On Minimum Average Stretch Spanning Trees in Polygonal 2-trees *

N.S.Narayanaswamy and G.Ramakrishna

Department of Computer Science and Engineering
Indian Institute of Technology Madras, India.
{swamy,grama}@cse.iitm.ac.in

Abstract. A spanning tree of an unweighted graph is a minimum average stretch spanning tree if it minimizes the ratio of sum of the distances in the tree between the end vertices of the graph edges and the number of graph edges. We consider the problem of computing a minimum average stretch spanning tree in polygonal 2-trees, a super class of 2-connected outerplanar graphs. For a polygonal 2-tree on \( n \) vertices, we present an algorithm to compute a minimum average stretch spanning tree in \( O(n \log n) \) time. This also finds a minimum fundamental cycle basis in polygonal 2-trees.

1 Introduction

Average stretch is a parameter used to measure the quality of a spanning tree in terms of distance preservation, and finding a spanning tree with minimum average stretch is a classical problem in network design. Let \( G = (V(G), E(G)) \) be an unweighted graph and \( T \) be a spanning tree of \( G \). For an edge \((u, v) \in E(G)\), \( d_T(u, v) \) denotes the distance between \( u \) and \( v \) in \( T \). The average stretch of \( T \) is defined as

\[
\text{AvgStr}(T) = \frac{1}{|E(G)|} \sum_{(u,v) \in E(G)} d_T(u,v)
\]

A minimum average stretch spanning tree of \( G \) is a spanning tree that minimizes the average stretch. Given an unweighted graph \( G \), the minimum average stretch spanning tree (MAST) problem is to find a minimum average stretch spanning tree of \( G \). Due to the unified notation for tree spanners, the MAST problem is equivalent to the problem, MFCB, of finding a minimum fundamental cycle basis in unweighted graphs [17]. Minimum average stretch spanning trees are used to solve symmetric diagonally dominant linear systems [17].

Further, minimum fundamental cycle bases have various applications including determining the isomorphism of graphs, frequency analysis of computer programs, and generation of minimal perfect hash functions (See [4,11] and the references there in). Due to these vast applications, finding a minimum average stretch spanning tree is useful in theory and practice. The MAST problem was studied in a graph theoretic game in the context of the \( k \)-server problem by Alon et al. [1]. The MFCB problem was introduced by Hubika and Syslo in 1975 [12]. The MFCB problem was proved to be NP-complete by Deo et al. [4] and APX-hard by Galbiati et al. [11]. Another closely related problem is the problem of probabilistically embedding a graph into its spanning trees. A graph \( G \) is said to be probabilistically embedded into its spanning trees with distortion \( t \), if there is a probability distribution \( D \) of spanning trees of \( G \), such that for any two vertices the expected stretch of the spanning trees in \( D \) is at most \( t \). The problem of probabilistically embedding a graph into its spanning trees with low distortion has interesting connections with low average stretch spanning trees.

In the literature, spanning trees with low average stretch has received significant attention in special graph classes such as \( k \)-outerplanar graphs and series-parallel graphs. In case of planar graphs, Kavitha et al. remarked that the complexity of MFCB is unknown and there is no \( O(\log n) \) approximation algorithm [13]. For \( k \)-outerplanar graphs, the technique of peeling-an-onion decomposition is employed to obtain a spanning tree whose average stretch is at most \( c^k \), where \( c \) is a constant [7]. In case of series-parallel graphs, a spanning tree with average stretch at most \( O(\log n) \) can be obtained in polynomial time (See Section 5 in [8]). The bounds on the size of a minimum fundamental cycle basis is studied in graph classes such as planar,

* Supported by the Indo-Max Planck Centre for Computer Science Programme in the area of Algebraic and Parameterized Complexity for the year 2012 - 2013
outerplanar and grid graphs [13]. The study of probabilistic embeddings of graphs is discussed in [7,8]. To the best of our knowledge, there is no published work to compute a minimum average stretch spanning tree and minimum fundamental cycle basis in any subclass of planar graphs.

We consider polygonal 2-trees in this work, which are also referred to as polygonal-trees. They have a rich structure that make them very natural models for biochemical compounds, and provide an appealing framework for solving associated enumeration problems.

**Definition 1 ([14]).** A graph is a polygonal 2-tree if it can be obtained by edge-gluing a set of cycles successively.

Edge gluing on two graphs $G_1$ and $G_2$ results in a graph $G$ such that $V(G) = V(G_1) \cup V(G_2)$, $E(G) = E(G_1) \cup E(G_2)$, $|V(G_1) \cap V(G_2)| = 2$ and $|E(G_1) \cap E(G_2)| = 1$. A graph is a $k$-gonal tree, if it can be obtained by edge-gluing a set of cycles of length $k$ successively [14]. For example, a 2-tree is a 3-gonal tree. The class of polygonal 2-trees is a subclass of planar graphs and it includes 2-connected outerplanar graphs and $k$-gonal trees. 2-trees, in other words 3-gonal trees, are extensively studied in the literature. In particular, previous work on various flavours of counting and enumeration problems on 2-trees is compiled in [10]. Formulas for the number of labeled and unlabeled $k$-gonal trees with $r$ polygons (induced cycles) are computed in [15]. The family of $k$-gonal trees with same number of vertices is claimed as a chromatic equivalence class by Chao and Li, and the claim has been proved by Wakelin and Woodal [14]. The class of polygonal 2-trees is shown to be a chromatic equivalence class by Xu [14]. Further, various subclasses of generalized polygonal 2-trees have been considered, and it has been shown that they also form a chromatic equivalence class [14,19,20]. The enumeration of outerplanar $k$-gonal trees is studied by Harary, Palmer and Read to solve a variant of the cell growth problem [6]. Molecular expansion of the species of outerplanar $k$-gonal trees is shown in [6]. Also outerplanar $k$-gonal trees are of interest in combinatorial chemistry, as the structure of chemical compounds like catacondensed benzenoid hydrocarbons forms an outerplanar $k$-gonal tree.

### 1.1 Our Results

We state our main theorem.

**Theorem 2.** Given a polygonal 2-tree $G$ on $n$ vertices, a minimum average stretch spanning tree of $G$ can be obtained in $O(n \log n)$ time.

A quick overview of our approach to solve MAST is presented in **Algorithm 1** below. The detailed implementation is given in Section 4.

```
Algorithm 1: An algorithm to find an MAST of a polygonal 2-tree $G$
1   $A \leftarrow \emptyset$;
2   for each edge $e \in E(G)$ do $c[e] \leftarrow 0$;
3   while $G - A$ has a cycle do
4     Choose an edge $e$ from $G - A$, such that $e$ belongs to exactly one induced cycle in $G - A$ and $c[e]$ is minimum;
5     Let $C$ be the induced cycle containing $e$ in $G - A$;
6     for each $\hat{e} \in E(C) \setminus \{e\}$ do $c[\hat{e}] \leftarrow c[\hat{e}] + c[e] + 1$;
7     $A \leftarrow A \cup \{e\}$;
8   Return $G - A$;
```

Due to the equivalence of MAST and MFCB (shown in Lemma 5), our result implies the following corollary. For a set $B$ of cycles in $G$, the size of $B$, denoted by size($B$), is the number of edges in $B$ counted according to their multiplicity.
We consider simple, connected, unweighted and undirected graphs. We use standard graph terminology as a graph with vertex set \( V \) and edge set \( E \) respectively in \( G \). We denote \( |V(G)| \) by \( n \) and \( |E(G)| \) by \( m \).

The union of graphs \( G_1 \) and \( G_2 \) is defined as a graph with vertex set \( V(G_1) \cup V(G_2) \) and edge set \( E(G_1) \cup E(G_2) \) and is denoted by \( G_1 \cup G_2 \). The intersection of graphs \( G_1 \) and \( G_2 \) written as \( G_1 \cap G_2 \) is a graph with vertex set \( V(G_1) \cap V(G_2) \) and edge set \( E(G_1) \cap E(G_2) \). The removal of a set \( X \) of edges from \( G \) is denoted by \( G \setminus X \). For a set \( X \subset V(G) \), \( G[X] \) denotes the induced graph on \( X \). An edge \( e \in E(G) \) is a cut-edge (bridge) if \( G - e \) is disconnected. A graph is 2-connected if it can not be disconnected by removing less than two vertices. A 2-connected component of \( G \) is a maximal 2-connected subgraph of \( G \).

Let \( T \) be a spanning tree of \( G \). An edge \( e \in E(G) \setminus E(T) \) is a non-tree edge of \( T \). For a non-tree edge \((u,v)\) of \( T \), a cycle formed by the edge \((u,v)\) and the unique path between \( u \) and \( v \) in \( T \) is referred to as a fundamental cycle. For an edge \((u,v)\) of \( G \), stretch of \((u,v)\) is the distance between \( u \) and \( v \) in \( T \). The total stretch of \( T \) is defined as the sum of the stretches of all the edges in \( G \). We remark that there are slightly different definitions exit in the literature to refer the average stretch of a spanning tree. We use the definition in Equation 1 presented by Emek and Peleg in \([8]\), to refer the average stretch of a spanning tree.

Corollary 3. Given a polygonal 2-tree \( G \) on \( n \) vertices, a minimum fundamental cycle basis \( B \) of \( G \) can be obtained in \( O(n \log n + \text{size}(B)) \) time.

We characterize polygonal 2-trees using a kind of ear decomposition and present the structural properties of polygonal 2-trees that are useful in finding a minimum average stretch spanning tree (In Section 2). We then identify a set of edges in a polygonal 2-tree, called safe edges, whose removal results in a minimum average stretch spanning tree (In Section 3). We present an algorithm with necessary data-structures to identify the safe set of edges efficiently and compute a minimum average stretch spanning tree in sub-quadratic time (In Section 4). Also, we characterize polygonal 2-trees using cycle basis (In Section 5).

