Algorithms and Bounds for Path Covers of Tree-Structured Graphs

Madhura Dutta[1,2(✉)], Florent Foucaud[3], and Subhas C. Nandy[4]

[1] TCG CREST, Kolkata 700091, India
[2] Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India
madhura1998kkg@gmail.com
[3] Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Étienne, LIMOS, 63000 Clermont-Ferrand, France
[4] Dhirubhai Ambani University, Gandhinagar 382007, India

Abstract. The path cover problem is a natural graph covering problem that generalizes the Hamiltonian path problem. Given an input graph, it requires to find a minimum-size set of (not necessarily disjoint) paths that cover all the vertices of the graph. In this paper, we give two lower bounds for the path cover number in general graphs and show that they are tight for some graph classes. The second bound uses block-cut trees, a structure that highlights the tree-likeness of the graph. For outerplanar graphs, we use block-cut trees to design a linear-time algorithm for the path cover problem. There already exists a polynomial-time algorithm based on dynamic programming over a tree-decomposition [Foucaud et al., CALDAM 2025] that provides a polynomial-time algorithm for outerplanar graphs, but the exponent is huge.

Keywords: Path cover · block-cut tree · outerplanar graph

1 Introduction

In the context of graph algorithms, the aim of covering problems is to fully or partially cover the vertices or edges of a given graph. In the case of path covering problems, the graph is required to be covered via paths. (We will use the term *path* to denote simple paths only.) For a given simple undirected graph $G = (V, E)$, in the *path cover problem*, it is asked to find a minimum sized set of paths such that each vertex of the graph lies on at least one path in this set. The cardinality of a minimum path cover is called the *path cover number* of the graph. The case when the paths are required to be vertex-disjoint is called the *path partition problem*. Though this version is also termed as path cover

M. Dutta—Supported by DST INSPIRE PhD Fellowship.
F. Foucaud—This author was supported by the French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20–25), the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20–25, by the ANR project GRALMECO (ANR-21-CE48-0004) and by the CNRS IRL ReLaX.

Y. Li et al. (Eds.): COCOON 2026, LNCS 16835, pp. 231–244, 2027.
https://doi.org/10.1007/978-981-92-3309-0_18

in some papers, we reserve the name path partition for vertex-disjoint path cover throughout our work. Here, by the term path cover we will always mean a collection of paths (which need not be vertex-disjoint) that cover all the vertices of the graph (adopting the notation of [17]).

Definition 1. *Given a graph $G = (V, E)$, a* path cover *is a set $\mathcal{P}$ of paths in G (not necessarily vertex disjoint) such that for every vertex $v \in V$ there exists a path $P \in \mathcal{P}$ containing v. The cardinality of a minimum-size path cover is called the* path cover number *of the graph and is denoted by $\pi_c(G)$.*

We give lower bounds for the path cover number for arbitrary graphs and identify some graph classes that reach these bounds. The class of outerplanar graphs is a well-studied tree-like graph class, where the path cover number can be strictly larger than our given bounds. Our aim is to study the computational complexity of the path cover problem in this class from an algorithmic perspective. We use the concept of *block-cut tree* to provide one lower bound and to design an algorithm.

Related Work. Although the path partition problem has been extensively studied in the literature, the path cover problem is less explored. There are numerous applications of the path cover and path partition problems in software testing, bio-informatics, code optimization, database design, VLSI design, and parallel architecture [1,4,18]. The decision version of the path cover problem is NP-complete by reduction from the Hamiltonian path problem. Hence, the path cover problem is NP-hard for all the graph classes for which the Hamiltonian path problem is so, for example planar graphs, chordal graphs, bipartite graphs, chordal bipartite graphs, strongly chordal graphs, etc. [17]. There exist polynomial-time algorithms for the path cover problem for trees and graphs with bounded treewidth [12], block graphs [10] and cographs [16]. For graphs of bounded treewidth at most t, the algorithm from [12] runs in time $n^{(10t)!}$ (which places it in the XP complexity class when parameterized by t) and it is suspected in [12] that perhaps no algorithm running in time $f(t)n^{O(1)}$ (that is, in FPT time for parameter t) exists, but this remains open.

The path cover problem for directed acyclic graphs has been explored in [4]. The version in which the edges of the graph are to be covered has also been explored, where upper bounds on the minimum number of paths have been studied [7]. For covering all the edges of a graph with a minimum number of paths (not necessarily edge-disjoint), some bounds have been given in [13] using the cycle rank of the graph. Other variants of path cover, such as induced path cover [8], isometric path cover [9], identifying path cover [11], etc., have been studied, see the survey [17]. The path partition problem is well-studied in the literature [15,17,19,20,23], including for outerplanar graphs [21].

Our Results. In this paper, we first give a lower bound for the path cover number for arbitrary graphs and characterize some graphs for which this lower bound is tight. Although several bounds on path partition were studied, the path cover problem has not been explored much in this regard. In [10], some lower bounds

on path cover number were studied for some specific graph classes, using the notion of scattering number of a graph. We give another lower bound for the path cover number for general graphs using the block-cut tree structure. Similar ideas have been used in the PhD thesis [10] to provide an algorithm for solving the path cover problem in block graphs.

In [12], polynomial-time algorithms of the path cover problem for trees and bounded treewidth graphs were studied. Since outerplanar graphs have treewidth at most 2, the results of [12] provide a polynomial-time algorithm to solve the path cover problem for this class. But the time complexity of that algorithm is very high: $n^{t^{O(t)}}$, where t is the treewidth. We give a linear-time algorithm for the path cover problem on outerplanar graphs using the properties of block-cut trees and biconnected outerplanar graphs.

