Tree 3-spanners of diameter at most 5

Ioannis Papoutsakis

November 26, 2004

Abstract

A subgraph \( T \) of a graph \( G \) is a tree \( t \)-spanner of \( G \) if and only if \( T \) is a tree and for every pair of vertices in \( G \) their distance in \( T \) is at most \( t \) times their distance in \( G \), where \( t \) is called a stretch factor of \( T \). An efficient algorithm to determine whether a graph admits a tree 2-spanner has been developed, while for each \( t \geq 4 \) the problem to determine whether a graph admits a tree \( t \)-spanner has been proven to be NP-complete. Although it is not known whether it is tractable to decide graphs that admit a tree 3-spanner, an efficient algorithm to determine whether a graph admits a tree 3-spanner of diameter at most 5 is presented.

1 Introduction

There are applications of spanners in a variety of areas, such as distributed computing \cite{2, 14}, communication networks \cite{12, 13}, motion planning and robotics \cite{1, 7} and phylogenetic analysis \cite{3}. Furthermore, spanners are used in embedding finite metric spaces in graphs approximately \cite{15}.

On one hand, in \cite{4, 6, 5} an efficient algorithm to decide tree 2-spanner admissible graphs is presented, where a method to construct all the tree 2-spanners of a graph is also given. On the other hand, in \cite{6, 5} it is proven that for each \( t \geq 4 \) the problem to decide graphs that admit a tree \( t \)-spanner is an NP-complete problem. The complexity status of the tree 3-spanner problem is unresolved. In \cite{9} it is shown that the problem to determine whether a graph admits a tree \( t \)-spanner of diameter at most \( t + 1 \) is tractable, when \( t \leq 3 \), while it is an NP-complete problem, when \( t \geq 4 \). The remainder of the paper is on tree 3-spanners, while, in general, terminology of \cite{17} is used.

Definition 1 A graph \( T \) is a tree 3-spanner of a graph \( G \) if and only if \( T \) is a subgraph of \( G \) that is a tree and, for every pair \( u \) and \( v \) of vertices of \( G \), if \( u \) and \( v \) are at distance \( d \) from each other in \( G \), then \( u \) and \( v \) are at distance at most \( 3 \cdot d \) from each other in \( T \).

Note that in order to check that a spanning tree of a graph \( G \) is a tree 3-spanner of \( G \), it suffices to examine pairs of adjacent in \( G \) vertices. Toward
an efficient algorithm for the tree 3-spanner problem, graphs that admit a tree 3-spanner of diameter at most 5 are studied. To focus on trees that have bounded diameter, the concept of a \( k \)-center is introduced. Note that a path of even length has a central vertex, while a path of odd length has a central edge.

**Definition 2** A \( k \)-center \( K \) of a graph \( G \) is a subgraph of \( G \) consisting exactly of either a vertex when \( k \) is even, or a pair of adjacent in \( K \) vertices when \( k \) is odd, such that for all \( u \) in \( G \), \( d_G(K, u) \leq \lfloor \frac{k}{2} \rfloor \).

For any \( k \)-center \( K \), it holds that \( |K| = |E(K)| = k \mod 2 \). Clearly, when \( k > 0 \), a tree has a \( k \)-center if and only if it is of diameter at most \( k \).

Assume that a graph \( G \) contains a \( k \)-center \( K \). Any Breadth-First Search tree \( T \) of \( G \) starting from \( K \) has the property that \( d_T(K, u) = d_G(K, u) \), for every vertex \( u \) of \( G \). Therefore, since \( K \) is a \( k \)-center of \( G \), \( T \) is a tree \( k \)-spanner of \( G \); observe that the distance in \( T \) between any pair of vertices \( u \) and \( v \) is at most equal to the distance from \( u \) to \( K \) plus \( |K| \) plus the distance from \( K \) to \( v \). The difficulty of the tree 3-spanner problem appears when graphs that do not contain any 3-center are examined. A frequently used lemma follows.

