# Finding a Tree 4-Spanner on Trapezoid Graphs

Hon-Chan Chen[1] , Shin-Huei Wu[2], Chang-Biau Yang[2]

[1] *Department of Information Management,*
*National Chin-yi Institute of Technology,*
*Taichung, Taiwan*

[2] *Department of Computer Science and Engineering,*
*National Sun Yat-sen University,*
*Kaohsiung, Taiwan*

**Abstract**

*Let G be a graph. A t-spanner of G is a spanning subgraph H of G such that the distance between any two vertices in H is at most t times their distance in G. A tree t-spanner is a t-spanner which is a tree. In this paper, we propose the first O(n) algorithm for finding a tree 4-spanner on trapezoid graphs.*

**Keywords:** tree, spanner, trapezoid graph, algorithm

## 1. Introduction

Let $G = (V, E)$ be a graph with vertex set $V$ and edge set $E$. A graph $H = (V', E')$ is a *spanning subgraph* of $G$ if $V' = V$ and $E' \subset E$. A spanning subgraph $H$ is a *t-spanner* of $G$ if the distance between any two vertices in $H$ is at most $t$ times their distance in $G$. If $t$-spanner $H$ is also a tree, then $H$ is a *tree t-spanner* of $G$.

The tree spanner problem has applications in communication networks, distributed systems, synchronizers, and so on [1, 8]. Cai et al. [1] have proved that the problem of finding a tree $t$-spanner on general graphs can be solved in polynomial time when $t < 3$, and it is an NP-complete problem when $t > 3$. The case where $t = 3$ is an open problem; however, it is conjectured to be NP-complete. Many researchers have focused the tree spanner problem on special types of graphs such as hypercube graphs, permutation graphs, and interval graphs [4, 7].

In this paper, we will propose an $O(n)$ algorithm for finding a tree 4-spanner of a trapezoid graph, assuming the trapezoid diagram is given. This paper is organized as follows. In Section 2, the definition and some properties of a trapezoid graph are introduced. In Section 3, we present the algorithm for finding a tree 4-spanner on trapezoid graphs. The $O(n)$ implementation of our algorithm is shown in Section 4. Finally, in Section 5, we give our concluding remarks.

## 2. Preliminaries

The type of trapezoid graphs was introduced by Dagan et al. [3] and independently by Corneil et al. [2]. A trapezoid graph can be represented by a trapezoid diagram such that each trapezoid in the diagram corresponds to a distinct vertex of the trapezoid graph. A trapezoid $i$ has four corner points $i.a$, $i.b$, $i.c$, and $i.d$ which respectively stand for the upper left, the upper right, the lower left, and the lower right corner of trapezoid $i$. All $a$ and $b$ corner points are on the upper line while $c$ and $d$ corner points are on the lower line in the trapezoid diagram. Two corner points $i$ and $j$ at the same line are denoted by $i < j$ if $i$ is at the left of $j$. We assume that all trapezoids are labeled by the order of their $b$ corner points and any two distinct corners do not share a common corner point. In a trapezoid graph, two vertices $i$ and $j$ are adjacent if and only if trapezoids $i$ and $j$ intersect each other in the corresponding trapezoid diagram. Figure 1 shows an example of a trapezoid graph.

Let $x$ and $y$ be two vertices of a trapezoid graph $G$. Denote the relation of $x$ and $y$ by $x < y$ if $x.b < y.b$ in the trapezoid diagram. Moreover, denote $x << y$ if $x.b < y.a$ and $x.d < y.c$; i.e., trapezoid $x$ is entirely on the left of trapezoid $y$ in the diagram. In the example of Figure 1, $1 << 2$ and $2 < 3$. It is easy to see that for three vertices $i$, $j$, and $k$ of $G$, where $i < j < k$, if $i$ is adjacent to $k$, then $j$ is adjacent to $i$ or $k$. In addition, the length of any cycle in $G$ is at most 4.