Preliminaries. Let $G = (V, E)$ be a simple undirected connected graph.

A *cut-vertex* of G is a vertex v of G such that deleting it (and its incident edges) will result in at least two non-empty disjoint connected components. These obtained connected components are known as *cut components* for v.

Assume $U = \{u_1, u_2, \ldots, u_k\} \subset V$ (where $k < n = |V|$) is the set of all cut-vertices of G; hence $deg(u_i) \geq 2$ for all $i \in \{1, 2, \ldots, k\}$, where $deg(u_i)$ denotes the degree of the vertex u_i (i.e., the number of vertices adjacent to u_i in G). We use $\rho(u_i)$ to denote the number of cut components for u_i.

Block-cut Tree of a Graph. A *block*, also known as a *biconnected component*, of a graph G is a maximal biconnected subgraph of G. Let $G = (V, E)$ be a connected graph. Observe that two blocks of G can share at most one common vertex, which is a cut-vertex. Using this observation, we construct a tree $T_G = (V_T, E_T)$, called the *block-cut tree* of the graph G, as follows (See Fig. 1):

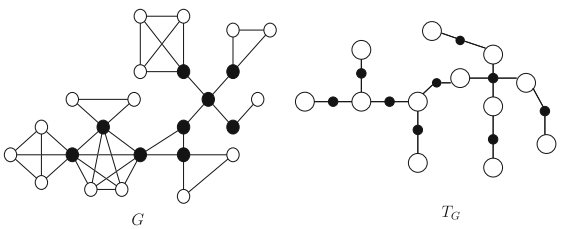

Fig. 1. A graph G and its block-cut tree T_G

(i) For every block B_i of G, create a vertex $b_i \in V_T$. Note that a cut edge (i.e., an edge not contained in any cycle) of G will also be considered by us as a block.
(ii) For every cut-vertex u_i of G, we create a vertex $\tilde{u}_i \in V_T$.
(iii) Whenever two blocks B_i and B_j share a common cut-vertex $u_l \in V$, we put two edges $(b_i, \tilde{u_l}), (b_j, \tilde{u_l}) \in E_T$.

Block-cut trees have been well-studied in the literature [3,5,14].

Lemma 2 ([14]). *A block-cut tree of a connected graph can be constructed in linear time.*

2 Lower Bounds for General Graphs

Let G be a connected graph, U and $\rho(u_i)$ are as defined earlier. Observe that at least one path P (say) is required to cover the vertices of G. Hence, $\pi_c(G) \geq 1$. Now, each vertex of U may be covered by this path. But then for a fixed u_i in this path P, the vertex preceding u_i and post u_i will lie in at most two different components among the $\rho(u_i)$ components corresponding to u_i. So, the path P can cover the vertices of at most two components among the $\rho(u_i)$ components corresponding to each u_i. So, there are still at least $\sum_{i=1}^{k}\{\rho(u_i) - 2\}$ vertices which are not covered by P. This indicates the following result:

Lemma 3. *Let P' be a path of G. Then there are at least $\sum_{i=1}^{k}\{\rho(u_i) - 2\}$ vertices of G that are not covered by P'.*

Proof. For a cut-vertex $u_i \in U$ and a path P' in G, P' intersects at most two components corresponding to u_i. Since the only cut-vertices of G are $u_1, u_2, \ldots, u_k$, the result follows. □

The above lemma gives a lower bound on the path cover number as follows.

Theorem 4. *For any graph G, we must have $\pi_c(G) \geq 1 + \left\lceil \frac{\sum_{i=1}^{k}\{\rho(u_i)-2\}}{2} \right\rceil$.*

Proof. Let $P_1, P_2, \ldots, P_\beta$ be a path cover of G. By Lemma 3, at least $\sum_{i=1}^{k}\{\rho(u_i) - 2\}$ vertices, which all belong to different blocks of G, are not covered by P_1. Note that any other path P_i, $i > 1$, may cover vertices from at most two distinct blocks among these $\sum_{i=1}^{k}\{\rho(u_i) - 2\}$ blocks in G. Hence, in order to cover the vertices uncovered by P_1, we need at least $\left\lceil \frac{\sum_{i=1}^{k}\{\rho(u_i)-2\}}{2} \right\rceil \leq \beta - 1$ additional paths, and the result follows. □

2.1 Graphs Attaining the Lower Bound

Now, we study some graph classes for which the path cover number is equal to the lower bound given in Theorem 4. We know that a graph G has a Hamiltonian path if and only if $\pi_c(G) = 1$. So, from Theorem 4, we have the following.

Proposition 5. *Any graph with a Hamiltonian path reaches the bound.*

Next, consider the class of trees. From [12], we already know the following result.

Theorem 6 ([12]). *An optimal path cover of a tree with ℓ leaves can be found in linear time and it has size $\left\lceil \frac{\ell}{2} \right\rceil$.*

The proof of the fact that the size of a minimum path cover of a tree is $\lceil \frac{\ell}{2} \rceil$ was earlier shown in [13].

Here, we show that for a tree, the path cover number is equal to the lower bound.

Proposition 7. *Let T be a tree with n vertices, among which there are ℓ leaves. Then we have $\lceil \frac{\ell}{2} \rceil = \pi_c(T) = 1 + \lceil \frac{\sum_{i=1}^{k} \{\rho(u_i)-2\}}{2} \rceil$, where $U = \{u_1, \ldots, u_k\}$ is the set of all cut-vertices of the tree T.*

Proof. Since T is a tree with n vertices and ℓ leaves, all $n - \ell$ non-leaf vertices are cut-vertices; thus $k = n - \ell$. Also, $\rho(u_i) = deg(u_i)$ for all $u_i \in U$.