A graph \( G \) can be probabilistically embedded into its spanning trees with distortion \( t \) if and only if the multigraph obtained from \( G \) by replicating its edges has a spanning tree with average stretch at most \( t \) (See \([1]\)). It is easy to observe that, a spanning tree \( T \) of \( G \) is a minimum average stretch spanning tree for \( G \) if and only if \( T \) is a minimum average stretch spanning tree for a multigraph of \( G \). As a consequence of our result, we have the following corollary.

Corollary 4. For a polygonal 2-tree \( G \), let \( t \) be the average stretch of a minimum average stretch spanning tree of \( G \). Then \( G \) can be probabilistically embedded into its spanning trees with distortion \( t \).

1.2 Graph Preliminaries

We consider simple, connected, unweighted and undirected graphs. We use standard graph terminology from \([24]\). Let \( G = (V(G), E(G)) \) be a graph, where \( V(G) \) and \( E(G) \) denote the set of vertices and edges, respectively in \( G \). We denote \( |V(G)| \) by \( n \) and \( |E(G)| \) by \( m \).

The union of graphs \( G_1 \) and \( G_2 \) is defined as a graph with vertex set \( V(G_1) \cup V(G_2) \) and edge set \( E(G_1) \cup E(G_2) \) and is denoted by \( G_1 \cup G_2 \). The intersection of graphs \( G_1 \) and \( G_2 \) written as \( G_1 \cap G_2 \) is a graph with vertex set \( V(G_1) \cap V(G_2) \) and edge set \( E(G_1) \cap E(G_2) \). The removal of a set \( X \) of edges from \( G \) is denoted by \( G \setminus X \). For a set \( X \subset V(G) \), \( G[X] \) denotes the induced graph on \( X \). An edge \( e \in E(G) \) is a cut-edge (bridge) if \( G - e \) is disconnected. A graph is 2-connected if it can not be disconnected by removing less than two vertices. A 2-connected component of \( G \) is a maximal 2-connected subgraph of \( G \).

Let \( T \) be a spanning tree of \( G \). An edge \( e \in E(G) \setminus E(T) \) is a non-tree edge of \( T \). For a non-tree edge \((u,v)\) of \( T \), a cycle formed by the edge \((u,v)\) and the unique path between \( u \) and \( v \) in \( T \) is referred to as a fundamental cycle. For an edge \((u,v)\) of \( G \), stretch of \((u,v)\) is the distance between \( u \) and \( v \) in \( T \). The total stretch of \( T \) is defined as the sum of the stretches of all the edges in \( G \). We remark that there are slightly different definitions exit in the literature to refer the average stretch of a spanning tree. We use the definition in Equation 1 presented by Emek and Peleg in \([8]\), to refer the average stretch of a spanning tree.

Proposition 14 in \([17]\) states that, \( T \) is a minimum total stretch spanning tree of \( G \) if and only if the set of fundamental cycles of \( T \) is a minimum fundamental cycle basis of \( G \). Then, we can have the following lemma.

Lemma 5. Let \( G \) be an unweighted graph and \( T \) be a spanning tree of \( G \). \( T \) is a minimum average stretch spanning tree of \( G \) if and only if the set of fundamental cycles of \( T \) is a minimum fundamental cycle basis of \( G \).

We use the following notation crucially. A path is a connected graph in which two vertices have degree one and the rest of vertices have degree two.

Lemma 6. Let \( G' \) be a 2-connected component in an arbitrary graph \( G \) and \( T \) be a subgraph of \( G \).

(a) If \( T \) is a spanning tree of \( G \), then \( T \cap G' \) is a spanning tree of \( G' \).

(b) If \( T \) is a path in \( G \), then \( T \cap G' \) is a path.

Proof. We first prove the following claim: If \( T \) is a tree, then \( T \cap G' \) is a tree.

Let \( T' = T \cap G' \). Suppose \( T' \) is not connected, then there exist two vertices \( x \) and \( y \) in \( V(G') \) such that there is no path between \( x \) and \( y \) in \( T' \). Since \( T \) is a tree, there is a path \( P \) between \( x \) and \( y \) in \( T \). Since \( T' \) is
not connected, we can observe that $V(P) \setminus V(G')$ contains at least one vertex, say, $u$. Further, the two edges incident on $u$ in $P$ are not in $G'$. Now we can obtain a graph $G' \cup P$ which is a 2-connected component in $G$. This contradicts the maximality of the 2-connected component $G'$. Therefore, $T'$ is connected. As $T'$ is acyclic, we conclude that $T'$ is a tree.

If $T$ is a spanning tree of $G$, then the set of vertices in $T \cap G'$ is $V(G')$. Therefore from the above claim, $T \cap G'$ is a spanning tree of $G'$. Thus (a) holds. Further, (b) also holds from the above claim. □

**Special Graph Classes.** A *partial 2-tree* is a subgraph of a 2-tree. A graph is a *series-parallel* graph, if it can be obtained from an edge, by repeatedly duplicating an edge or replacing an edge by a path. An alternative equivalent definition for series-parallel graphs is given in [9].

2 Structural Properties of Polygonal 2-trees

In this section, we present our key structural result in Lemma[10] which presents crucial structural properties of polygonal 2-trees. This lemma will be used significantly in proving the correctness of our algorithm. Another major result in this section is Theorem[13] which computes a kind of ear decomposition for polygonal 2-trees. This helps in obtaining an efficient algorithm to solve MAST. The notion of open ear decomposition is well known to characterize 2-connected graphs. An open ear decomposition of $G$ is a partition of $E(G)$ into a sequence $(P_0, \ldots, P_k)$ of edge disjoint graphs called as ears such that,

1. For each $i \geq 0$, $P_i$ is a path.
2. For each $i \geq 1$, end vertices of $P_i$ are distinct and the internal vertices of $P_i$ are not in $P_0 \cup \ldots \cup P_{i-1}$.

Further, a restricted version of open ear decomposition called nested ear decomposition is used to characterize series-parallel graphs [9]. An open ear decomposition $(P_0, \ldots, P_k)$ of $G$ is said to be nested if it satisfies the following properties:

1. For each $i \geq 1$, there exists $j < i$, such that the end vertices of path $P_i$ are in $P_j$.
2. Let the end vertices of $P_i$ and $P_j$ are in $P_j$, where $0 \leq j < i, i' \leq k$ and $i \neq i'$. Let $Q_i \subseteq P_i$ be the path between the end vertices of $P_i$ and $Q_{i'} \subseteq P_{i'}$ be the path between the end vertices of $P_{i'}$. Then $E(Q_i) \subseteq E(Q_{i'})$ or $E(Q_{i'}) \subseteq E(Q_i)$ or $E(Q_i) \cap E(Q_{i'}) = \emptyset$.

We define nice ear decomposition to characterize polygonal 2-trees and we show how it helps in efficiently computing the induced cycles. A nested ear decomposition $(P_0, \ldots, P_k)$ is said to be nice if it has the following property: $P_0$ is an edge and for each $i \geq 1$, if $x_i$ and $y_i$ are the end vertices of $P_i$, then there is some $j < i$, such that $(x_i, y_i)$ is an edge in $P_j$. A nice ear decomposition of a polygonal 2-tree is shown in Fig[11] Definition 1 naturally gives a nice ear decomposition for polygonal 2-trees. Further, a unique polygonal 2-tree can be constructed easily from a nice ear decomposition. Thus we have the following observation.

**Observation 7** A graph $G$ is a polygonal 2-tree if and only if $G$ has a nice ear decomposition.

In the following lemmas, we present results from the literature that establish polygonal 2-trees as a subclass of 2-connected partial 2-trees, which we formalize in Lemma[10]

**Lemma 8 (Theorem 42 in [2]).** A graph $G$ is a partial 2-tree if and only if every 2-connected component of $G$ is a series-parallel graph.

According to Lemma[8] 2-connected series-parallel graphs and 2-connected partial 2-trees are essentially same.

**Lemma 9 (Lemma 1, Lemma 7 and Theorem 1 in [9]).** A graph $G$ is 2-connected if and only if $G$ has a open ear decomposition in which the first ear is an edge. Further, for a 2-connected series-parallel graph, every open ear decomposition is nested. A graph is series-parallel if and only if it has a nested ear decomposition.

The above lemma implies that every 2-connected partial 2-tree has a nested ear decomposition starting with an edge (first ear is an edge) and vice versa. We strengthen the first part of this result in Lemma[13]
Fig. 1: For the polygonal 2-tree $G$ shown, $(P_0, \ldots, P_{10})$ is a nice ear decomposition of $G$, where $P_0 = (a, b)$, $P_1 = (a, d, c, e, b)$, $P_2 = (a, f, b)$, $P_3 = (c, g, d)$, $P_4 = (c, h, e)$, $P_5 = (b, i, e)$, $P_6 = (a, j, d)$, $P_7 = (a, k, j)$, $P_8 = (a, l, j)$, $P_9 = (d, m, j)$, $P_{10} = (d, n, j)$ are paths in $G$.

2.1 Necessary and Sufficient Conditions

From Propositions 1.7.2 and 12.4.2 in [5], partial 2-trees do not contain a $K_4$-subdivision (as a subgraph). The following lemma presents a few necessary properties of polygonal 2-trees, which are useful in the rest of the paper.

**Lemma 10.** Let $G$ be a polygonal 2-tree. Then,
(a) $G$ is a 2-connected partial 2-tree and $G$ does not contain a $K_4$-subdivision.
(b) Any two induced cycles in $G$ share at most one edge and at most two vertices.
(c) For $u, v \in V(G)$ such that $(u, v) \notin E(G)$, $G - \{u, v\}$ has at most two components.