**Lemma 1** Let \( G \) be a graph and \( T \) a tree 3-spanner of \( G \). If \( u \) is in a \( p \)-, \( q \)-path of \( T \) and \( p \), \( q \) are not neighbors of \( u \) in \( T \), then every \( p \)-, \( q \)-path \( P' \) of \( G \) contains a vertex which is a neighbor of \( u \) in \( T \).

**Proof.** Consider the components of \( T \setminus u \). Obviously, vertices \( p \) and \( q \) belong to different such components. Therefore, for any \( p \)-, \( q \)-path \( P' \) of \( G \) there is an edge \( uu' \) in \( P' \) such that \( w \) is in a different component than \( w' \) is. Since all the tree paths connecting vertices of different such components pass through \( u \), it holds that \( d_T(w, w') = d_T(w, u) + d_T(u, w') \). But the tree distance between \( w \) and \( u' \) can be at most 3, therefore at least one of \( w \) or \( w' \) is a neighbor of \( u \) in \( T \).

Tree 3-spanners have been studied for various families of graphs. If a connected graph is a cograph or a split graph or the complement of a bipartite graph, then it admits a tree 3-spanner \([5]\). Also, all convex bipartite graphs have a tree 3-spanner, which can be constructed in linear time \([16]\). Efficient algorithms to recognize graphs that admit a tree 3-spanner have been developed for interval, permutation and regular bipartite graphs \([11]\), \([12]\), planar graphs \([8]\), directed path graphs \([10]\), 2-trees \([?]\), very strongly chordal graphs (containing all interval graphs), 1-split graphs (containing all split graphs) and chordal graphs of diameter at most 2 \([?]\).

### 2 The algorithm

**Input.** A graph \( G \) and a pair of adjacent in \( G \) vertices \( u \), \( v \).

**Question.** Does \( G \) admit a tree 3-spanner for which vertices \( u \), \( v \) form a 5-center?
Vertices that are at distance 2 in $G$ from $\{u, v\}$ induce in $G$ a number of components: $Q = \{X \subseteq G : X$ is a component of $G \setminus N_G[u, v]\}$. Note that if $Q$ is empty, then $G$ trivially admits such a tree 3-spanner, so hereafter $Q$ is not empty.

2.1 Structures related to each member of $Q$

It can be proved that all the vertices of a component in $Q$ are in the same component of $T$ minus edge $uv$, for every tree 3-spanner $T$ of $G$ with 5-center $uv$. The adjacencies in such a spanner of neighbors in $G$ of a component in $Q$ depend on the placement of the component in the spanner, i.e. the neighbors of a component should be close to the component. For each $Q$ in $Q$ consider the following sets to store such dependencies.

1. In case that the vertices of $Q$ are to be closer to $u$ than to $v$ in the under construction spanner declare sets:
   - $U_{Q,u}$ to store neighbors of $v$ but not of $u$ that should be at distance 2 from the 5-center in the spanner and closer to $u$ than to $v$. (the Up vertices).
   - $D_{Q,u}$ to store neighbors of $u$ that should be neighbors of $u$ in the spanner as well (the Down vertices).

2. In case that the vertices of $Q$ are to be closer to $v$ than to $u$ in the under construction spanner similarly declare sets: $U_{Q,v}$, $D_{Q,v}$.

These sets take some initial values and then they are defined through an iteration. The definitions of the sets related to vertex $u$ follow, while the sets related to $v$ are defined similarly.

\[
U_{Q,u} := N_G(Q) \setminus N_G(u) \\
D_{Q,u} := N_G(Q) \cap N_G(u) \\
do \\
U_{Q,u} := U_{Q,u} \cup (((N_G(U_{Q,u}^a) \setminus D_{Q,u}^a) \cap N_G(u, v)) \setminus N_G(u)) \\
D_{Q,u} := D_{Q,u} \cup (((N_G(U_{Q,v}^u) \setminus D_{Q,u}^u) \cap N_G(u, v)) \cap N_G(u)) \\
until((N_G(U_{Q,v}^a) \setminus D_{Q,v}^a) \cap N_G(u, v) = \emptyset)
\]
2.2 Formation of complexes