Now, $2(n-1) = \ell + \sum_{i=1}^{k} deg(u_i) \implies 2n - 2 = n - k + \sum_{i=1}^{k} \rho(u_i)$
$\implies n - k - 2 = -2k + \sum_{i=1}^{k} \rho(u_i) \implies \ell - 2 = \sum_{i=1}^{k} \{\rho(u_i) - 2\}$
$\implies \lceil \frac{\ell}{2} - 1 \rceil = \lceil \frac{\sum_{i=1}^{k} \{\rho(u_i)-2\}}{2} \rceil \implies \lceil \frac{\ell}{2} \rceil = 1 + \lceil \frac{\sum_{i=1}^{k} \{\rho(u_i)-2\}}{2} \rceil$ □

Proposition 8. *Let T_G be the block-cut tree of a connected graph G. If each block of G contains at most two cut vertices, each block containing two cut vertices has a path between its two cut vertices that covers all other vertices of that block, and each block containing one cut vertex has a path starting at the cut vertex and covering all other vertices of that block, then the graph G reaches the lower bound given in Theorem 4.*

Proof. By Theorem 6, there is an optimal path cover $\mathcal{P}^T$ of T_G of size $\lceil \frac{\ell}{2} \rceil$, where ℓ is the number of leaves of T_G.

Note that there is a path associated to each block of G that covers all the vertices of that block. These associated paths have one or both endpoints at the cut-vertices depending on whether the corresponding blocks contain one or two cut vertices. For each path of $\mathcal{P}^T$, construct a path in G by following its trajectory, replacing the $\tilde{u}_i$'s by the corresponding cut vertex u_i's and the b_j's by the path associated to the corresponding block B_j's. Thus, we obtain a path cover of G of size $\lceil \frac{\ell}{2} \rceil$.

Let $deg_H(v)$ denote the degree of the vertex v in the graph H. Then, using the notations from the definition of a block-cut tree T_G, we have
$\sum_{v \in V_T} deg_{T_G}(v) = 2(|V_T| - 1) = \ell + \sum_{i=1}^{k} deg_{T_G}(\tilde{u}_i) + 2(|V_T| - \ell - k)$
$\implies \ell = 2 + \sum_{i=1}^{k} \{deg_{T_G}(\tilde{u}_i) - 2\} \implies \ell = 2 + \sum_{i=1}^{k} \{\rho(u_i) - 2\}$
$\implies \lceil \frac{\ell}{2} \rceil = 1 + \lceil \frac{\sum_{i=1}^{k} \{\rho(u_i)-2\}}{2} \rceil$
Hence, the result follows. □

Remark 9. There exist graphs for which the path cover number is strictly larger than the bound of Theorem 4. For an example, consider the block graph G in Fig. 2; here $\pi_c(G) = 3 > 2 = 1 + \lceil \frac{2+0+0}{2} \rceil$.

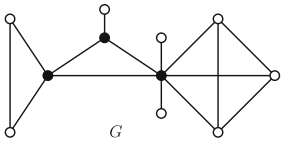

Fig. 2. A graph G with $\pi_c(G) = 3$.

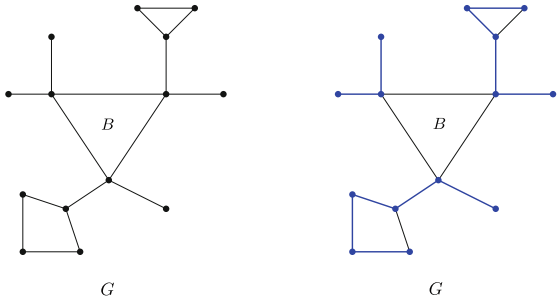

Fig. 3. For the graph G, no edge of its block B is covered by the optimal path cover shown by blue paths. (Color figure online)

2.2 Lower Bound for General Graphs Using Their Block-Cut Tree

Let G be a connected graph with block-cut tree T_G. Then we have the following.

Lemma 10. $\pi_c(T_G) \leq \pi_c(G)$.

Proof. Let $\mathcal{P}$ be an optimal path cover of G. Any path of $\mathcal{P}$ may or may not cover all the edges of the block(s) in which its two endpoints lie. We first modify the paths of $\mathcal{P}$ to construct another optimal path cover $\mathcal{P}'$ such that each path of $\mathcal{P}'$ covers at least one edge of the block(s) in which its two endpoints lie.

Now, consider the blocks in G. For a block of G, the paths in $\mathcal{P}'$ may cover some edges of it or may not cover any edge of the block (See Fig. 3). We first find all those blocks in G for which no edge is covered by the paths of $\mathcal{P}'$. Note that all the vertices of such a block must be cut vertices. Let (u_i, u_j) be an edge of such a block. Since $\mathcal{P}'$ is a path cover, the vertices u_i and u_j will be covered by at least two different paths $P'_1(= P'_{1_l} u_i P'_{1_r}), P'_2(= P'_{2_l} u_j P'_{2_r})$ (say) of $\mathcal{P}'$. We can modify the paths P'_1, P'_2 to obtain two new paths $P''_1 = P'_{1_l} u_i u_j P'_{2_l}$ and $P''_2 = P'_{1_r} u_i u_j P'_{2_r}$ so that the paths P''_1, P''_2 cover all the vertices and edges covered by P'_1, P'_2 as well as the edge (u_i, u_j) (See Fig. 4). Using this method a linear number of times, we construct another optimal path cover $\mathcal{P}''$ such that the paths in $\mathcal{P}''$ cover at least one edge of the block(s) in which their endpoints lie, as well as cover at least one edge of each block of G.