**Proof.** From Lemma 9, a graph is a 2-connected partial 2-tree if and only if it has a nested ear decomposition. Observe that nice ear decomposition is a restricted version of nested ear decomposition. Therefore, a polygonal 2-tree is a 2-connected partial 2-tree. Recall that partial 2-trees do not contain $K_4$-subdivision as a subgraph. It follows that polygonal 2-trees do not contain $K_4$-subdivision as a subgraph.

We now prove that any two induced cycles in $G$ share at most one edge and at most two vertices. Let $D = (P_0, \ldots, P_k)$ be a nice ear decomposition of $G$. The proof is by induction on the number of ears in $G$. If the number of ears in $D$ is one, the claim is trivially true. If the number of ears in $D$ is at least two, then we remove the internal vertices of $P_k$ from $G$ and let $G'$ be the resultant graph. Let $D' = (P_0, \ldots, P_{k-1})$. As $G'$ is a polygonal 2-tree and $D'$ is a nice ear decomposition of $G'$, inductively $G'$ satisfies (b). Let $u$ and $v$ be the end vertices of $P_k$. For the induced cycle $C = P_k \cup (u, v)$, $C \cap G'$ is $(u, v)$. Therefore $C$ has at most one edge and two vertices in common with the induced cycles in $G'$. Because $V(P_k) \cap V(G') = \{u, v\}$ and $(u, v) \in E(G')$, $C$ is the only induced cycle not in $G'$. Hence, any two induced cycles in $G$ share at most one edge and at most two vertices.

We now prove the last claim of this lemma. The proof is by contradiction. We assume that the removal of vertices $u$ and $v$ from $G$ such that $(u, v) \notin E(G)$ disconnects $G$ into at least three components $G_1, G_2$ and $G_3$. Note that $(u, v)$ is a minimal vertex separator in $G$, because $G$ is 2-connected. It follows that, for each $1 \leq i \leq 3$, there is an induced path $P_i$ between $u$ and $v$ in $G$, such that the internal vertices of $P_i$ are in $G_i$ and $|E(P_i)| \geq 2$. We have induced cycles $C_1 = P_1 \cup P_3$ and $C_2 = P_2 \cup P_3$ that share at least two edges, which contradicts that any two induced cycles in $G$ have at most one edge common.

We now present a sufficient condition for a graph to be a polygonal 2-tree.

**Lemma 11.** If $G$ is a 2-connected partial 2-tree and every two induced cycles in $G$ share at most one edge, then $G$ is a polygonal 2-tree.

**Proof.** On the contrary, assume that $G$ is not a polygonal 2-tree. By Lemma 13, since $G$ is a 2-connected partial 2-tree, $G$ has a nested ear decomposition $D = (P_0, \ldots, P_k)$ such that $P_0$ is an edge and for each
i ≥ 1, |E(P_i)| ≥ 2. Since G is not a polygonal 2-tree, D is not a nice ear decomposition. Therefore, there exists an index i ∈ {1, . . . , k} with the property that, let u and v be the end vertices of P_i, then for every j < i, (u, v) /∈ E(P_j). For every j ≥ i, since |E(P_j)| ≥ 2 and no internal vertex of P_j is in P_1, . . . , P_{i−1}, (u, v) /∈ E(P_j). Thereby (u, v) /∈ E(G). As P_0 ∪ . . . ∪ P_{i−1} is 2-connected, there exist two internally vertex disjoint paths P'_i and P''_i between u and v. Since (u, v) /∈ E(G), P'_i, P''_i and P_i are internally vertex disjoint paths and each of these paths have at least one internal vertex. Due to Lemma 12, for 1 ≤ i ≠ j ≤ 3, there is no path between any internal vertex in P_i and any internal vertex in P_j that excludes the vertices u and v. Now we have two induced cycles P'_i ∪ P_i and P''_i ∪ P_i that share at least two edges. This contradicts the premise of the lemma. Therefore, G is a polygonal 2-tree.

2.2 Computation of Induced Cycles in Polygonal 2-trees

Our algorithm will perform several computations on the induced cycles of a polygonal 2-tree. It is therefore important to obtain the set of induced cycles in a polygonal 2-tree in linear time. We prove this in Theorem 14. This is based on the following two lemmas and a linear-time algorithm for obtaining an open ear decomposition 21.

Lemma 12. Let G be a partial 2-tree and let P_1, P_2 and P_3 be three internally vertex disjoint paths between vertices u and v in G such that (u, v) /∈ E(G). Then G − {u, v} has at least three components.

Proof. Suppose G − {u, v} has at most two components, then without loss of generality there is a path between an internal vertex x in P_1 and an internal vertex y in P_2 without going through any internal vertex in P_3. Then there is a K_4-subdivision on the vertices x, y, u and v in G. It contradicts that a partial 2-tree does not contain a K_4-subdivision. Thus G − {u, v} has at least three components.

Lemma 13. Let G be a 2-connected partial 2-tree. Then there exists a nested ear decomposition (P_0, . . . , P_k) of G, such that P_0 is an edge and for each i ≥ 1, |E(P_i)| ≥ 2.

Proof. From Lemma 9 G has a nested ear decomposition D = (P_0, . . . , P_k) such that P_0 is an edge. Suppose D does not satisfy the given constraint, then we update D as follows, so that the resultant nested ear decomposition satisfies the given constraint. Let P_i be the first path in the sequence D, such that |E(P_i)| = 1, where i ≥ 1. Let P_j be the first path in the sequence D', such that the end vertices of P_j are in P_i, where j < i. Let x and y be the end vertices of P_i. We obtain new paths P'_i and P''_i from P_i and P_j as follows: P'_i is the path between x and y in P_i and P''_i is P'_i ∪ P_i − x, where x is the set of internal vertices in P''_i. We replace P_i with P'_i, delete P_i and add P''_i immediately after P_j. By performing the update steps mentioned above for at most k − 2 times, we obtain a nested ear decomposition that satisfies the desired constraint.

In the lemma below, we show that a nested ear decomposition as in Lemma 13 is a nice ear decomposition for polygonal 2-trees and it can be computed in linear time.

Theorem 14. Let G be a polygonal 2-tree on n vertices. Let D be a nested ear decomposition of G as in Lemma 13 and B be the set of induced cycles in G. Then D is a nice ear decomposition. Further, D and B can be computed in linear time and size(B) is O(n).

Proof. On the contrary, assume that D is not a nice ear decomposition. By Lemma 13 G has a nested ear decomposition D = (P_0, . . . , P_k) such that P_0 is an edge and for each i ≥ 1, |E(P_i)| ≥ 2. Since D is not a nice ear decomposition, there exists an index i ∈ {1, . . . , k} with the property that, let u and v be the end vertices of P_i, and for every j < i, (u, v) /∈ E(P_j). For every j ≥ i, since |E(P_j)| ≥ 2 and no internal vertex of P_j is in P_1, . . . , P_{i−1}, (u, v) /∈ E(P_j). Thereby (u, v) /∈ E(G). As P_0 ∪ . . . ∪ P_{i−1} is 2-connected, there exist two internally vertex disjoint paths P'_i and P''_i between u and v. Since (u, v) /∈ E(G), P'_i, P''_i and P_i are internally vertex disjoint paths. Due to Lemma 12 G − {u, v} has at least three components, which is a contradiction to Lemma 10 (c).

We now prove that a nice ear decomposition of G can be obtained in O(n) time. First obtain an open ear decomposition D’ starting with an edge by using linear-time algorithm in 21. We then apply Lemma 13 on D’. This takes linear time, because the number of ears in D’ is at most n and we spend only a constant
amount of time at each ear. From the first part of this lemma, the resultant ear decomposition is a nice ear decomposition. Also note that \( |E(G)| \leq 2n - 3 \). Thus a nice ear decomposition \((P_0, \ldots, P_k)\) is computed in \( O(n) \) time.

From the nice ear decomposition \( D = (P_0, \ldots, P_k) \) of \( G \), we now present a linear-time procedure to obtain the set of induced cycles in \( G \). Since \( P_0 \) is an edge, \( C_1 = P_0 \cup P_1 \) is an induced cycle in \( G \). For every \( i \geq 2 \), let \( x_i \) and \( y_i \) be the end vertices of \( P_i \), we obtain an induced cycle \( C_i = P_i \cup (x_i, y_i) \) in \( G \). Observe that \( C_1, \ldots, C_k \) are the only induced cycles in \( G \). This can be proved easily by applying induction on the number of ears in \( D \). The number of ears in \( D \) is at most \( n \). Thus the set of induced cycles in \( G \) can be obtained in \( O(n) \). The ears \( P_0, \ldots, P_k \) is a partition of \( E(G) \). Therefore, \( |E(C_0)| + \ldots + |E(C_k)| \leq |E(G)| + n \). Thus size(\( B \)) is \( O(n) \).

\[ \square \]

\section{Structure of Paths, Spanning Trees and MASTs in Polygonal 2-trees}

For the rest of the paper, \( G \) denotes a polygonal 2-tree. In this section we design an iterative procedure to delete a subset of edges from a polygonal 2-tree, so that the graph on the remaining edges is a minimum average stretch spanning tree. This result is shown in Theorem \([21]\).