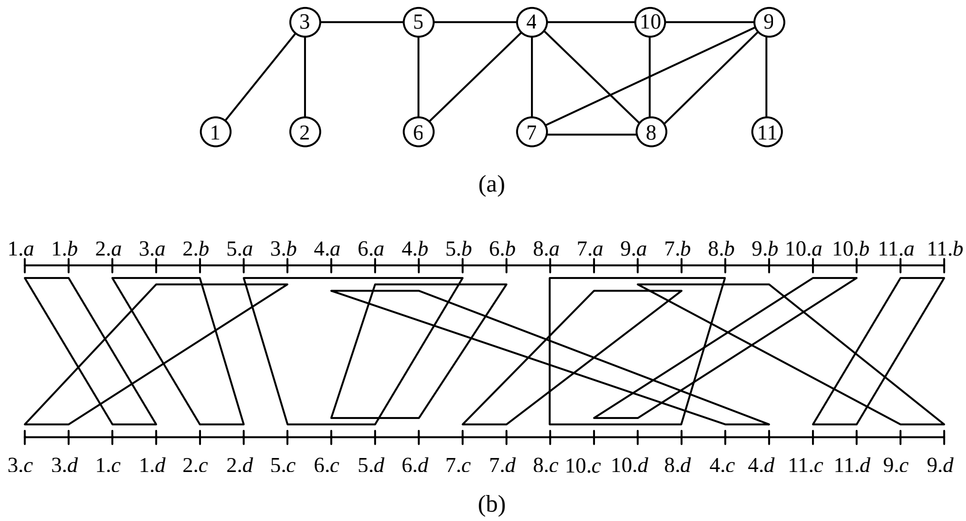


Figure 1. An example of a trapezoid graph. (a) A trapezoid graph. (b) The corresponding diagram.

A *dominating set* $S$ of a graph $G$ is a vertex set such that any vertex in $G - S$ is adjacent to at least one vertex of $S$. If vertices of $S$ induce a connected subgraph of $G$, then $S$ is a *connected dominating set*. A *minimum cardinality connected dominating set* (*MCCDS*) has the minimum cardinality among all connected dominating sets. We denote the cardinality of a MCCDS of $G$ by $\gamma(G)$. In every connected cocomparability graph $G$, there is a MCCDS that induces a simple path in $G$ [6]. Since the class of trapezoid graphs is a subclass of cocomparability graphs, the above property is followed in a trapezoid graph. Köhler [5] has presented an $O(n)$ algorithm for finding a MCCDS of a trapezoid graph. We call the path induced by a MCCDS a *minimum dominating path* (*MDP*). In Figure 1, $\gamma(G) = 5$ and {3, 5, 4, 10, 9} is a MCCDS which induces a MDP.

Our algorithm of finding a tree 4-spanner on trapezoid graphs includes two steps. In the first step, we apply Köhler's algorithm to determine a MDP. In the second step, we carefully add edges to the MDP to complete a tree 4-spanner.

## 3. The algorithm for finding a tree 4-spanner

Let $P = x_1 - x_2 - \ldots - x_k$ be a MDP of a trapezoid graph $G$, in which $x_1 < x_k$. For one vertex $x \in P$, let $g(x)$ be the position on $P$ where $x$ can be found searching from $x_1$ to $x_k$. For example, in Figure 1, $g(5) = 2$ and $g(4) = 3$.

Let $Q$ be the vertex set obtained from $G - P$, and let $v$ be a vertex of $Q$. Since $P$ is a dominating path, $v$ is adjacent to at least one vertex of $P$. The adjacency between $v$ and $P$ can be categorized into four types. With the adjacency type of each vertex $v \in Q$, we can construct a tree 4-spanner of $G$. Our algorithm is described below, and the correctness will be shown later.

**Algorithm A** (Finding a tree 4-spanner of a trapezoid graph)
**Input:** A connected trapezoid graph $G$ represented by a trapezoid diagram.
**Output:** A tree 4-spanner $T$ of $G$.
**Methods:**
**Step 1.** Apply Köhler's algorithm to find a minimum dominating path $P$ of $G$. Let $T = \emptyset$.
**Step 2.** Do one of the following three cases:
Case 1. $\gamma(G) = 1$. Assume vertex $x$ dominates all vertices of $Q = G - \{x\}$. For each vertex $v \in Q$, add edge $(v, x)$ to $T$.
Case 2. $\gamma(G) = 2$. Assume vertices $x$ and $y$ dominate all vertices of $Q = G - \{x, y\}$. At first, add edge $(x, y)$ to $T$. For each vertex $v \in Q$, if $v$ is adjacent to $x$, then add edge $(v, x)$ to $T$; otherwise, add edge $(v, y)$ to $T$.
Case 3. $\gamma(G) \geq 3$. At first, add all edges of $P$ to $T$. For each vertex $v$ of $Q = G - P$, attach $v$ to $T$ by one of the following four rules:
Rule 1. If $v$ is adjacent to only one vertex $x$ of $P$, then add edge $(v, x)$ to $T$.
Rule 2. If $v$ is adjacent to two vertices $x$ and $y$ of $P$ with $g(y) = g(x) + 1$, then add edge $(v, y)$ to $T$.
Rule 3. If $v$ is adjacent to two vertices $x$ and $y$ of $P$ with $g(y) = g(x) + 2$,