From this optimal path cover $\mathcal{P}''$ of G, we construct a path cover of T_G. A path P_G of $\mathcal{P}''$ starts at some block B_{x_1} of G, covers some vertices of that block, and enters into another block B_{x_2} (say) through a cut vertex $u_{x_1 x_2}$ (say) which is shared by B_{x_1} and B_{x_2}, and continues similarly. It finally ends at some

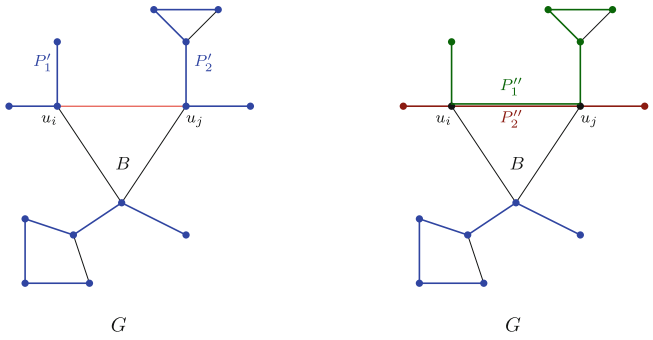

Fig. 4. The modified paths P_1'' (in green) and P_2'' (in brown) cover the edge (u_i, u_j). (Color figure online)

block B_{x_y} covering some vertices of this block. For such a path P_G, we get a corresponding path $b_{x_1}\tilde{u}_{x_1x_2}b_{x_2}\ldots b_{x_y}$ in T_G by following the trajectory of P_G in G. Thus, we get a set $\mathcal{P}_T$ (say) of $|\mathcal{P}''| = \pi_c(G)$ many paths in T_G. Due to the construction procedure of $\mathcal{P}''$, the paths of $\mathcal{P}_T$ will cover all the vertices of T_G. □

Lemma 10 gives a lower bound for the path cover number in general graphs.

Theorem 11. *For a graph $G = (V, E)$, if T_G is its block-cut tree and ℓ is the number of leaves of T_G, then $\pi_c(G) \geq \lceil \frac{\ell}{2} \rceil$.*

A graph is called a *block graph* if each of its blocks is a clique. The path cover number is equal to the lower bound given in Theorem 11 for trees, graphs having a Hamiltonian path, and block graphs (the latter can be deduced from analyzing the PhD thesis [10]). The sketch of an algorithm to find a minimum path cover of a block graph was also given in [10]. Analyzing this algorithm shows that it runs in linear time.

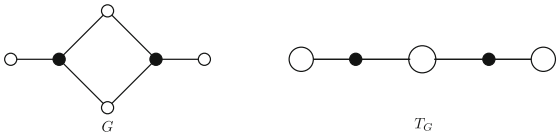

Fig. 5. A graph with path cover number strictly greater than the lower bound given using the block-cut tree.

Remark 12. There are graphs with path cover number strictly greater than the lower bound given in Theorem 11. See Fig. 5, where $\pi_c(G)(= 2) > 1 = \lceil \frac{\ell}{2} \rceil$, where ℓ is the number of leaves of the block-cut tree of the graph G.

3 A Linear-Time Algorithm for Outerplanar Graphs

We now show that block-cut trees can also lead to a linear-time algorithm to determine optimal path covers of outerplanar graphs. The algorithm is more involved than the one for block graphs [10], because the path cover number of an outerplanar graph is not necessarily equal to the one of its block-cut tree, as seen in Fig. 5.

3.1 Lower Bound for Path Cover Number in Outerplanar Graphs

Let $G = (V, E)$ be a connected outerplanar graph. We can construct the block-cut tree T_G of the graph G in linear time using Lemma 2. Let us call the vertices of T_G that correspond to the blocks in G the *block vertices*.

We distinguish between three types of block vertices in T_G, namely (i) degree 1 block vertices; we call them leaf block vertices, (ii) degree 2 block vertices, and (iii) degree > 2 block vertices. The corresponding blocks of types (i), (ii), and (iii) in G are called Type-1 blocks (or leaf blocks), Type-2 blocks, and Type-3 blocks, respectively. The vertices of T_G corresponding to the cut vertices of G are called (with slight abuse of the nomenclature) cut tree vertices.

Each Type-2 block in G has exactly two cut vertices. For each such block, we check whether there exists a path that covers all the vertices of that block and whose endpoints are the two cut vertices of that block. If no such path exists for a Type-2 block, we mark that block as a special block. The corresponding degree 2 block vertex in T_G is called a special block vertex. Since each Type-2 block of G is a biconnected outerplanar graph G', finding the mentioned path is equivalent to finding a Hamiltonian path of G' with two specified endpoints, which can be done in linear time using standard treewidth-based techniques [2,6], as outerplanar graphs have treewidth at most 2. Hence, finding all special blocks of G can be done in linear time.