\textbf{Important Definitions:} We introduce some necessary definitions on polygonal 2-trees. Two induced cycles in \( G \) are \textit{adjacent} if they share an edge. An edge in \( G \) is \textit{internal} if it is part of at least two induced cycles; otherwise it is \textit{external}. An induced cycle in \( G \) is \textit{external} if it has an external edge; otherwise it is \textit{internal}. A fundamental cycle of a spanning tree, created by a non-tree edge is said to be \textit{external} if the associated non-tree edge is external. For a cycle \( C \) in \( G \), the enclosure of \( C \) is defined as \( G[V(C)] \) and is denoted by \( Enc(C) \). A set \( A \subseteq E(G) \) consisting of \( k \) (\( \geq 0 \)) edges is said to be an \textit{iterative} set for \( G \) if the edges in \( A \) can be ordered as \( e_1, \ldots, e_k \) such that \( e_1 \) is external and not a bridge in \( G \), and for each \( 2 \leq i \leq k \), \( e_i \) is external and not a bridge in \( G - \{e_1, \ldots, e_{i-1}\} \). Let \( A \) be an iterative set of edges in \( G \). For every edge \( (u, v) \in A \), both \( u \) and \( v \) are not present in the same 2-connected component in \( G - A \). We define \( \text{bound}(A, G) \) to be the set of external edges in \( G - A \) that are not bridges. For an edge \( e \in \text{bound}(A, G) \), \( G_e \) denotes the 2-connected component in \( G - A \) that has \( e \). The following definition is illustrated in Fig \([2]\).

\textbf{Definition 15.} Let \( A \) be an iterative set of edges in \( G \) and \( e \in \text{bound}(A, G) \). The \textit{support} of \( e \) is defined as \( \{ (u, v) \in A \mid \text{there is a path } P \text{ joining } u \text{ and } v \text{ in } G - A \text{ such that } P \cap G_e = e \} \) and is denoted by \( \text{Support}(e) \). The \textit{cost}(\( e \)) is defined as \( |\text{Support}(e)| \).

![Fig. 2: For the polygonal 2-tree \( G \) shown, let \( A = \{ (a, f), (a, b), (b, p), (c, g) \} \). The edges in \( \text{bound}(A, G) \) are shown in thick. \( \text{Support}((a, d)) = \{ (a, b), (a, f) \} \) and \( \text{Support}((b, i)) = \{ (b, p) \} \). \( \text{cost}((a, d)) = 2 \), \( \text{cost}((b, i)) = 1 \) and for the rest of the edges in \( \text{bound}(A, G) \), cost is zero.](image)
3.1 Structural Properties of Paths

In the following lemmas we present a result on the structure of paths connecting the end points of edges in an iterative set $A$. This is useful in setting up the iterative approach for computing a minimum average stretch spanning tree. We apply the necessary properties of polygonal 2-trees (cf. Lemma 10) and sufficient condition for a graph to be a polygonal 2-tree (cf. Lemma 11) in the proofs of the following lemmas.

Lemma 16. Let $A$ be an iterative set of edges for $G$ and $(u, v) \in A$, $P$ be a path joining $u$ and $v$ in $G - A$, $G'$ be a 2-connected component in $G - A$ that has at least two vertices from $P$, and let $P' = P \cap G'$ be a path with end vertices $x$ and $y$. Then the following are true:

(a) $(x, y) \in E(G')$.

(b) If $P$ is a shortest path, then $P'$ is an edge.

(c) Every 2-connected component in $G - A$ is a polygonal 2-tree.

Proof. To show that $(x, y) \in E(G')$, assume to the contrary that $(x, y) \notin E(G')$. Since $G'$ is 2-connected, there exist two internally vertex disjoint paths $P_1$ and $P_2$ between $x$ and $y$ in $G'$. Since $A$ is an iterative set of edges for $G$ and $(u, v) \in A$, $|\{u, v\} \cap V(G')| \leq 1$. It follows that $P' \subset P$. Then from the cycle $P \cup (u, v)$, we choose a path $P_3$ joining $x$ and $y$, in such a way that $P_3$ is edge disjoint from $P'$. Consequently, none of the internal vertices in $P_3$ are from $G'$. Therefore, $P_1, P_2$ and $P_3$ are internally vertex disjoint paths joining $x$ and $y$ that have at least one internal vertex. By Lemma 12, $G - \{x, y\}$ has at least three components. Then the contrapositive of Lemma 10(c) implies that $G$ is not a polygonal 2-tree. This contradicts that $G$ is a polygonal 2-tree. Thus $(x, y) \in E(G')$.

If $P$ is a shortest path and $P'$ is not an edge, then we can replace $P'$ in $P$ by $(x, y)$ and obtain a path shorter than $P$. Therefore, $P'$ is an edge.

We now prove the third claim of this lemma. Let $H$ be a 2-connected component in $G - A$. From Lemma 10(a), $H$ is a partial 2-tree. Thereby $H$ is a 2-connected partial 2-tree. Since $A$ is an iterative set, the edges in $A$ can be ordered as $e_1, \ldots, e_k$, such that $e_1$ is external and not a bridge in $G$ and for each $2 \leq i \leq k$, $e_i$ is external and not a bridge in $G - \{e_1, \ldots, e_{i-1}\}$. We delete the edges in $A$ from $G$ one by one, in the order $e_1, \ldots, e_k$. Observe that each time, when an edge $e_i$ is deleted, exactly one induced cycle is destroyed and no new induced cycles are created. Also we know that any two induced cycles in $G$ share at most one edge. Consequently, any two induced cycles in $H$ share at most one edge. Therefore, Lemma 11 implies that $H$ is a polygonal 2-tree. □

Lemma 16 is illustrated in Fig 3.

![Fig. 3: For the polygonal 2-tree $G$ shown, let $A = \{(a, b), (a, f), (c, g)\}$. $G_1$, $G_2$ and $G_3$ are the 2-connected components in $G - A$. $G_1$, $G_2$ and $G_3$ are polygonal 2-trees. Let $P = (a, d, c, e, b, f)$ be the shortest path between vertices $a$ and $f$ in $G - A$. $P$ intersects exactly with one edge in the graphs $G_1$, $G_2$ and $G_3.$](image-url)

Lemma 17. Let $A$ be an iterative set of edges for $G$. $(u, v) \in \text{Support}(e)$ if and only if there is a shortest path $P$ joining $u$ and $v$ in $G - A$ and $P$ has $e$. 
We use the following lemma to prove Lemma 17.

**Lemma 18.** Let $P$ be a path with end vertices $u$ and $v$ in $G$. Let $G_1, \ldots, G_r$ be the 2-connected components in $G$ from which $P$ has at least two vertices. For each $1 \leq i \leq r$, let $P_i$ be a shortest path joining the end vertices of $G_i \cap P$. Let $P'$ be the path obtained from $P$ by replacing every $G_i \cap P$ with $P_i$. Then $P'$ is a shortest path joining $u$ and $v$ in $G$.

**Proof.** Assume that there exists a path $P''$ joining $u$ and $v$ in $G$ such that $|E(P'')| < |E(P')|$. For each $i$, let $x_i$ and $y_i$ be the end vertices of $P_i$. The set of edges in $P''$ that are bridges in $G$ are definitely in $P''$. Therefore, there exist an $1 \leq i \leq r$, such that the subpath between $x_i$ and $y_i$ in $P''$ is shorter than $P_i$. This contradicts that $P_i$ is a shortest path joining $x_i$ and $y_i$.

The above lemma holds when $G$ is an arbitrary graph.

**Proof (of Lemma 17).** $(\Rightarrow)$ Let $(u, v) \in \text{Support}(e)$. By the definition of $\text{Support}(e)$, there is a path $P'$ joining $u$ and $v$ in $G - A$ such that $G_e \cap P'$ is internal to $P'$. Let $G_1, \ldots, G_r$ be the 2-connected components in $G - A$ from which $P'$ has at least two vertices. For each $1 \leq i \leq r$, by Lemma 16(b), $P_i = G_i \cap P'$ is a path; let $x_i$ and $y_i$ be the end vertices of $P_i$; due to Lemma 16(a), $(x_i, y_i) \in E(G_i)$. Let $P$ be the path obtained from $P'$ after replacing every $P_i$ by $(x_i, y_i)$. Since $G_e \cap P'$ is internal to $P'$, $P$ has $e$. From Lemma 18, $P$ is a shortest path joining $u$ and $v$ in $G - A$ and $P$ has $e$.

$(\Leftarrow)$ Let $P$ be a shortest path joining $u$ and $v$ in $G - A$ such that $P$ has $e$. Let $G_e$ be a 2-connected component containing $e$ in $G - A$. Since $P$ has $e$, $G_e$ has at least two vertices from $P$. From Lemma 16(b), $G_e \cap P$ is an edge. Further, $G_e \cap P$ is $e$. Thus $(u, v) \in \text{Support}(e)$.

### 3.2 Structural Properties of Spanning Trees

**Lemma 19.** Let $T$ be a spanning tree of $G$ and $e$ be an external edge in $G$ such that $e \in E(T)$. For the spanning tree $T$, let $C_{min}$ be the smallest fundamental cycle containing $e$ and let $C_{max}$ be a largest fundamental cycle containing $e$. Let $e'$ and $e''$ be the non-tree edges associated with $C_{min}$ and $C_{max}$, respectively. Then, (a) $e''$ is an external edge (b) $\text{Enc}(C_{min}) \subseteq \text{Enc}(C_{max})$.

We use the following lemma to prove Lemma 19.

**Lemma 20.** Let $T$ be an arbitrary spanning tree of $G$. Let $C$ be a fundamental cycle of $T$ formed by a non-tree edge $(x, y)$ in $G$. Let $C_1$ be an induced cycle containing $(x, y)$ in $\text{Enc}(C)$ and $C_2$ be another induced cycle containing $(x, y)$ in $G$. Then (a) $V(C) \cap V(C_2) = \{x, y\}$. (b) For vertices $u \in V(C) \setminus \{x, y\}$ and $v \in V(C_2) \setminus \{x, y\}$, any path joining $u$ and $v$ in $G$ goes through $x$ or $y$.