then add edge $(v, x)$ to $T$.

Rule 4. If $v$ is adjacent to three vertices $x$, $y$, and $z$ of $P$ with $g(x) + 1 = g(y) = g(z) - 1$, then add edge $(v, y)$ to $T$.

**End of Algorithm**

We illustrate Algorithm A by the example of Figure 1. In Figure 1, $\gamma(G) = 5$ and the MDP is $3 - 5 - 4 - 10 - 9$. Moreover, $g(3) = 1$, $g(5) = 2$, $g(4) = 3$, $g(10) = 4$, and $g(9) = 5$. Since $\gamma(G) = 5$, we do Case 3 of Step 2. After adding edges (3, 5), (5, 4), (4, 10), and (10, 9) to $T$, we attach each vertex of $Q = \{1, 2, 6, 7, 8, 11\}$ to $T$ by the four rules. Since vertices 1 and 2 are only adjacent to vertex 3, we add edges (1, 3) and (2, 3) to $T$ by Rule 1. Vertex 6 is adjacent to vertices 5 and 4 with $g(4) = g(5) + 1$, thus we add edge (6, 4) to $T$ by Rule 2. Vertex 7 is adjacent to vertices 4 and 9 with $g(9) = g(4) + 2$, and we add edge (7, 4) to $T$ by Rule 3. Since vertex 8 is adjacent to vertices 4, 10, and 9 with $g(9) - 1 = g(10) = g(4) + 1$, edge (8, 10) is added to $T$ by Rule 4. Finally, we add edge (11, 9) to $T$ since vertex 11 is only adjacent to vertex 9. The resulting $T$ shown in Figure 2 is a tree 4-spanner of the input trapezoid graph $G$ of Figure 1.

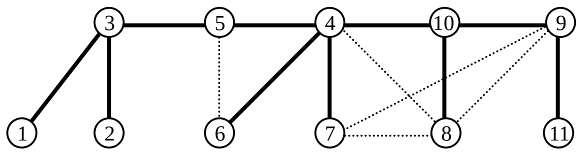


Figure 2. A tree 4-spanner of $G$.

Hereafter, we show the correctness of Algorithm A. For one vertex $v \in Q$, let $V_P(v)$ be the set of vertices in $P$ which are adjacent to $v$. Denote the vertex at the $i$th position on $P$ by $g^{-1}(i)$. For example, in Figure 1, $V_P(6) = \{5, 4\}$, $g(5) = 2$, and $g^{-1}(2) = 5$.

**Lemma 3.1.** *Let $v$ be a vertex of $Q$. Then, the adjacency between $v$ and $P$ is in one of the following four types*:

*Type* 1. *$v$ is adjacent to only one vertex $x$ of $P$*;

*Type* 2. *$v$ is adjacent to two vertices $x$ and $y$ of $P$ with $g(y) = g(x) + 1$*;

*Type* 3. *$v$ is adjacent to two vertices $x$ and $y$ of $P$ with $g(y) = g(x) + 2$*;

*Type* 4. *$v$ is adjacent to three vertices $x$, $y$, and $z$ of $P$ with $g(x) + 1 = g(y) = g(z) - 1$.*

**Proof.** Without loss of generality, let $g(x) = i$. Assume that the adjacency of $v$ and $P$ is not in the above four types. Then, $v$ is in one of the following three cases:

Case 1. $v$ is adjacent to two vertices $x$ and $y$ of $P$ with $g(y) \geq g(x) + 3$. In this case, the subpath of $P$ from $g^{-1}(1)$ to $y$ is

$g^{-1}(1) - \ldots - g^{-1}(i-1) - x - g^{-1}(i+1) - \ldots - g^{-1}(i+k) - y$,

where $k \geq 2$. Since $v$ is adjacent to $x$ and $y$, we can find that the path

$g^{-1}(1) - g^{-1}(2) - \ldots - g^{-1}(i-1) - x - v - y$

is shorter than the subpath of $P$ from $g^{-1}(1)$ to $y$. It contradicts that $P$ is a MDP.