Construction of the Tree $\tilde{T}_G$ from the Block-cut Tree T_G: From each leaf block vertex of T_G, move towards the interior of the tree T_G following the path consisting of only the degree 2 cut tree vertices and the degree 2 block vertices (excluding the special block vertices). This process will lead to encounter exactly one of the following four cases: (i) a degree > 2 block vertex, (ii) a degree > 2 cut tree vertex, (iii) a special block vertex, or (iv) another leaf block vertex. Stop when for the first time one of these four cases is encountered. If in this process, while stopping, case (iii) i.e., a special block vertex is encountered, then we name that special block vertex a bad block vertex. So, in linear time, we can find all bad block vertices[1]. Then, in T_G, a dummy leaf vertex is added to each bad block vertex. The tree obtained after adding all the dummy leaf vertices is called $\tilde{T}_G$ (See Fig. 6 for an example).

Lemma 13. $\pi_c(G) \geq \pi_c(\tilde{T}_G)$.

[1] After this process, there may remain special block vertices that are not bad. So, every bad block vertex is special, but in general the converse need not be true.

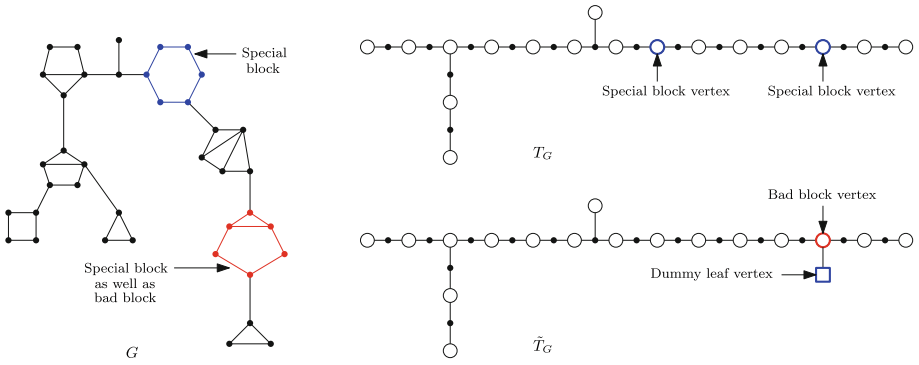

Fig. 6. An outerplanar graph G, its block cut tree T_G and its corresponding $\tilde{T}_G$. (Color figure online)

Proof. Given an optimal path cover of G, we will construct a path cover of $\tilde{T}_G$ of the same size. Let $\mathcal{P}$ be an optimal path cover of G. By the method used in the proof of Lemma 10, we construct another optimal path cover $\mathcal{P}''$ of G such that the paths in $\mathcal{P}''$ cover at least one edge of the block(s) in which their endpoints lie, as well as cover at least one edge of each block of G.

From $\mathcal{P}''$, we will construct a path cover of $\tilde{T}_G$ consisting of $|\mathcal{P}''| = |\mathcal{P}|$ many paths. A path $P_G \in \mathcal{P}''$ begins from some block B_{x_1} of G, covers some vertices of that block, and enters into another block B_{x_2} (say) through a cut vertex $u_{x_1x_2}$ (say) which is shared by B_{x_1} and B_{x_2}, and continues in a similar way. It ends at some block B_{x_y} after covering some vertices of this block. For such a path P_G, a corresponding path $b_{x_1}\tilde{u}_{x_1x_2}b_{x_2}\ldots b_{x_y}$ in $\tilde{T}_G$ is constructed by following the trajectory of P_G in G. In this manner, we obtain a set $\tilde{\mathcal{P}}_T$ (say) of $|\mathcal{P}''| = \pi_c(G)$ number of paths in $\tilde{T}_G$. The properties of $\mathcal{P}''$ ensure that the paths of $\tilde{\mathcal{P}}_T$ will cover all the vertices of $\tilde{T}_G$ except the dummy leaf vertices.

For a dummy leaf vertex, consider its corresponding bad block vertex. If there is a path in $\tilde{\mathcal{P}}_T$ whose one endpoint is this bad block vertex, we extend that path to cover the dummy leaf vertex. Otherwise, due to the definition of bad block, there exist at least two paths in $\tilde{\mathcal{P}}_T$ passing through the bad block vertex. One of these paths can be extended to cover the branch of $\tilde{T}_G$ emerging from the bad block vertex containing only one leaf block vertex, and the other path can be extended to cover the dummy leaf vertex[2]. Thus, we get a set of $\pi_c(G)$ number of paths in $\tilde{T}_G$ which cover all the vertices of $\tilde{T}_G$. Hence, the result follows. □

[2] Note that in $\tilde{T}_G$, the bad block vertex has three branches emerging from it; one containing only the dummy leaf vertex, one containing only one leaf block vertex, and another containing one or more leaf block vertices.

3.2 The Algorithm

An optimal path cover of a connected outerplanar graph G is computed in two steps: (i) Finding an optimal path cover of $\tilde{T}_G$ satisfying some required properties and (ii) Constructing a path cover of G from the computed path cover of $\tilde{T}_G$.

(i) Optimal Path Cover of $\tilde{T}_G$: Without loss of generality, assume that $\tilde{T}_G$ contains at least two vertices of degree ≥ 3. From each leaf vertex of $\tilde{T}_G$[3], move towards the interior of $\tilde{T}_G$ through the path consisting of only degree 2 vertices of $\tilde{T}_G$ and stop as soon as a vertex of degree ≥ 3 is encountered. Label this degree ≥ 3 vertex as truncating vertex. In linear time, all the truncating vertices can be found. Also, find a truncating vertex having the least degree in $\tilde{T}_G$ among all truncating vertices; and name it special truncating vertex.