**Proof.** Assume that $V(C) \cap V(C_2)$ has a vertex that is different from $x$ and $y$. In the path consisting of at least two edges from $x$ to $y$ in $C_2$, let $z$ and $z'$ be the first and last vertices from $C$, respectively. From Lemma 10(b), we know that any two induced cycles in a polygonal 2-tree share at most two vertices. Thus $V(C_1) \cap V(C_2) = \{x, y\}$ and $z, z' \notin V(C_1)$. Let $(x', y')$ be an edge in $C_1$ such that $(x', y') \neq (x, y)$. Further, without loss of generality, assume that $x' \neq x$. The graph $C \cup C_2$ is shown in Fig. 14a where the edges in $C$ and $C_2$ other than $(x, y)$ are shown by solid edges and bold edges, respectively. There is a $K_4$-subdivision in $C \cup C_2$ on the vertices $\{x, y, z, z'\}$, because of the following six paths that are internally vertex disjoint: the edge $(x, y)$; the path joining $x'$ and $x$ in $C_1$ without going through $y$; the path joining $x'$ and $y$ in $C_1$ without going through $x$; the path between $z$ and $x$ in $C_2$ without going through $y$; the path between $z$ and $x'$ in $C_2$ without going through $y$. This contradicts that $G$ does not contain a $K_4$-subdivision. Thus $V(C) \cap V(C_2) = \{x, y\}$.

We now prove the second part of the lemma. Let $P$ be a path that joins vertices $u$ and $v$, such that $x, y \notin V(P)$. In the sequence of vertices in $P$ from $u$ to $v$, let $u'$ be the last vertex in $C$ and $v'$ be the first subsequent vertex in $C_2$. Let $P' \subseteq P$ be the path joining the vertices $u'$ and $v'$. From the first part of this lemma, $x$ and $y$ are the only vertices common in $C$ and $C_2$. Thereby $u'$ is different from $v'$. It follows that the edges in $P'$ are disjoint from the edges in $C \cup C_2$. Now, we consider the graph $H = C \cup C_2 \cup P'$. The subgraph $H$ of $G$, shown in Fig. 14b, is a $K_4$-subdivision on the vertices $x, y, u', v'$, because for every two vertices in $\{x, y, u', v'\}$, there is an internally vertex disjoint path. We have a contradiction, as $G$ does not contain a $K_4$-subdivision. Hence the lemma.
Proof (Lemma 19). Assume that \( e'' \) is an internal edge in \( G \). Then \( e'' \) is contained in at least two induced cycles \( C_1 \) and \( C_2 \) in \( G \). Without loss of generality assume that \( C_1 \) is in \( \text{Enc}(C_{\max}) \) and let \( P = C_2 - e'' \) be a path. From Lemma 20(a), \( V(C_2) \cap V(C_{\max}) = \{ x, y \} \). Thus the path between \( x \) and \( y \) in \( T \) and the path \( P \) are internally vertex disjoint. As a consequence, there is an edge \((u, v)\) in \( P \) but not in \( T \); otherwise the tree \( T \) has a cycle. By Lemma 20(b), the fundamental cycle formed by the non-tree edge \((u, v)\) is of larger length than \( C_{\max} \) and also has \( e \). Because \( C_{\max} \) is a maximum length fundamental cycle containing \( e \), this is a contradiction. Therefore, \( e'' \) is an external edge in \( G \).

We now prove the second part of the lemma. Let \( C' \) be the induced cycle containing \( e \) in \( \text{Enc}(C_{\max}) \). Suppose \( C' \) and \( C_{\min} \) are different, then \( e \) is being shared by two induced cycles. This contradicts that \( e \) is an external edge. Thus \( C' = C_{\min} \). Hence \( C_{\min} \subseteq \text{Enc}(C_{\max}) \). \( \square \)

### 3.3 Structural Properties of MASTs

A set \( A \) of edges in \( G \) is referred to as a safe set for \( G \), if \( A \) is an iterative set of edges for \( G \) and a minimum average stretch spanning tree of \( G \) is in \( G - A \).

**Theorem 21.** Let \( A \) be a safe set of edges for \( G \) such that \( \text{bound}(A, G) \neq \emptyset \). Let \( e \) be an edge in \( \text{bound}(A, G) \) for which \( \text{cost}(e) \) is minimum. Then \( A \cup \{ e \} \) is a safe set for \( G \).

**Proof.** For a safe set \( A \), let \( T^* \) be a minimum average stretch spanning tree of \( G \); that is, \( T^* \subseteq G - A \) as \( \text{bound}(A, G) \neq \emptyset \). If \( e \notin E(T^*) \), then we are done. Assume that \( e \in E(T^*) \). Clearly, \( A \cup \{ e \} \) is an iterative set for \( G \). To show that \( A \cup \{ e \} \) is a safe set for \( G \), we use the technique of cut-and-paste to obtain a spanning tree \( T' \) (by deleting the edge \( e \) from \( T^* \) and adding an appropriately chosen edge \( e' \)) and show that \( \text{AvgStr}(T') \leq \text{AvgStr}(T^*) \).

Let \( G_e \) be a 2-connected component in \( G - A \) containing \( e \) and \( G_1, \ldots, G_k \) be the 2-connected components in \( G - A \). For clarity, \( G_e \in \{ G_1, \ldots, G_k \} \). From Lemma 20(c), \( G_e \) is a polygonal 2-tree. For \( 1 \leq i \leq k \), by Lemma 19(a), \( T_i = T^* \cap G_i \) is a spanning tree of \( G_i \). For the spanning tree \( T^* \), let \( C_{\min} \) be the smallest fundamental cycle containing \( e \) in \( G_e \) and let \( C_{\max} \) be a largest fundamental cycle containing \( e \) in \( G_e \). Let \( e', e'' \in E(G_e) \) be the non-tree edges associated with \( C_{\min} \) and \( C_{\max} \), respectively. From Lemma 19 \( e'' \) is an external edge in \( G_e \) and \( \text{Enc}(C_{\min}) \subseteq \text{Enc}(C_{\max}) \). Let \( e' = (x_{\min}, y_{\min}), e'' = (x_{\max}, y_{\max}) \). For a non-tree edge \((u, v)\) in \( T^* \), we use \( P_{uv} \) to denote the path between \( u \) and \( v \) in \( T^* \) and \( C_{uv} \) to denote the fundamental cycle of \( T^* \) formed by \((u, v)\). Let \( X = \{(u, v) \in E(G) \setminus E(T^*) \mid e \in E(P_{uv}), e' \notin \text{Enc}(C_{uv}) \}, Y = \{(u, v) \in E(G) \setminus E(T^*) \mid e \notin E(P_{uv}) \} \). The
The set of non-tree edges in $T^*$ is $X \cup Y \cup \{e\} \cup Z$. Let $T' = T^* + e' - e$. The set of non-tree edges in $T'$ is $X \cup Y \cup Z \cup \{e\}$. To prove the theorem, we prove the following claims.

**Claim 1:** $X \subseteq A$.
**Claim 2:** $\text{Support}(e) \subseteq X$.
**Claim 3:** $\text{Support}(e') \subseteq Y$.
**Claim 4:** $X \subseteq \text{Support}(e)$.
**Claim 5:** For every $(u, v) \in Z$, the path between $u$ and $v$ in $T'$ is in $T^*$.

Assuming that the above five claims are true, we complete the proof of the theorem. We know that $\text{cost}(e) \leq \text{cost}(e'')$. As $e$ and $e''$ are in $G_e$, from the definition of Support, we further know that $\text{Support}(e) \cap \text{Support}(e'') = \emptyset$. Therefore, from Claims 2, 3 and 4, it follows that $|X| \leq |Y|$. Since $e', e \in E(C_{\text{min}})$, $e \in E(T^*)$ and $e' \notin E(T^*)$, the stretch of $e'$ in $T^*$ is equal to the stretch of $e$ in $T'$. From Claim 5, stretch do not change for the edges in $Z$. For all the edges in $X$, stretch increases by $|C_{\text{min}}| - 2$. Further, for all the edges in $Y$, stretch decreases by $|C_{\text{min}}| - 2$. If $|X| < |Y|$, shown in Fig 5b, then $\text{AvgStr}(T') < \text{AvgStr}(T^*)$; it contradicts that $T^*$ is a minimum average stretch spanning tree. Thereby $|X| = |Y|$, shown in Fig 5b, it implies that $\text{AvgStr}(T') = \text{AvgStr}(T^*)$. Since $T^*$ is a minimum average stretch spanning tree, $T'$ is also a minimum average stretch spanning tree. Clearly, $T'$ is in $G - (A \cup \{e\})$. Hence $A \cup \{e\}$ is a safe set for $G$.

![Fig. 5: Dashed and solid edges shown in thick are the edges of $G_e$. Dashed edges are the non-tree edges of $T^*$ and solid edges are the edges of $T'$.](image)

We now prove the five claims.