Case 2. $v$ is adjacent to three vertices $x$, $y$, and $z$ of $P$ with $g(z) \geq g(x) + 3$, where $g(x) < g(y) < g(z)$. Since $v$ is adjacent to $x$ and $z$, the path

$g^{-1}(1) - g^{-1}(2) - \ldots - g^{-1}(i-1) - x - v - z$

is shorter than the subpath of $P$ from $g^{-1}(1)$ to $z$. It is also a contradiction.

Case 3. $v$ is adjacent to at least four vertices of $P$. It is trivial from Cases 1 and 2 that if $v$ is adjacent to at least four vertices of $P$, then $P$ is not a MDP.

Q.E.D.

Figure 3 shows the four adjacency types between one vertex $v \in Q$ and $V_P(v)$. We use black vertices and white vertices to represent vertices in $P$ and in $Q$ respectively.

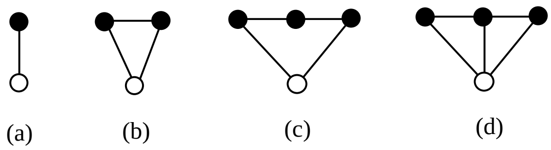


Figure 3. The four adjacency types. (a) Type 1. (b) Type 2. (c) Type 3. (d) Type 4.

**Lemma 3.2.** *Let $x$ and $y$ be two vertices of $P$. If $g(y) \geq g(x) + 2$, then $x << y$.*

**Proof.** It is obvious that if $x$ is adjacent to $y$ with $g(y) \geq g(x) + 2$, then $P$ is not a MDP.

Q.E.D.

**Lemma 3.3.** *Let $u$ and $v$ be two adjacent vertices of $Q$. If* $\max\{g(k) \mid k \in V_P(u)\} < \min\{g(k) \mid k \in V_P(v)\}$, *then* $\max\{g(k) \mid k \in V_P(u)\} + 1 = \min\{g(k) \mid k \in V_P(v)\}$.

**Proof.** Let $x = g^{-1}(\max\{g(k) \mid k \in V_P(u)\})$ and $y = g^{-1}(\min\{g(k) \mid k \in V_P(v)\})$. Assume to the contrary that there exists a cycle $C = u - x - \ldots - y - v - u$ in $G$ with $g(y) \geq g(x) + 2$. Then, the length of $C$ is at least 5. By Lemma 3.2 and the relation of $x$ and $y$, there is no arc in $C$. It is a contradiction since the length of any cycle in a trapezoid graph is at most 4.

Q.E.D.

Denote the distance between two vertices $u$ and $v$ in a graph $G$ by $d_G(u, v)$. Let $u$, $v$ be two adjacent vertices of $Q$ and $T^*$ be the tree obtained from attaching $u$ and $v$ to $V_P(u)$ and $V_P(v)$ by the rules of Algorithm A. According to Lemmas 3.1 and 3.3, we can list all possible cases of $T^*$ as shown in the following figures. We use solid lines (dotted lines, respectively) to represent edges in $T^*$ (not in $T^*$, respectively). Some of these cases result in $d_{T^*}(u,v) > 4$; however, we will show that they are contradictions.

Figure 4. Two cases of the combination of Types 1 and 1.

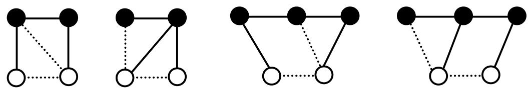

Figure 5. Four cases of the combination of Types 1 and 2.

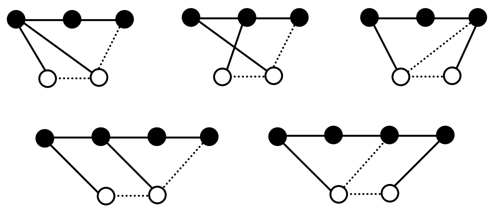


Figure 6. Five cases of the combination of Types 1 and 3.