Consider all the branches of $\tilde{T}_G$ emerging from a truncating vertex. Some of these branches contain only degree ≤ 2 nodes, name them as truncating branches. For each truncating vertex (including the special truncating vertex), we find the corresponding truncating branches. Now, removing all the truncating branches from $\tilde{T}_G$ we obtain a tree $\mathcal{T}$ (say). Observe that a leaf of $\mathcal{T}$ is a truncating vertex of $\tilde{T}_G$ and vice versa. Now, there are two cases:

Case-1: The number of leaves of $\mathcal{T}$ is odd: In this case, consider the leaf of $\mathcal{T}$ which was the special truncating vertex of $\tilde{T}_G$. In $\mathcal{T}$, starting from this leaf, move towards the interior of $\mathcal{T}$ through the path consisting of only degree 2 vertices of $\mathcal{T}$ and stop as soon as a vertex with degree ≥ 3 in $\mathcal{T}$ is encountered; name this vertex trimming branch root and the branch trimming branch. Delete this traversed branch (trimming branch) from $\mathcal{T}$. Thus, we get a subtree of $\mathcal{T}$ with even number of leaves; name it $\mathcal{T}'$.

Now, find an optimal path cover $\mathcal{C}'$ of $\mathcal{T}'$ in linear time using the algorithm from [12]. Extend the paths (if required) so that their endpoints are leaves of $\mathcal{T}'$. Let the optimal path cover thus obtained be denoted by $\mathcal{C}''$.

Now, consider the tree $\mathcal{T}$. Take two copies of each path of $\mathcal{C}''$ in $\mathcal{T}$. Since the trimming branch root has degree at least 2 in $\mathcal{T}'$, two copies of some path P_r (say) of $\mathcal{C}''$ will pass through it. Take one copy of the path P_r, cut this path at the trimming branch root and use the two pieces to form two paths P_{r_1} and P_{r_2} both of which cover the trimming branch[4] (See Fig. 7 for an example of the mentioned construction for a tree $\tilde{T}_G$ where the corresponding $\mathcal{T}$ has odd number of leaves). Thus, we get a collection of $2|\mathcal{C}''| + 1$ paths such that each vertex of $\mathcal{T}$ is covered at least twice by these paths and each leaf of $\mathcal{T}$ is endpoint of exactly two paths.

Case-2: The number of leaves of $\mathcal{T}$ is even: In this case, find an optimal path cover $\mathcal{C}$ of $\mathcal{T}$ in linear time using the algorithm from [12]. Extend the paths (if required) so that their endpoints are leaves of $\mathcal{T}$. Let the optimal path cover thus obtained be denoted by $\mathcal{C}^*$. Take two copies of each path of $\mathcal{C}^*$. Hence, we

[3] By the leaf vertices of $\tilde{T}_G$, we mean both the original leaf vertices as well as the dummy leaf vertices of $\tilde{T}_G$.

[4] The trimming branch contains the special truncating vertex as one of its endpoint.

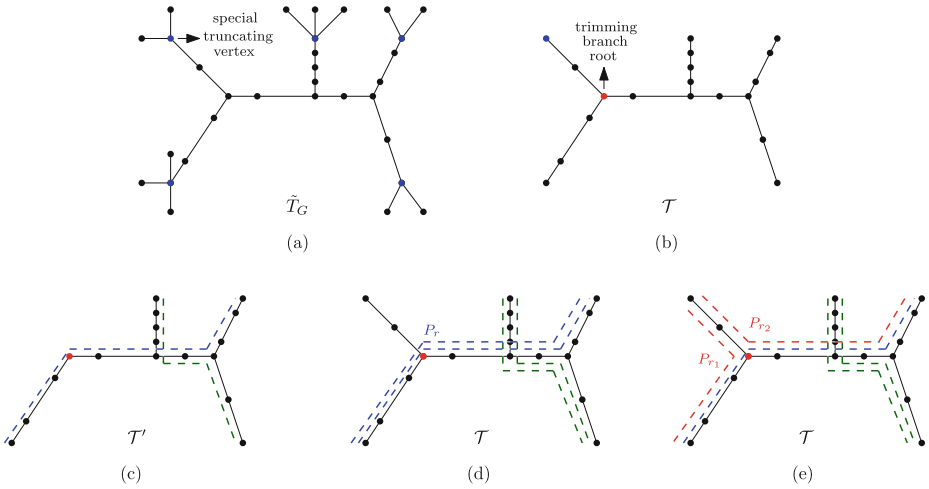

Fig. 7. (a) Truncating vertices (marked in blue) of a tree $\tilde{T}_G$, (b) the corresponding $\mathcal{T}$ with odd number of leaves, (c) the path cover $\mathcal{C}''$ of the corresponding $\mathcal{T}'$, (d) two copies of each path of $\mathcal{C}''$ in $\mathcal{T}$, (e) a collection of $2|\mathcal{C}''|+1$ paths that cover each vertex of $\mathcal{T}$ at least two times.(Color figure online)

obtain a collection of $2|\mathcal{C}^*|$ paths such that each vertex of $\mathcal{T}$ is covered at least twice by these paths and each leaf of $\mathcal{T}$ is endpoint of exactly two paths.

Next, in $\mathcal{T}$, add the deleted (truncating) branches to get back $\tilde{T}_G$. Extend the paths ending at each truncating vertex to cover as many of the added branches as possible. After extending all such paths, some added branches may still remain uncovered. If there are $\tilde{l}$ many uncovered branches, then they are covered by $\lceil \frac{\tilde{l}}{2} \rceil$ many paths which can be found in linear time by arbitrarily pairing any two branches and joining them by a truncating vertex when the branches correspond to the same truncating vertex, or by a path joining two truncating vertices (which exists since $\mathcal{T}$ is connected) when the branches correspond to two different truncating vertices.