**Proof of Claim 1:** On the contrary, assume that $(u, v) \in X$ and $(u, v) \notin A$. To arrive at a contradiction, we show that $e$ is an internal edge. Since $(u, v) \in X$, there is a fundamental cycle $C_{uv}$ of $T^*$ formed by the non-tree edge $(u, v)$ containing $e$. As $(u, v) \notin A$, clearly $(u, v)$ is in $G - A$. Further, $P_{uv}$ is in $G - A$, because $T^* \subseteq G - A$. So we know that $C_{uv}$ is in $G - A$. If $C_{uv}$ is not in $G_e$, then $G_e \cup C_{uv}$ becomes a 2-connected component in $G - A$, because $G_e$ is in $G - A$. $C_{uv}$ is in $G - A$, and $e$ is both in $G_e$ and $C_{uv}$. But, we know that $G_e$ is a maximal 2-connected subgraph (2-connected component), thereby $C_{uv}$ is in $G_e$. Clearly, $C_{uv}$ and $C_{\text{min}}$ are not edge disjoint cycles. If $\text{Enc}(C_{\text{min}}) \subseteq \text{Enc}(C_{uv})$, then either $(u, v) \in Y$ or $(u, v) = e'$, which contradicts the fact that $(u, v) \in X$. Also, $\text{Enc}(C_{uv})$ is not contained in $\text{Enc}(C_{\text{min}})$, because $C_{\text{min}}$ is a minimum length induced cycle containing $e$. Therefore, both $C_{\text{min}}$ and $C_{uv}$ are not contained in each other. Thus, $e$ is an internal edge in $G - A$. This is a contradiction, as we know that $e$ is external.
Lemma 17, there is a shortest path

Proof of Claim 2: Let \((u, v) \in X\). In order to prove that \((u, v) \in X\), we show the following: (a) \((u, v) \notin E(T^*)\), (b) \(P_{uv}\) has \(e\) and (c) \(e'\) is not in \(Enc(C_{uv})\).

By the definition of \(Support(e)\), \((u, v) \in A\). As \(T^* \subset G - A\), it follows that \((u, v) \notin E(T^*)\). By Lemma 17 there is a shortest path \(P\) joining \(u\) and \(v\) in \(G - A\) and \(P\) has \(e\). Let \(G_i^1, \ldots, G_i^r\) be the 2-connected components in \(G - A\) containing at least two vertices from \(P\). Due to Lemma 10(b), for each \(1 \leq i \leq r\), \(P \cap G_i^r\) is an edge, say \((x_i, y_i)\). Thus \(P \cap \max\) is \(e\). Further, \(P\) contains at most one vertex from \(e'\), because \(e' \in E(G_e)\). The set of edges in \(P\) that are cut-edges in \(G - A\) are present in \(T^*\). Due to Lemma 6(a), replacing every edge \((x_i, y_i)\) in \(P\) by the path between \(x_i\) and \(y_i\) in \(T^*\), \(P_{uv}\) is obtained. Since \(P \cap \max\) is \(e\) and \(e\) is in \(T^*\), it implies that \(P_{uv}\) has \(e\). Thus \(P_{uv}\) has \(e\) and \(e'\) is not in \(Enc(C_{uv})\).

Proof of Claim 3: Let \((u, v) \in X\). In order to prove that \((u, v) \in Y\), we show the following: (a) \((u, v) \notin E(T^*)\), (b) \(P_{uv}\) has \(e\) and (c) \(e'\) is in \(Enc(C_{uv})\).

Because \((u, v) \in A\) and \(T^* \subset G - A\), we have \((u, v) \notin E(T^*)\). As \(e, e'' \in E(G_e)\), due to Lemma 17, there is a shortest path \(P\) joining \(u\) and \(v\) in \(G - A\) and \(P\) has \(e''\). Let \(G_i^1, \ldots, G_i^r\) be the 2-connected components in \(G - A\) such that for each \(1 \leq i \leq r\), \(P \cap G_i^r\) is an edge, say \((x_i, y_i)\), due to Lemma 10(b). By Lemma 6(a), we replace every edge \((x_i, y_i)\) in \(P\) by the path between \(x_i\) and \(y_i\) in \(T^*\) and obtain the tree path \(P_{uv}\). Note that \(P \cap \max\) is \(e''\), \(e'' = (x_{\text{max}}, y_{\text{max}})\), and \(e'' \in P\) got replaced with the path between \(x_{\text{max}}\) and \(y_{\text{max}}\) in \(T^*\). Also, we know that the path between \(x_{\text{max}}\) and \(y_{\text{max}}\) in \(T^*\) has \(e\). Further by Lemma 19(b), \(e'\) is in \(Enc(C_{uv})\). These observations imply that \(P_{uv}\) has \(e\) and \(Enc(C_{uv})\) contains \(e'\).

Proof of Claim 4: Let \((u, v) \in X\). By Claim 1, clearly \((u, v) \in A\). Lemma 6(b) implies that \(P_{uv} \cap \max\) is a path. Let \(P' = P_{uv} \cap \max\) be a path and let \(x\) and \(y\) be the end vertices of \(P'\). If \(P'\) is an edge, shown in Fig 5c then the claim holds. On the contrary assume that \(P'\) has at least two edges. By Lemma 10(a), \((x, y) \in E(G)\). Further, \((x, y) \notin E(T^*)\) as it would then form a cycle in the tree. If \(x_{\text{min}}, y_{\text{min}} \in V(P')\), shown in Fig 6a and Fig 6b then \((u, v)\) must be in \(Y\). As we know that \((u, v) \in X\), the path \(P'\) is strictly contained in the path joining the vertices \(x_{\text{min}}\) and \(y_{\text{min}}\) in \(T^*\). Then the fundamental cycle of \(T\) formed by \((x, y)\) is of lesser length than the length of \(C_{\text{min}}\), shown in Fig 6a; a contradiction because \(C_{\text{min}}\) is a minimum length fundamental cycle in \(G_e\) containing \(e\). Therefore, \(P_{uv} \cap \max\) is \(e\). Thus \((u, v) \in Support(e)\).

Proof of Claim 5: Let \((u, v) \in Z\). By the definition of \(Z\), clearly \(e \notin \max\). It implies that \(e' \notin Enc(C_{uv})\) as the path between the end vertices of \(e'\) in \(T^*\) has \(e\). Therefore \(P_{uv}\) has at most one end vertex from \(e\) and \(e'\). Since the symmetric difference of \(E(T^*)\) and \(E(T')\) is \(\{e, e'\}\), the path \(P_{uv}\) in \(T^*\) remains same in \(T'\).

Hence the theorem.

We now show the termination condition for applying Theorem 21

Lemma 22. Let \(A\) be a safe set of edges for \(G\) such that \(bound(A, G) = \emptyset\). Then \(G - A\) is a minimum average stretch spanning tree of \(G\).

Proof. Since \(A\) is a safe set for \(G\), a minimum average stretch spanning tree is contained in \(G - A\). Since \(bound(A, G) = \emptyset\), \(G - A\) is acyclic. Therefore, \(G - A\) is a minimum average stretch spanning tree of \(G\).

4 Computing MAST in Polygonal 2-trees

In order to obtain a minimum average stretch spanning tree efficiently, we need to efficiently find an edge in \(bound(A, G)\) with minimum cost in every iteration, where \(A\) is a safe set for \(G\). In this section, we present necessary data-structures, so that a minimum average stretch spanning tree in polygonal 2-trees on \(n\) vertices can be computed in \(O(n \log n)\) time. This is shown in Algorithm 2. For each edge \(e \in bound(A, G)\), we show in Lemma 23 how to compute \(cost(e)\) efficiently.

Notation. Let \(Q\) be a min-heap that supports the following operations: \(Q.insert(x)\) inserts an arbitrary element \(x\) into \(Q\), \(Q.extract-min()\) extracts the minimum element from \(Q\), \(Q.decrease-key(x, k)\) decreases the key value of \(x\) to \(k\) in \(Q\), \(Q.delete(x)\) deletes an arbitrary element \(x\) from \(Q\). \(Q.delete(x)\) can be implemented by calling \(Q.decrease-key(x, -\infty)\) followed by \(Q.extract-min()\). For a set \(A\) of safe
The min heap

For an edge \( e \) the count of unprocessed cycles are updated. Thus L2 holds. For each edge \( e \) in \( \text{bound}(A, G) \), \( \text{unpCount}[e] \) is one. Therefore, \( \text{Cycles}[e] \setminus \text{pCycles}[e] \) gives the unique induced cycle in \( G - A \) containing \( e \), thereby L3 holds.

**Proof.** An edge gets inserted into \( Q \) only when it is in a unique induced cycle of \( G - A \). Further, all the bridges in \( G - A \) are getting deleted in line 11. Thus L1 holds. In lines 8 and 9, processed induced cycles and the count of unprocessed cycles are updated. Thus L2 holds. For each edge \( e \) in \( \text{bound}(A, G) \), \( \text{unpCount}[e] \) is one. Therefore, \( \text{Cycles}[e] \setminus \text{pCycles}[e] \) gives the unique induced cycle in \( G - A \) containing \( e \), thereby L3 holds. □
Lemma 24. Let \( a \) be a subpath in \( E(x) \) the shortest path between \( P \). We use the following lemma to prove Lemma 23. For a path the destructive edge of \( C \) u ungenerality the path is an induced cycle. \( P \) replacing the path \( a \rightarrow i \) in the heap \( Q \) this step in Corollary 25 using Lemma 23.

Let \( \text{Lemma 23.} \)

\[ 4.1 \text{ Cost Updation} \]

During the execution of our algorithm, for each edge \( e \) in \( G - A \), such that \( e \) is external and not a bridge, we need to compute \( \text{cost}(e) \) efficiently. This is done in Algorithm 2 in line 7. We prove the correctness of this step in Corollary 23 using Lemma 23.

Let \( A_i \subseteq E(G) \) denote the set of safe edges in \( G \) at the end of \( i^{th} \) iteration. Let \( e \) be an edge extracted from the heap \( Q \) in \( i^{th} \) iteration and \( C \) be the unique induced cycle containing \( e \) in \( G - A_{i-1} \). That is, \( C \) is a cycle in \( G - A_{i-1} \) and \( C \) is not a cycle in \( G - A_i \) as \( e \) is added to \( A \) in iteration \( i \). Then we say that \( C \) is processed in iteration \( i \) and \( e \) is the destructive edge for \( C \).