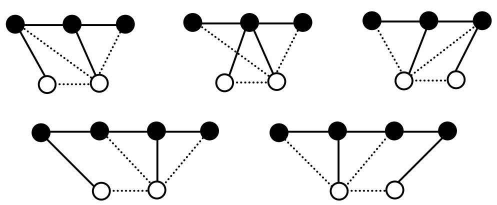

Figure 7. Five cases of the combination of Types 1 and 4.

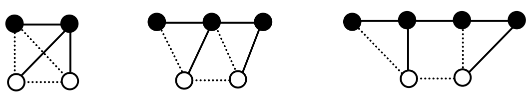

Figure 8. Three cases of the combination of Types 2 and 2.

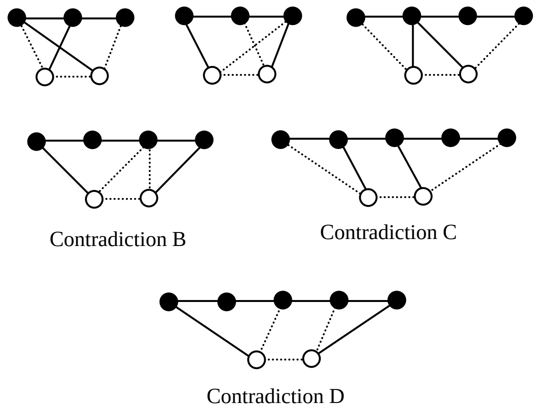


Figure 9. Six cases of the combination of Types 2 and 3.

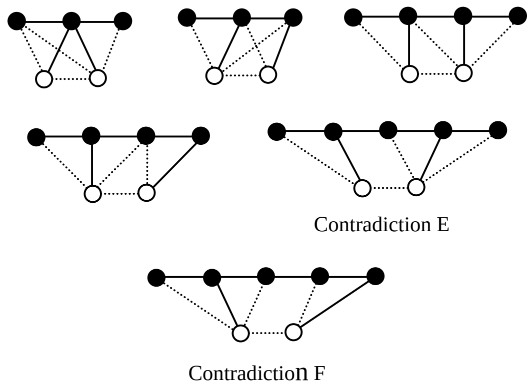


Figure 10. Six cases of the combination of Types 2 and 4.

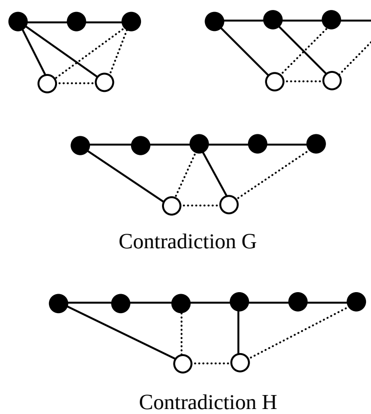


Figure 11. Four cases of the combination of Types 3 and 3.

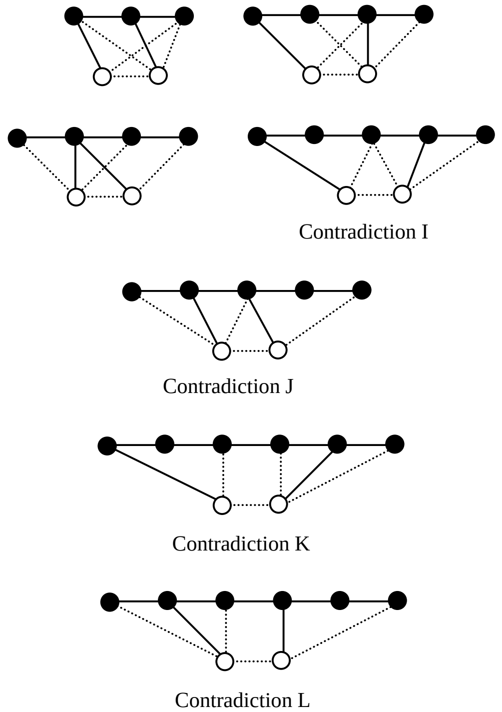


Figure 12. Seven cases of the combination of Types 3 and 4.