In case that the vertices of a component in $Q$ are to be closer to $u$ than to $v$ in the under construction spanner, the vertices of some other components in $Q$ should also be closer to $u$. The structure of a complex is to describe such consequences. Let $Q$ be a component in $Q$ and declare the following sets and structures:

1. In case that the vertices of $Q$ are to be closer to $u$ than to $v$ in the under construction spanner declare sets:
   - $M^Q_{C,u}$ to store components in $Q$ that should follow $Q$ (the components of the complex).
   - $U^Q_{C,u}$ to store neighbors of $v$ but not of $u$ that will be at distance 2 from the 5-center in the spanner (the Up vertices).
   - $D^Q_{C,u}$ to store neighbors of $u$ that will be neighbors of $u$ in the spanner as well (the Down vertices).
   - $R^Q_{C,u}$ to store vertices in $D^Q_{C,u}$ that can carry the paths in the spanner from vertices in $Q$ to the 5-center (the Representatives of $Q$).

   Also, to refer to these sets, let $C^Q_{u}$ point to the above four sets.

2. In case that the vertices of $Q$ are to be closer to $v$ than to $u$ in the under construction spanner similarly declare sets: $M^Q_{C,v}$, $U^Q_{C,v}$, $D^Q_{C,v}$, $R^Q_{C,v}$, while $C^Q_{v}$ points to these sets.

Again, these sets take some initial values and then they are defined through an iteration. The definitions of sets related to vertex $u$ follow, while sets related to $v$ are defined similarly.

\[
\begin{align*}
M^Q_{C,u} &:= \{Q\} \\
U^Q_{C,u} &:= U^Q_{C,u} \\
D^Q_{C,u} &:= D^Q_{C,u} \\
R^Q_{C,u} &:= \emptyset \\
do &
M^Q_{C,u} := M^Q_{C,u} \cup \{X \in Q \setminus M^Q_{C,u} : N_G(X) \cap U^Q_{C,u} \neq \emptyset\} \\
U^Q_{C,u} := U^Q_{C,u} \cup (N_G(\cup M^Q_{C,u}) \setminus N_G(u)) \\
D^Q_{C,u} := D^Q_{C,u} \cup (N_G(\cup M^Q_{C,u}) \cap N_G(u)) \\
do &
U^Q_{C,u} := U^Q_{C,u} \cup (((N_G(U^Q_{C,u}) \setminus D^Q_{C,u}) \cap N_G(u,v)) \setminus N_G(u)) \\
D^Q_{C,u} := D^Q_{C,u} \cup (((N_G(U^Q_{C,u}) \setminus D^Q_{C,u}) \cap N_G(u,v)) \cap N_G(u)) \\
until((N_G(U^Q_{C,u}) \setminus D^Q_{C,u}) \cap N_G(u,v) = \emptyset)
until(\{X \in Q \setminus M^Q_{C,u} : N_G(X) \cap U^Q_{C,u} \neq \emptyset\} = \emptyset) \\
R^Q_{C,u} := \{x \in D^Q_{C,u} : N_G(x) \supseteq (U^Q_{C,u} \cup V(\cup M^Q_{C,u}))\} \\
C^Q_{u} := \langle M^Q_{C,u}, U^Q_{C,u}, D^Q_{C,u}, R^Q_{C,u}\rangle
\end{align*}
\]
2.3 Putting complexes together

Each complex is to be attached either to $u$ or to $v$. Also, two complexes are thought to be compatible to each other in case that the implementation of one does not prevent the implementation of the other. This leads to the decision whether graph $G$ admits a tree 3-spanner for which pair $u, v$ is a 5-center.

For each $Q$ in $Q$, ordered sets $C^{Q,u}$ and $C^{Q,v}$ have already been defined. For $Q, Q'$ in $Q$, it holds that if $Q'$ is in $M^{Q,u}_C$, then $C^{Q,u}$ and $C^{Q',u}$ are the same, i.e. $M^{Q,u}_C = M^{Q',u}_C$, $U^{Q,u}_C = U^{Q',u}_C$, $D^{Q,u}_C = D^{Q',u}_C$, and $R^{Q,u}_C = R^{Q',u}_C$.