Lemma 23. Let \( e \in \text{bound}(A_j, G) \), where \( 1 \leq j < m - n + 1 \). Let \( C \) be the unique external induced cycle in \( G - A_j \) containing \( e \) and \( C_1, \ldots, C_k \) be the other induced cycles in \( G \) containing \( e \). For \( 1 \leq i \leq k \), let \( e_i \) be the destructive edge of \( C_i \). Then \( \text{Support}(e) = \text{Support}(e_1) \cup \ldots \cup \text{Support}(e_k) \).\( \{e_1, \ldots, e_k\} \).

We use the following lemma to prove Lemma 23. For a path \( P \) and for vertices \( x, y \in V(P) \), \( P(x, y) \) denotes a subpath in \( P \) with end vertices \( x \) and \( y \). For an edge \( (x, y) \) in \( G \), if \( (x, y) \) is external, then there is a unique shortest path between \( x \) and \( y \) in \( G - (x, y) \).

Lemma 24. Let \( A \) be a safe set for \( G \) and \( P \) be a path with end vertices \( u \) and \( v \) in \( G - A \). Let \( (a, b) \) an edge in \( P \) such that \( (a, b) \in \text{bound}(A, G) \). Let \( G' = G - (A \cup \{(a, b)\}) \) and \( P' \) be the path obtained from \( P \) by replacing \( (a, b) \) with the shortest path between \( a \) and \( b \) in \( G' \). \( P \) is a shortest path in \( G - A \) if only if \( P' \) is a shortest path in \( G' \).

Proof. (\( \Rightarrow \)) Assume that \( P' \) is not a shortest path joining \( u \) and \( v \) in \( G' \). Then there exist a path \( P'' \) joining \( u \) and \( v \) in \( G' \) such that \( |E(P'')| < |E(P')| \). In the path \( P \) from \( u \) to \( v \), without loss of generality, assume that \( a \) appears before \( b \). We choose \( a' = V(P(u, a)) \) and \( b' = V(P(b, v)) \) in such a way that \( P(a', b') \cup P''(a', b') \) is an induced cycle.

Let \( C' = P(a', b') \cup P''(a', b') \) and \( C \) be an external induced cycle containing \( (a, b) \) in \( G - A \). Thereby \( (a, b) \in E(C) \cap E(C') \). Recall that any two induced cycles in a polygonal 2-tree share at most one edge. Therefore, either \( E(C) \cap E(C') = \{(a, b)\} \) or \( C = C' \). If \( E(C) \cap E(C') = \{(a, b)\} \), then \( (a, b) \) is an internal edge in \( G - A \), but we know that \( (a, b) \) is an external edge in \( G - A \). Consider the other case where \( C = C' \). Then \( a' = a, b' = b, P(a', b') = (a, b) \) and \( P''(a', b') = C = (a, b) \). Because \( P'(a, b) \) is \( P''(a, b) \), by not loss of generality the path \( P''(u, a) \) consisting of lesser number of edges than \( P'(u, a) \). As \( P(u, a) \) is \( P''(u, a) \), replacing the path \( P(u, a) \) in \( P \) with \( P''(u, a) \) leads to a path shorter than \( P \) in \( G \), which contradicts that \( P \) is a shortest path.

(\( \Leftarrow \)) Assume that \( P \) is not a shortest path joining \( u \) and \( v \) in \( G - A \). Then there exist a path \( P'' \) joining \( u \) and \( v \) in \( G - A \) such that \( |E(P'')| < |E(P)| \). Consider the case where \( a, b \in V(P'') \). \( P'' \) is a disjoint

Algorithm 2: An algorithm to find an MAST of a polygonal 2-tree \( G \)

1 Perform the steps described in Initialization ;
2 while \( Q \neq \emptyset \) do
3 \( e \leftarrow Q.\text{extract-min}() \);
4 \( A \leftarrow A \cup \{e\} \);
5 \( C \leftarrow C.\text{Cycles}[e] \setminus p.\text{Cycles}[e] \);
6 for each edge \( \hat{e} \in E(C) \setminus \{e\} \) do
7 \( c[\hat{e}] \leftarrow c[\hat{e}] + c[e] + 1 \);
8 \( p.\text{Cycles}[\hat{e}] \leftarrow p.\text{Cycles}[\hat{e}] \cup C \);
9 \( \text{unpCount}[\hat{e}] \leftarrow \text{unpCount}[\hat{e}] - 1 \);
10 if \( \text{unpCount}[\hat{e}] = 1 \) then \( Q.\text{insert}(\hat{e}, c[\hat{e}]) \);
11 if \( \text{unpCount}[\hat{e}] = 0 \) then \( Q.\text{delete}(\hat{e}) \);
12 Return \( G - A \);
union of $P''(u, a)$, $P''(a, b)$ and $P''(b, v)$. Similarly $P$ is a disjoint union of $P(u, a)$, $P(a, b)$ and $P(b, v)$. Since $P(a, b) = P''(a, b) = e$, without loss of generality the path $P''(u, a)$ consisting of lesser number of edges than $P(u, a)$. Replacing the path $P''(u, a)$ in $P'$ with $P''(u, a)$ leads to a path shorter than $P'$ in $G'$, which contradicts that $P'$ is a shortest path. Consider the other case where at most one vertex from $\{a, b\}$ is in $P''$. Observe that $|E(P'')| < |E(P)| < |E(P')|$ and $P''$ is in $G'$. Consequently, $P''$ is shorter than $P'$ in $G'$. This contradicts that $P'$ is a shortest path joining $u$ and $v$ in $G'$. □

**Proof** (of Lemma 23). For $1 \leq i \leq k$, let $f(i) + 1$ be the iteration number in which $C_i$ is processed.

($\Leftarrow$) Let $(u, v) \in \text{Support}(e_i)$ for some $1 \leq i \leq k$. Then by Lemma 17 there is a shortest path $P$ joining $u$ and $v$ in $G - A_{f(i)}$ and $P$ has $e_i$. From the premise, $e_i$ gets added to $A$ in the iteration $f(i) + 1$. Thereby $e_i \in A_{f(i)+1}$. It results that $e_i$ is not in $G - A_{f(i)+1}$. Consider the path $P' = (P - e_i) \cup (C_i - e_i)$ in $G - A_{f(i)+1}$. As $e_i$ is exterior in $G - A_{f(i)+1}$, $C_i - e_i$ is a shortest path between the end vertices of $e_i$ in $G - A_{f(i)+1}$. Also, $C_i - e_i$ has $e$. Thus $P'$ has $e$. By forward direction of Lemma 24 $P'$ is a shortest path joining $u$ and $v$ in $G - A_{f(i)+1}$. Note that $e$ is in $G - A_j$. By forward direction of Lemma 24 it follows that there is a shortest path $P_j$ joining $u$ and $v$ in $G - A_j$ and $P_j$ has $e$. Thus $(u, v) \in \text{Support}(e_i)$. Observe that $e_i \in \text{Enc}(P_j \cup (u, v))$. Further, the path between the end vertices of $e_i$ in $P_j$ is a shortest path containing $e$ in $G - A_j$. Therefore, we have $e_i \in \text{Support}(e)$.

($\Rightarrow$) Let $(u, v) \in \text{Support}(e)$. Then by Lemma 17 there is a shortest path $P'$ joining $u$ and $v$ in $G - A_j$ and $P'$ has $e$. $\text{Enc}(P' \cup (u, v))$ contains $e_i$ for some $1 \leq i \leq k$. Consider the case $(u, v) \neq e_i$. Let $x_i$ and $y_i$ be the end vertices of $e_i$. We replace the path between $x_i$ and $y_i$ by the edge $e_i$ and $P$ be the resultant path consisting of $e_i$. From backward direction of Lemma 24 $P$ is a shortest path joining $u$ and $v$ in $G - A_{f(i)}$, and $P$ has $e_i$. Thereby $(u, v) \in \text{Support}(e_i)$. As a result, $(u, v) \in \text{Support}(e_i) \cup \{e\}$ for some $1 \leq i \leq k$. □

**Lemma 23** is illustrated in Fig 7.

![Fig 7](image_url)

**Fig. 7:** For the polygonal 2-tree $G$ shown, dashed edges are the edges in $A$, solid edges are the edges of $G - A$ and solid edges shown in thick are the edges of $G_e$. $\text{Support}(e_1) = \{(j, k)\}$, $\text{Support}(e_2) = \{(l, f)\}$, $\text{Support}(e_3) = \{(m, p)\}$, $\text{Support}(e) = \{(j, k), (l, f), (m, p), e_1, e_2, e_3\}$

**Corollary 25.** Let $e, e_1, \ldots, e_k$ be the edges as mentioned in Lemma 23. Then $\text{cost}(e) = \text{cost}(e_1) + \ldots + \text{cost}(e_k)$.

The algorithm terminates when $Q$ becomes $\emptyset$, that is, $\text{bound}(A, G) = \emptyset$. Then by Lemma 22 $G - A$ is a minimum average stretch spanning tree of $G$.

**Lemma 26.** For a polygonal 2-tree $G$ on $n$ vertices, **Algorithm** 3 takes $O(n \log n)$ time.
Proof. The set of induced cycles in $G$ can be obtained in linear time (cf. Theorem 1.9.1), thereby line 1 takes linear time. As the size of induced cycles in $G$ is $O(n)$ (Theorem 14), line 5 and lines 7-9 contribute $O(n)$ towards the run time of the algorithm. Also every edge in $G$ gets inserted into the heap $Q$ and gets deleted from $Q$ only once and $|E(G)| \leq 2n - 3$. It takes $O((n log n)$ time for the operations $\text{insert}()$, $\text{delete}()$ and $\text{extract-min}()$. Thus Algorithm 2 takes $O(n log n)$ time.

This concludes the presentation of our main result, namely Theorem 2.

5 Characterization of Polygonal 2-trees

The main result in this section is the characterization of polygonal 2-trees using cycle basis, which is presented in Theorem 31. Also, we show that there is a unique minimum cycle basis in polygonal 2-trees, which can be computed in linear time.