Thus, the constructed collection $\mathcal{C}_T$ (say) of $l_{\mathcal{T}} + \lceil \frac{\tilde{l}}{2} \rceil$ paths covers all the vertices of $\tilde{T}_G$, where $l_{\mathcal{T}}$ denotes the number of leaves of the tree $\mathcal{T}$. Also, $l_{\mathcal{T}} + \lceil \frac{\tilde{l}}{2} \rceil = \lceil \frac{1}{2} \cdot (\text{no. of leaves of } \tilde{T}_G) \rceil = \pi_c(\tilde{T}_G)$ by our construction. Hence, $\mathcal{C}_T$ is an optimal path cover of $\tilde{T}_G$.

(ii) Construction of a Path Cover of G: Corresponding to each path of $\mathcal{C}_T$ in $\tilde{T}_G$, we can get a path in G by following the trajectory of the path. We want to form those paths in G in such a way that all the vertices of each block of G are covered. Due to the construction of $\mathcal{C}_T$, at least two paths will pass through each special block and each Type-3 block. So, all the Type-2 blocks and leaf blocks can be covered. For Type-3 blocks, there are two cases:

Case 1: There are exactly three cut-vertices u_1, u_2, u_3 (say) on the unique Hamiltonian cycle[5] of the Type-3 block B_i (say) under consideration and one path of $\mathcal{C}_T$ passing through b_i traverses as $\ldots \tilde{u}_1 b_i \tilde{u}_3 \ldots$ in $\tilde{T}_G$. Then we create the corresponding path in G by taking the path from u_1 to u_3 that contains u_2 on the unique Hamiltonian cycle (see Fig. 8(a)). The other path passing through b_i covers the vertex $\tilde{u}_2$ and goes through $\tilde{u}_3$ (or $\tilde{u}_1$). In G, we take that path from u_2 to u_3 (or u_1) on the unique Hamiltonian cycle which will cover all the remaining uncovered vertices of B_i (see Fig. 8(b)) as its corresponding path.

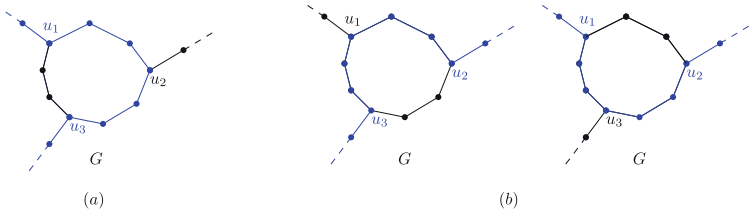

Fig. 8. (a) Path (marked in blue) in G which corresponds to the path $\ldots \tilde{u}_1 b_i \tilde{u}_3 \ldots$ of $\tilde{T}_G$. (b) Path (marked in blue) in G which corresponds to the path $\ldots \tilde{u}_2 b_i \ldots$ in the tree $\tilde{T}_G$. (Color figure online)

Case 2: There are at least four cut-vertices u_1, u_2, u_3, u_4 (say) in this order on the Hamiltonian cycle, and there are two paths in $\mathcal{C}_T$ such that each path uses two of these cut-vertices. Note that there may exist some other cut-vertices in between these four vertices. In this case there are two sub-cases, (i) if one path of $\mathcal{C}_T$ traverses as $\ldots \tilde{u}_1 b_i \tilde{u}_3 \ldots$ and another path as $\ldots \tilde{u}_2 b_i \tilde{u}_4 \ldots$ (see Fig. 9(a)), then we create two new paths $\ldots \tilde{u}_1 b_i \tilde{u}_2 \ldots$ and $\ldots \tilde{u}_3 b_i \tilde{u}_4 \ldots$ in such a way that the vertices of $\tilde{T}_G$ covered by the previous two paths and these two paths are the same. Then the corresponding paths in G are created as a path from u_1 to u_2 that contains u_4 and u_3 on the unique Hamiltonian cycle and a path from u_3 to u_4 that contains u_2 and u_1 on the unique Hamiltonian cycle (see Fig. 9(b)). (ii) if one path of $\mathcal{C}_T$ traverses as $\ldots \tilde{u}_1 b_i \tilde{u}_2 \ldots$ and another path as $\ldots \tilde{u}_3 b_i \tilde{u}_4 \ldots$, then the paths in G are created in the same manner (see Fig. 9(b)). Hence, we obtain a path cover of G of size $|\mathcal{C}_T| = \pi_c(\tilde{T}_G)$.

Correctness and Time Complexity of the Algorithm: From Lemma 13, we already have $\pi_c(\tilde{T}_G) \leq \pi_c(G)$. Also, the constructed path cover of G has $\pi_c(\tilde{T}_G)$ many paths. Hence, the algorithm gives a minimum path cover of G.

The constructions of T_G from G and $\tilde{T}_G$ from T_G both require linear time. Finding all the truncating vertices and constructing the tree $\mathcal{T}$ can be done in linear time. Finding $l_{\mathcal{T}}$ many paths as described in the algorithm requires linear time irrespective of whether $\mathcal{T}$ has odd or even number of leaves. Thus, the path cover $\mathcal{C}_T$ of $\tilde{T}_G$ can be constructed in linear time. Finally, the construction of

[5] Each block of G being biconnected outerplanar has a unique Hamiltonian cycle [22].