Along these lines, procedure `identify` below assigns to the entries of the first argument the values of the entries of the second one. Each auxiliary variable $C^u_i$ has entries $M^u_{C,i}, U^u_{C,i}, D^u_{C,i}$ and $R^u_{C,i}$, while each $C^v_i$ has similar entries.

```
X := an element in Q
identify(C^u_1, C^{X,u})
C^u := \{C^u_1\}
i := 1
while(Q \ (\bigcup_{Y \in C^u} M^u_Y) \neq \emptyset)
{
    X := an element in Q \ (\bigcup_{Y \in C^u} M^u_Y)
i := i + 1
    identify(C^u_i, C^{X,u})
    C^u := C^u \cup \{C^u_i\}
}
similarly evaluate C^v
```

Let $\Gamma$ be a graph with vertex set:

$V(\Gamma) := \{X \in C^u : R^u_X \neq \emptyset\} \cup \{X \in C^v : R^v_X \neq \emptyset\}$

and edge set defined as follows:

$XY \in E(\Gamma)$ if and only if

$((X \in (C^u \cap V(\Gamma)) \ni Y \lor X \in (C^v \cap V(\Gamma)) \ni Y) \lor (X \in (C^u \cap V(\Gamma)) \ni Y \in (C^v \cap V(\Gamma)) \ni \land M^u_X \land M^v_Y = \emptyset \land (D^u_X \cup U^u_X) \cap (D^v_Y \cup U^v_Y) = \emptyset))$

If ($\Gamma$ contains a clique $K$: $\bigcup_{X \in (K \cap C^u)} M^u_X \cup \bigcup_{X \in (K \cap C^v)} M^v_X = Q$)
then output(YES)
2.4 Finding such a clique

If a component in $Q$ is contained only in one vertex of $\Gamma$, then this vertex should be in $K$ to meet the containment requirement.

\[
K := \emptyset
\]

\[
\text{FLAG} := (Q \subseteq (\bigcup_{X \in (V(\Gamma) \cap C_u)} M_X^u \cup \bigcup_{X \in (V(\Gamma) \cap C_v)} M_X^v))
\]

\[
\text{while}(\{Q \setminus (\bigcup_{X \in (V(\Gamma) \cap C_u)} M_X^u \cup \bigcup_{X \in K} M_X^u) \neq \emptyset\) AND \text{FLAG})
\]

\[
K := K \cup \{X \in (C_v \cap V(\Gamma)) : \exists Y \in Q(Y \in M_X^u \land Y \not\in \bigcup_{Z \in (V(\Gamma) \cap C_u)} M_Z^u)\}
\]

\[
V(\Gamma) := V(\Gamma) \setminus \{X \in V(\Gamma) : \exists Y \in K \land XY \not\in E(\Gamma)\}
\]

\[
\text{FLAG} := (Q \subseteq (\bigcup_{X \in (V(\Gamma) \cap C_u)} M_X^u \cup \bigcup_{X \in (V(\Gamma) \cap C_v)} M_X^v))
\]

IF \text{FLAG output}(K \cup (V(\Gamma) \setminus C_v) induces such a clique)

3 Correctness of the algorithm

It turns out that it suffices to examine tree 3-spanners with a 5-center whose vertices are as close to the 5-center as possible.

**Definition 3** A tree 3-spanner $T$ of a graph $G$, where $uv$ is a 5-center of $T$, is concentrated if and only if, first, there is no neighbor of $u$ in $G$ which is closer in $T$ to $u$ than to $v$ and it is not a neighbor of $u$ in $T$ and, second, there is no neighbor of $v$ in $G$ which is closer in $T$ to $v$ than to $u$ and it is not a neighbor of $v$ in $T$.

**Lemma 2** If $G$ admits a tree 3-spanner with 5-center $uv$, then $G$ admits a concentrated tree 3-spanner with 5-center $uv$.