A graph is Eulerian if the degree of every vertex is even. The cycle space of $G$ is a set of Eulerian subgraphs of $G$. For the cycles $C_1, \ldots, C_k$, the graph $C_1 \oplus \ldots \oplus C_k$ consists only the edges that appear odd number of times in $C_1, \ldots, C_k$. A minimal set $B$ of cycles is a cycle basis of $G$, if every cycle in $G$ can be expressed as exclusive-or of a subset of cycles in $B$. A minimum cycle basis (MCB) of $G$ is a cycle basis that minimizes the sum of the lengths of the cycles in the cycle basis. The cardinality of a cycle basis is $m - n + 1$. [9]

Planar graphs and Halin graphs are characterized based on a cycle basis. A cycle basis is said to be planar if every edge in the graph appears in at most two cycles in the cycle basis. A graph is planar if and only if it has a planar basis [13]. A 3-connected planar graph is Halin if and only if it has a planar basis and every cycle in the planar basis has an external edge [23]. Halin graphs that are not necklaces have a unique minimum cycle basis [22]. Also, outerplanar graphs have a unique minimum cycle basis [16].

Lemma 27. (Proposition 1.9.1 in [3]) The induced cycles in an arbitrary graph $G$ generate its entire cycle space.

Lemma 28. The number of induced cycles in $G$ is $m - n + 1$.

Proof. We apply induction on the number of internal edges in $G$. Let $E_{in}(G)$ denote the set of internal edges in $G$. If $|E_{in}(G)| = 0$, then $G$ has one induced cycle and $m - n + 1$ is one. For the induction step, let $|E_{in}(G)| > 0$. We decompose $G$ into polygonal 2-trees $G_1, \ldots, G_k$ such that $G_1 \sqcup \ldots \sqcup G_k = G$ and $G_1 \cap \ldots \cap G_k$ is an edge in $G$, where $k \geq 2$. Let $m_i = |E(G_i)|$ and $n_i = |V(G_i)|$. For every $1 \leq i \leq k$, $|E_{in}(G_i)| < |E_{in}(G)|$ as one internal edge of $G$ has become external in $G_i$. By induction hypothesis, for every $1 \leq i \leq k$, the number of induced cycles in $G_i$ is $m_i - n_i + 1$. Observe that the set of induced cycles in $G$ is equal to the disjoint union of the set of induced cycles in $G_1, \ldots, G_k$. Further, we know that $m = m_1 + \ldots + m_k - k + 1$ and $n = n_1 + \ldots + n_k - 2k + 2$. Consequently, we can see that the number of induced cycles in $G$ is $m - n + 1$.

Lemma 29. For an arbitrary 2-connected partial 2-tree $G$, if the set of induced cycles in $G$ is a cycle basis, then $G$ is a polygonal 2-tree.

Proof. Assume that $G$ is not a polygonal 2-tree. Then by Lemma 10 there exist two induced cycles $C_1$ and $C_2$ in $G$ such that $|E(C_1) \cap E(C_2)| \geq 2$. Let $C_3 = C_1 \oplus C_2$. Since $C_1$ and $C_2$ are induced cycles, clearly $C_1 \cap C_2$ is a path and $C_3$ is a cycle. Let $P$ be the maximal common path in $C_1$ and $C_2$. Let $P_1$ and $P_2$ be the maximal private paths in $C_1$ and $C_2$, respectively.

Consider the case when $C_3$ is an induced cycle. The set $\{C_1, C_2, C_3\}$ do not be a part of a cycle basis. It contradicts that the set of induced cycles in $G$ is a cycle basis.

Consider the other case when $C_3$ is not an induced cycle. Let $C_1', \ldots, C_k'$ be the set of induced cycles in $Enc(C_3)$. Since $C_1$ and $C_2$ are induced cycles and $C_3$ is not an induced cycle, there exist a chord $e$ in $C_3$ such that, one end vertex of $e$ is in $P_1$ and the other end vertex of $e$ is in $P_2$. Note that at least one induced cycle in $Enc(C)$ has $e$, where as $C_1$ and $C_2$ do not have $e$. It follows that $\{C_1, C_2\}$ is different from $\{C_1', \ldots, C_k'\}$. We can express $C_3$ as $C_1 \oplus C_2$ as well as $C_1' \oplus \ldots \oplus C_k'$. Therefore, $\{C_1, C_2\} \cup \{C_1', \ldots, C_k'\}$ do not be part of a cycle basis. It is a contradiction, because we know that the set of induced cycles in $G$ is a cycle basis. Therefore, our assumption is incorrect and hence $G$ is a polygonal 2-tree.
Theorem 30. The set of induced cycles in $G$ is a unique minimum cycle basis in $G$.

Proof. Recall that the cardinality of a cycle basis is $m - n + 1$. Therefore, from Lemma 27 and Lemma 28 it follows that induced cycles in $G$ is a cycle basis. Assume that $B$ is a minimum cycle basis of $G$ such that $B$ contains at least one non-induced cycle. Let $C$ be a smallest non-induced cycle in $B$ and $C_1, \ldots , C_k$ be the set of induced cycles in $Enc(C)$. Observe that $C_1 \oplus \ldots \oplus C_k = C$. Clearly, there exists $1 \leq i \leq k$ such that $C_i \notin B$ as $B$ is a cycle basis. We replace $C$ with $C_i$ and obtain a cycle basis such that its size is strictly less than the size of $B$ as $|E(C)| > |E(C_i)|$. We have got a contradiction, because $B$ is a minimum cycle basis. Therefore, the set of induced cycles in $G$ is a unique minimum cycle basis of $G$.

The following theorem follows from Lemma 29 and Theorem 30.

Theorem 31. A graph $G$ is a polygonal 2-tree if and only if $G$ is a 2-connected partial 2-tree and the set of induced cycles in $G$ is a cycle basis.

As the set of induced cycles in polygonal 2-trees is a minimum cycle basis, Theorem 14 computes a minimum cycle basis in polygonal 2-trees in linear time.

References

1. N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. SIAM J. of Comput., 24:78–100, 1995.
2. H. L. Bodlaender. A partial $k$-arboretum of graphs with bounded treewidth. Theoretical Computer Science, 209(1-2):1–45, 1998.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms (3rd ed.). MIT Press, 2009.
4. N. Deo, G. Prabhun, and M. S. Krishnamoorthy. Algorithms for generating fundamental cycles in a graph. ACM Transactions on Math. Software, 8:26–42, 1982.
5. R. Diestel. Graph Theory, fourth edition, 2010.
6. M. Ducharme, G. Labelle, C. Lamathe, and P. Leroux. A classification of outerplanar $k$-gonal 2-trees. In 19th Intern. conf. on FPSAC, 2007. Appeared as Poster.
7. Y. Emek. $k$-outerplanar graphs, planar duality, and low stretch spanning trees. Algorithmica, 61(1):141–160, 2011.
8. Y. Emek and D. Peleg. A tight upper bound on the probabilistic embedding of series-parallel graphs. SIAM J. Discrete Math., 23(4):1827–1841, 2009.
9. D. Eppstein. Parallel recognition of series-parallel graphs. Inf. Comput., 98(1):41–55, 1992.
10. T. Fowler, I. Gessel, G. Labelle, and P. Leroux. The specification of 2-trees. Advances in Applied Mathematics, 28(2):145 – 168, 2002.
11. G. Galbiati, R. Rizzi, and E. Amaldi. On the approximability of the minimum strictly fundamental cycle basis problem. Discrete Appl. Math, 159(4):187–200, 2011.
12. E. Hubicka and M. Syslo. Minimal bases of cycles of a graph. In Recent advances in graph theory, second Czech symposium in graph theory, second Czech symposium in graph theory, Prague, pages 283–293, 1975.
13. T. Kavitha, C. Lieben, K. Mehlhorn, D. Michael, R. Rizzi, T. Ueckerdt, and K. A. Zweig. Cycle bases in graphs characterization, algorithmization, complexity, and applications. Computer Science Review, 3(4):199–243, 2009.
14. K. M. Koh and C. P. Teo. Chromaticity of series-parallel graphs. Discrete Mathematics, 154(1-3):289–295, 1996.
15. G. Labelle, C. Lamathe, and P. Leroux. Labelled and unlabelled enumeration of $k$-gonal 2-trees. J. Comb. Theory, Ser. A, 106(2):193–219, 2004.
16. J. Leydold and P. F. Stadler. Minimal cycle bases of outerplanar graphs. Elec. J. Comb., pages 209–222, 1998.
17. C. Lieben and G. Wunsch. The zoo of tree spanner problems. Discrete Appl. Math., 156:569–587, 2008.
18. S. MacLane. A combinatorial condition for planar graphs. Fund. Math. 28, pages 22–32, 1937.
19. B. Oomooni and Y.-H. Peng. Chromatic equivalence classes of certain generalized polygon trees, iii. Discrete Mathematics, 271(1-3):223–234, 2003.
20. Y.-H. Peng, C. H. C. Little, K. L. Teo, and H. Wang. Chromatic equivalence classes of certain generalized polygon trees. Discrete Mathematics, 172(1-3):103–114, 1997.
21. V. Ramachandran. Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity. Morgan-Kaufmann, 1992.
22. P. F. Stadler. Minimum cycle bases of halin graphs. J. Graph Theory, 43:150–155, 2000.
23. M. M. Syslo and A. Prokurowski. On halin graphs. In Graph theory, Proc. Conf., Lagon/Polf., volume 1018 of Lecture Notes Math., pages 248–256, New York, Springer, 1983.
24. D. B. West. Introduction to graph theory - second edition. Prentice Hall, 2001.