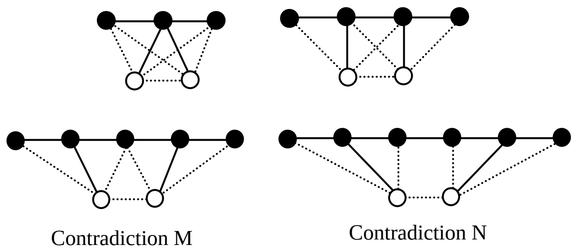


Figure 13. Four cases of the combination of Types 4 and 4.

**Lemma 3.4.** *Let $x$, $y$ be two vertices of $P$ and $v$ be a vertex of $Q$. If $g(y) \geq g(x) + 2$ and $v$ is adjacent to $y$ but not adjacent to $x$, then $x << v$.*

**Proof.** By Lemma 3.2, we know $x << y$. Since $v$ is not adjacent to $x$, we have $x << v$ or $v << x$. If $v << x$, then $v << x << y$ and $v$ is not adjacent to $y$. It is a contradiction, and we have $x << v$.

Q.E.D.

**Lemma 3.5.** *Let $x$, $y$ be two vertices of $P$ and $v$ be a vertex of $Q$. If $g(x) + 1 = g(y)$ and $v$ is adjacent to $x$ but not adjacent to $y$, then $v << y$.*

**Proof.** Since $v$ is not adjacent to $y$, $v << y$ or $y << v$. By Köhler's algorithm, the next vertex of $x$ in the MDP is the vertex with the maximal value of $b$ corner point or $d$ corner point among all vertices adjacent to $x$. If $y << v$, then $y.b < v.b$ and $y.d < v.d$ and $v$ must be the next vertex of $x$ in the MDP. It is a contradiction. Thus, $v << y$.

Q.E.D.

**Lemma 3.6.** *Let $u$, $v$ be two adjacent vertices of $Q$ and $T^*$ be the tree obtained from attaching $u$ and $v$ to $V_P(u)$ and $V_P(v)$ by the rules of Algorithm A. Then, $d_{T^*}(u,v) \leq 4$.*

**Proof.** In Figures 4 to 13, there are 14 cases, labeled by Contradictions A to N, which have $d_{T^*}(u,v) > 4$. Since the paths of black vertices on Contradictions C to N result in longer dominating paths, they cannot be subpaths of $P$, and Contradictions C to N do not exist. We show that Contradictions A and B do not exist either. Let $w$, $x$, $y$, and $z$ be four continuous vertices of $P$ with $g(w) = g(x) - 1 = g(y) - 2 = g(z) - 3$. For Contradiction A, assume $u$ is adjacent to $w$ and $y$ while $v$ is adjacent to $z$. By Lemmas 3.4 and 3.5, we have $u << x << v$ which contradicts that $u$ is adjacent to $v$. For Contradiction B, assume $u$ is adjacent to $w$ and $y$ while $v$ is adjacent to $y$ and $z$. Since $v$ is not adjacent to $x$, $v << x$ or $x << v$. If $v << x$, by Lemma 3.2, we have $v << x << z$. It contradicts that $v$ is adjacent to $z$. If $x << v$, then $u << x << v$. It also contradicts that $u$ is adjacent to $v$. This completes the proof.

Q.E.D.

**Theorem 3.7.** *Algorithm A finds a tree 4-spanner $T$ of a trapezoid graph $G$.*

**Proof.** Any edge $(x, y)$ of $G$ must be in one of the following three cases:

Case 1. $x, y \in P$. In this case, $|g(x) - g(y)| = 1$ by Lemma 3.2. Thus, $(x, y)$ is an edge of $T$.

Case 2. $x \in P$ and $y \in Q$. In this case, $x$ and $y$ are in one of the four adjacency types as described in Lemma 3.1. By the rules of Algorithm A, we obtain $d_T(x, y) \leq 3$.

Case 3. $x, y \in Q$. By Lemma 3.6, we have $d_T(x, y) \leq 4$.

Q.E.D.