**Proof.** Let $T$ be a tree 3-spanner of $G$ with 5-center $uv$ and let $w$ be a vertex of $T$ which certifies that $T$ is not concentrated. Hence, $w$ is a leaf of $T$. Assume that $wq$ is an edge of $T$. Also, assume without loss of generality that $w$ is a neighbor of $u$ in $G$ and $w$ is closer in $T$ to $u$ than to $v$. Then, graph with vertex set $V(T)$ and edge set $E(T) - wq + wu$ is a tree 3-spanner of $G$ with 5-center $uv$ that has fewer vertices which certify that it is not concentrated than $T$ has. $\square$

**Lemma 3** Let $G$ be a graph that admits a concentrated tree 3-spanner $T$ with 5-center $uv$. Also, let $Q$ be a component in $Q$ which contains a vertex at distance 2 from $u$ in $T$. Then, every edge of the form $ux$, where $x$ is in $D_C^{Q,u}$ belongs to $E(T)$. Also, there is a vertex $r$ in $R_C^{Q,u}$, such that every edge of the form $rx$, where $x$ is in $(\bigcup_{X \in M_C^{Q,u}} V(X)) \cup U_C^{Q,u}$, belongs to $E(T)$.

**Proof.** Prove this lemma with induction on the number of iterations of the outer loop toward the construction of $C^{Q,u}$ (section 2.2), except of the requirement that $r$ should be in $R_C^{Q,u}$. 6
For 0 iterations, since a vertex of $Q$ is at distance 2 from $u$ in $T$, then all vertices of $Q$ are closer in $T$ to $u$ than to $v$, because of Lemma 1. Also, it cannot be that two vertices of $Q$ are connected to $u$ in $T$ through different vertices in $N_T(u)$, because of the same lemma and the fact that $uv$ is a 5-center of $T$; hence, all vertices in $Q$ are adjacent in $T$ to the same vertex, say $r$, where $r$ has to be in $D^{Q,u}$. Vertices which are added iteratively (section 2.1) to $U^{Q,u}$ should be adjacent in $T$ to $r$, because, first, these vertices are not adjacent to $u$, second, they have to be within distance 3 from particular vertices in $Q$ union the so far constructed set $U^{Q,u}$ and, third, $uv$ is a 5-center of $T$. Finally, vertices in $D^{Q,u}$ should be within distance 3 in $T$ from particular vertices in $Q \cup U^{Q,u}$ and as close in $T$ to the 5-center as possible, since $T$ is a concentrated tree 3-spanner with 5-center $uv$; so, all vertices in $D^{Q,u}$ are adjacent to $u$ in $T$. Note that for 0 iterations $M_{Q,u}^0 = \{Q\}$, $U_{C}^{Q,u} = U^{Q,u}$, $D_{C}^{Q,u} = D^{Q,u}$.

As sets $M_{Q,u}^0$, $U_{C}^{Q,u}$, and $D_{C}^{Q,u}$ increase iteratively, following similar syllogism as above, one verifies that the induction step holds. Finally, vertex $r$ because of its neighborhood in $T$ belongs to $R_{C}^{Q,u}$. \hfill \Box

**Proposition 1** A graph $G$ admits a tree 3-spanner with 5-center $uv$ if and only if the algorithm in section 2 outputs $\text{YES}$ on input $(G, u, v)$.

**Proof.** Assume that $G$ admits a tree 3-spanner with 5-center $uv$. Then, due to Lemma 2, $G$ admits a concentrated tree 3-spanner $T$ with 5-center $uv$.

Given $G$ and $uv$, sets $C^u$ and $C^v$ are constructed. Let $C$ be an element of $C^u$. If a vertex in a component in $M_{C}^{u}$ is closer in $T$ to $u$ than to $v$, then all vertices in $(\bigcup_{X \in M_{C}^{u}} V(X)) \cup U_{C}^{u}$ are adjacent in $T$ to a vertex in $R_{C}^{u}$, due to Lemma 3. Therefore, $R_{C}^{u}$ is not empty. Similarly, if $C$ is an element of $C^v$, such that a vertex in a component in $M_{C}^{v}$ is closer in $T$ to $v$ than to $u$, then $R_{C}^{v}$ is not empty. Let $A$ be the subset of $C^u$ such that $X \in A$ if and only if a vertex in a component in $M_{X}^{v}$ is closer in $T$ to $u$ than to $v$. Also, let $B$ be the subset of $C^v$ such that $X \in B$ if and only if a vertex in a component in $M_{X}^{v}$ is closer in $T$ to $u$ than to $u$.

