, vol. 406, 1974, pp. 262–272.

[10] R.W. Hung, M.S. Chang, "Finding a minimum path cover of a distance-hereditary graph in polynomial time," *Discrete Appl. Math.*, vol. 155, 2007, pp. 2242–2256.

[11] R.W. Hung, M.S. Chang, "Solving the path cover problem on circular-arc graphs by using an approximation algorithm," *Discrete Appl. Math.*, vol. 154, 2006, pp. 76–105.

[12] L.T. Quoc Hung, M.M. Sysło, M.L. Weaver, and D.B. West, "Bandwidth and density for block graphs," *Discrete Math.*, vol. 189, 1998, pp. 163–176.

[13] M.S. Krishnamoorthy, "An NP-hard problem in bipartite graphs," *SIGACT News*, vol. 7, 1976, pp. 26–26.

[14] S. Moran, Y. Wolfstahl, "Optimal covering of cacti by vertex disjoint paths," *Theoret. Comput. Sci.*, vol. 84, 1991, pp. 179–197.

[15] H. Müller, "Hamiltonian circuits in chordal bipartite graphs," *Discrete Math.*, vol. 156, 1996, pp. 291–298.

[16] G. Narasimhan, "A Note on the Hamiltonian circuit problem on directed path graphs," *Inform. Process. Lett.*, vol. 32, 1989 pp. 167–170.

[17] S.C. Ntafos, S. Louis Hakimi, "On path cover problems in digraphs and applications to program testing," *IEEE Trans. Software Engrg.*, vol. 5, 1979, pp. 520–529.

[18] S. Pinter, Y. Wolfstahl, "On mapping processes to processors," *Internat. J. Parallel Programming*, vol. 16, 1987, pp. 1–15.

[19] J.W. Robert E., "Complexity and block graphs," *Congr. Numer.*, vol. 33, 1981, pp. 129–142.

[20] 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.*, vol. 115, 1993, pp. 351–357.

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

[22] P.K. Wong, "Optimal path cover problem on block graphs," *Theoret. Comput. Sci.*, vol. 225, 1999, pp. 163–169.

[23] J.H. Yan,G.J. Chang, , "The path-partition problem in block graphs," *Inform. Process. Lett.*, vol. 52, 1994, pp. 317–322.

ISBN: 978-988-98671-8-8