## 4. The $O(n)$ implementation of Algorithm A

In Algorithm A, Step 1 determines a MDP of $G$ and Step 2 constructs a tree 4-spanner of $G$. Step 1 can be done in $O(n)$ time by applying Köhler's algorithm. In Step 2, we can easily find a tree 4-spanner in $O(n)$ time when $\gamma(G) \leq 2$.

When $\gamma(G) \geq 3$, we need to determine the adjacency type for each vertex of $Q$. Let $A[n, 3]$ be a matrix of $n$ rows and 3 columns, in which each element is initialized by zero. For one vertex $v \in Q$, $A[v, 1]$, $A[v, 2]$, and $A[v, 3]$ record the information of $g(x)$, $x \in V_P(v)$. For instance, in Figure 2, vertex 8 is adjacent to $g^{-1}(3)$, $g^{-1}(4)$, and $g^{-1}(5)$, then $A[8, 1] = 3$, $A[8, 2] = 4$, and $A[8, 3] = 5$. To complete array $A$, we visit all vertices of $P$ from $g^{-1}(1)$ to $g^{-1}(\gamma(G))$. During visiting one vertex $x \in P$, we update $A[v, 1]$, $A[v, 2]$, and $A[v, 3]$ by the following two conditions for all vertices $v \in Q$ adjacent to $x$:

Condition 1. $A[v, 1] = 0$. Then, let $A[v, 1] = g(x)$.

Condition 2. $A[v, 1] \neq 0$. If $g(x) = A[v, 1] + 1$, then let $A[v, 2] = g(x)$; otherwise, let $A[v, 3] = g(x)$.

After completing array $A$, we can know the adjacency type of each vertex $v \in Q$: $v$ is with Type 1 if $A[v, 1] \neq 0$ and $A[v, 2] = A[v, 3] = 0$; $v$ is with Type 2 if $A[v, 1] \neq 0$, $A[v, 2] \neq 0$, and $A[v, 3] = 0$; $v$ is with Type 3 if $A[v, 1] \neq 0$, $A[v, 2] = 0$, and $A[v, 3] \neq 0$; $v$ is with Type 4 if $A[v, 1] \neq 0$, $A[v, 2] \neq 0$, and $A[v, 3] \neq 0$. Since the number of edges incident to both $P$ and $Q$ is at most $3n$, array $A$ can be obtained in $O(n)$ time. Finally, we add edges to $T$ for all vertices of $Q$ by the rules of Algorithm A. The complexity of Algorithm A is therefore $O(n)$.

## 5. Concluding remarks

In this paper, we propose the first $O(n)$ time algorithm for finding a tree 4-spanner of a trapezoid graph. The main idea of our algorithm is to find a minimum dominating path $P$ at first as the backbone, then attach other vertices not in $P$ to the backbone. The contribution of this paper is significant, and it may be motivated to apply the appraoch in this paper for a variety of spanner problems.

## References


[1] L. Cai and D. G. Corneil, Tree Spanners, *SIAM Journal on Discrete Mathematics*, Vol. 8, 1995, pp. 359-387.

[2] D. G. Corneil and P. A. Kamula, Extensions of Permutation and Interval Graphs, *Congressus Numerantium*, Vol. 58, 1987, pp. 267-275.

[3] I. Dagan, M. C. Golumbic, and R. Y. Pinter, Trapezoid Graphs and Their Coloring, *Discrete Applied Mathematics*, Vol. 21, 1988, pp. 35-46.

[4] W. Duckworth and M. Zito, Sparse Hypercube 3-Spanners, *Discrete Applied Mathematics*, Vol. 103, 2000, pp. 289-295.

[5] E. Köhler, Connected Domination and Dominating Clique in Trapezoid Graphs, *Discrete Applied Mathematics*, Vol. 99, 2000, pp. 91-110.

[6] D. Kratsch, L. Stewart, Domination in Cocomparability Graphs, *SIAM Journal on Discrete Mathematics*. Vol. 6, 1993, pp. 400-417.

[7] M. S. Madanlal, G. Venkatesan, C. Pandu Rangan, Tree 3-Spanners on Interval, Permutation and Regular Bipartite Graphs, *Information Processing Letters*, Vol. 59, 1996, pp. 97-102.

[8] D. Peleg and J. Ullman, An Optimal Synchronizer for the Hypercube, *SIAM Journal on Computing*, Vol. 18, 1989, pp. 740-747.