First, $A \cup B \subseteq V(\Gamma)$, because $A$ and $B$ contain complexes that have nonempty sets of representatives. Second, there is no component in $Q$, which is not in $\bigcup_{X \in A} M_{X}^{v} \cup \bigcup_{X \in B} M_{X}^{v}$. Third, for any complex $C_1$ in $A$ and for any complex $C_2$ in $B$, on one hand, $M_{C_1}^{u} \cap M_{C_2}^{u} = \emptyset$ and, on the other hand, $(U_{C_1}^{u} \cup D_{C_1}^{u}) \cap (U_{C_2}^{u} \cup D_{C_2}^{u}) = \emptyset$, because otherwise it is a contradiction to Lemma 3. Therefore, $A \cup B$ induces a clique $K$ in $\Gamma$, such that $\bigcup_{X \in (K \cap C^v)} M_{X}^{v} \cup \bigcup_{X \in (K \cap C^v)} M_{X}^{u} = Q$. Hence, the algorithm outputs $\text{YES}$.

For the converse, assume that the algorithm on input $(G, u, v)$ outputs $\text{YES}$. Let $K$ be the clique of $\Gamma$ that the algorithm picked to cover all the components in $Q$.

Let $X$ be in $K$. In case that $X$ belongs to $C^u$, let $r_X$ be a vertex in $R_{X}^{u}$, where $R_{X}^{u}$ is nonempty, since $X$ is a vertex of $\Gamma$. Furthermore, let $E_X$ be the set that contains exactly the following edges: first, all edges of the form $r_X x$, where $x$ is in $(\bigcup_{Y \in M_{X}^{v}} V(Y)) \cup U_{X}^{u}$ and, second, all edges of the form $ux$, where $x$ is in $D_{X}^{u}$. In case that $X$ belongs to $C^v$, define $E_X$ similarly.
Also, let \( F_u \) be the edges of the form \( ux \), where \( x \) is in \( N_G(u) \), \((\bigcup_{Y \in K \cap C_u} (U_Y^u \cup D_Y^u)) \cup \bigcup_{Y \in K \cap C_u} (U_Y^u \cup D_Y^v) \). Finally, let \( F_v \) be the edges of the form \( vx \), where \( x \) is in \( N_G(v) \), \((\bigcup_{Y \in K \cap C_v} (U_Y^u \cup D_Y^v)) \cup \bigcup_{Y \in K \cap C_v} (U_Y^u \cup D_Y^v) \cup N_G(v) \).

Now, let \( T \) be the graph with vertex set \( V(G) \) and edge set \( \bigcup_{X \in K} E_X \cup F_u \cup F_v \cup \{uv\} \).

For each \( X \) in \( K \) the edges in \( E_X \) form a tree 3-spanner of the subgraph of \( G \) induced by vertices incident to these edges, because of the definition of a complex. Also, for \( X, Y \) in \( K \cap C_u \), the intersection of \( E_X \) and \( E_Y \) is a set of edges of the form \( ux \), where \( x \) is in \( D_X^u \cap D_Y^u \). Therefore, one can see that edges in \( \bigcup_{X \in K \cap C_u} E_X \) form a tree, say \( T_u \); similarly, edges in \( \bigcup_{X \in K \cap C_v} E_X \) form a tree, which, additionally, is vertex disjoint to \( T_u \). Obviously, no edge in \( F_u \cup F_v \) can participate in a cycle in \( T \). First, using the aforementioned remarks, it can be shown that \( T \) has no cycles. Second, one can conclude that each vertex in \( V(G) \) is within distance 2 in \( T \) from \( uv \); note that \( K \) covers \( Q \). By these two facts, \( T \) is a spanning tree of \( G \) with 5-center \( uv \).