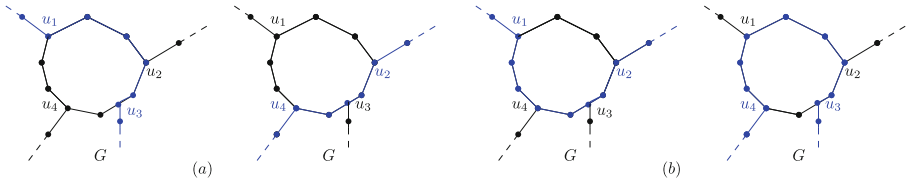

Fig. 9. (a) Paths (marked in blue) in G which corresponds to the paths $\dots \tilde{u}_1 b_i \tilde{u}_3 \dots$ and $\dots \tilde{u}_2 b_i \tilde{u}_4 \dots$ of $\tilde{T}_G$, respectively. These paths do not cover all the vertices of B_i in G. (b) Paths (marked in blue) after transforming the paths of Fig.9.(a). Here the paths correspond to $\dots \tilde{u}_1 b_i \tilde{u}_2 \dots$ and $\dots \tilde{u}_3 b_i \tilde{u}_4 \dots$ of the tree $\tilde{T}_G$ and covers all the vertices of B_i in G. (Color figure online)

a path cover of G from $\mathcal{C}_T$ requires linear number of steps. Hence, we have the following result.

Theorem 14. *An optimal path cover of an outerplanar graph can be constructed in linear time.*

4 Conclusion

We have studied some lower bounds for the path cover number in arbitrary graphs. Some graph classes attaining this bound have been explored. A linear-time algorithm for the path cover problem for the class of outerplanar graphs has been proposed. Improving the lower bound and exploring the path cover problem for some other graph classes is an interesting future research direction.

References

1. Andreatta, G., Mason, F.: Path covering problems and testing of printed circuits. Discret. Appl. Math. **62**(1–3), 5–13 (1995)
2. Beisegel, J., Ratazczak, F., Scheffler, R.: Computing Hamiltonian paths with partial order restrictions. ACM Trans. Comput. Theory **17**(1(5)), 1–24 (2025)
3. Bourneuf, R., Planken, T.: A structural linear-time algorithm for computing the Tutte decomposition (2025). arXiv:2508.06212
4. Cáceres, M., Mumey, B., Toivonen, S., Tomescu, A.I.: Practical minimum path cover. In: Liberti, L., (ed.), 22nd International Symposium on Experimental Algorithms, SEA 2024, Article 3, pp. 1–19. (Leibniz International Proceedings in Informatics, LIPIcs; Vol. 301). Schloss Dagstuhl - Leibniz-Zentrum für Informatik
5. Chakraborty, D., Foucaud, F., Parreau, A., Wagler, A.K.: On three domination-based identification problems in block graphs. Fund. Inform. **191**(3–4), 197–229 (2024)
6. Cygan, M., et al.: Parameterized Algorithms. Springer Cham (2015)
7. Donald, A.: An upper bound for the path number of a graph. J. Graph Theory **4**, 189–201 (1980)
8. Fernau, H., Foucaud, F., Mann, K., Padariya, U., Rao, R.K.N: Parameterizing path partitions; Theoretical Computer Science **1028**, 115029 (2025)

9. Fisher, D.C., Fitzpatrick, S.L.: The isometric path number of a graph, J. Comb. Math. Comb. Comput. **38**, 97–110 (2001)
10. Florkowski, S.F.: Extensions of the Hamiltonian Cycle/Path Problems in Special Families of Graphs. PhD thesis, Lehigh University (2017)
11. Foucaud, F., Kovse, M.: Identifying path covers in graphs. J. Discrete Algorithms **23**, 21–34 (2013)
12. Foucaud, F., Majumder, A., Mömke, T., Roshany-Tabrizi, A.: Polynomial-time algorithms for path cover on trees and graphs of bounded treewidth. In: Proceedings of the 11th International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2025), Lecture Notes in Computer Science, vol. 15536, pp. 147–159 (2025)
13. Harary, F., Schwenk, A.: Evolution of the path number of a graph: covering and packing in graphs, II. Graph Theory Comput. 39–45 (1972)
14. Hopcroft, J.E., Tarjan, R.E.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (1973)
15. Hung, R., Chang, M.: Solving the path cover problem on circular-arc graphs by using an approximation algorithm. Discret. Appl. Math. **154**(1), 76–105 (2006)
16. Lin, R., Olariu, S., Pruesse, G.: An optimal path cover algorithm for cographs. Comput. Math. Appl. **30**(8), 75–83 (1995)
17. Manuel, P.: Revisiting path-type covering and partitioning problems (2018). arXiv:1807.10613
18. Ntafos, S., Hakimi, S.: On path cover problems in digraphs and applications to program testing. IEEE Trans. Softw. Eng. **SE-5**(5), 520–529 (1979)
19. Pan, J., Chang, G.J.: Path partition for graphs with special blocks. Discret. Appl. Math. **145**(3), 429–436 (2005)
20. Peroche, B.: NP-completeness of some problems of partitioning and covering in graphs. Discret. Appl. Math. **8**(2), 195–208 (1984)
21. Sinkovic, J.: Maximum nullity of outerplanar graphs and the path cover number. Linear Algebra Appl. **432**(8), 2052–2060 (2010)
22. Syslo, M.: Characterizations of outerplanar graphs. Discrete Math. **26**, 47–53 (1979)
23. Yeh, H., Chang, G.J.: The path-partition problem in bipartite distance-hereditary graphs. Taiwan. J. Math. **2**(3), 353–360 (1998)