To see that \( T \) is also a 3-spanner of \( G \), observe that for each complex in \( K \) only its down vertices may be adjacent in \( G \) to vertices that are not registered in the complex and are other than \( u \) and \( v \).

\[ \square \]

4 Concluding remarks

Clearly, the algorithm presented in section 2 on input \((G, u, v)\) halts before a polynomial on \(|G|\) number of steps. To determine whether \( G \) admits a tree 3-spanner of diameter at most 5, the algorithm should be run for each possible 5-center of the under construction spanner; of course, there are only \(|E(G)|\) such centers. Therefore, there is an efficient algorithm to determine whether a graph admits a tree 3-spanner of diameter at most 5.

Also, each concentrated tree 3-spanner of the input graph \( G \) with 5-center \( uv \) may be constructed through the algorithm, when minor changes in evaluating its output are made, following the method used in the proof of correctness. Indeed, one can show that, depending on the choice of clique \( K \), the choice of representatives of complexes in \( K \), and the choice of sets \( F_u \) and \( F_v \), a list of all concentrated tree 3-spanners of \( G \) with 5-center \( uv \) can be produced.

Finally, it seems that this procedure may be used as a building block toward the design of an efficient algorithm to decide graphs that admit a tree 3-spanner. Note that the problem to decide graphs that admit a tree 3-spanner is reduced to the problem to determine whether a graph admits a tree 3-spanner with given \( k \)-center, where \( k \) is part of the input.

References

[1] Srinivasa Arikati, Danny Z. Chen, L. Paul Chew, Gautam Das, Michiel Smid, and Christos D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In *Algorithms—ESA ’96 (Barcelona)*, pages 514–528. Springer, Berlin, 1996.
[2] Baruch Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32(4):804–823, October 1985.

[3] Hans-Jürgen Bandelt and Andreas Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Adv. in Appl. Math.*, 7(3):309–343, 1986.

[4] J. A. Bondy. Trigraphs. *Discrete Mathematics*, 75:69–79, 1989.

[5] Leizhen Cai. *Tree Spanners: Spanning trees that approximate distances*. PhD thesis, Department of Computer Science, University of Toronto, 1992. (Available as Technical Report 260/92).

[6] Leizhen Cai and Derek G. Corneil. Tree spanners. *SIAM J. of Discrete Mathematics*, 8(3):359–378, 1995.

[7] L. Paul Chew. There are planar graphs almost as good as the complete graph. *J. Comput. System Sci.*, 39(2):205–219, 1989. Computational geometry.

[8] Sándor P. Fekete and Jana Kremer. Tree spanners in planar graphs. *Discrete Appl. Math.*, 108(1-2):85–103, 2001. International Workshop on Graph-Theoretic Concepts in Computer Science (Smolenice Castle, 1998).

[9] Papoutsakis Ioannis. *Tree Spanners of simple graphs*. PhD thesis, Department of Computer Science, University of Toronto, 2013. (Available at university T-space on the internet).

[10] Hoàng-Oanh Le and Van Bang Le. Optimal tree 3-spanners in directed path graphs. *Networks*, 34(2):81–87, 1999.

[11] M. S. Mandanlal, G. Venkatesan, and C. Pandu Rangan. Tree 3-spanners on interval, permutation, and regular bipartite graphs. *Information Processing Letters*, 59:97–102, 1996.

[12] D. Peleg and E. Upfal. A tradeoff between space and efficiency for routing tables. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1988.

[13] David Peleg and Eilon Reshef. A variant of the arrow distributed directory with low average complexity (extended abstract). In *Automata, languages and programming (Prague, 1999)*, pages 615–624. Springer, Berlin, 1999.

[14] David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

[15] Y. Rabinovich and R. Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete Comput. Geom.*, 19(1):79–94, 1998.

[16] G. Venkatesan, U. Rotics, M. S. Madanlal, J. A. Makowsky, and C. Pandu Rangan. Restrictions of minimum spanner problems. *Information and Computation*, 136(2):143–164, 1997.

[17] D. B. West. *Introduction to Graph Theory*. Prentice Hall, Inc., 1996.