Tamsui Oxford Journal of Mathematical Sciences **24(1)** (2008) 11-24
Aletheia University

An Efficient Algorithm to Compute a Steiner Set and Steiner Tree on Trapezoid Graphs

Prabir K. Ghosh and Madhumangal Pal*

Department of Applied Mathematics with

Oceanology and Computer Programming,

Vidyasagar University, Midnapore-721102, India

Received March 12, 2006, Accepted June 7, 2006.

Abstract

This paper presents an efficient algorithm to compute a minimum cardinality Steiner set and Steiner tree on trapezoid graphs. The algorithm takes $O(m+n\sqrt{\log C})$ time for a trapezoid graph with n vertices and m edges, where cost of each arc is a non-negative integer number bounded by C.

Keywords and Phrases: *Design and analysis of algorithms, Spanning tree, Steiner set, Steiner tree, Trapezoid graph.*

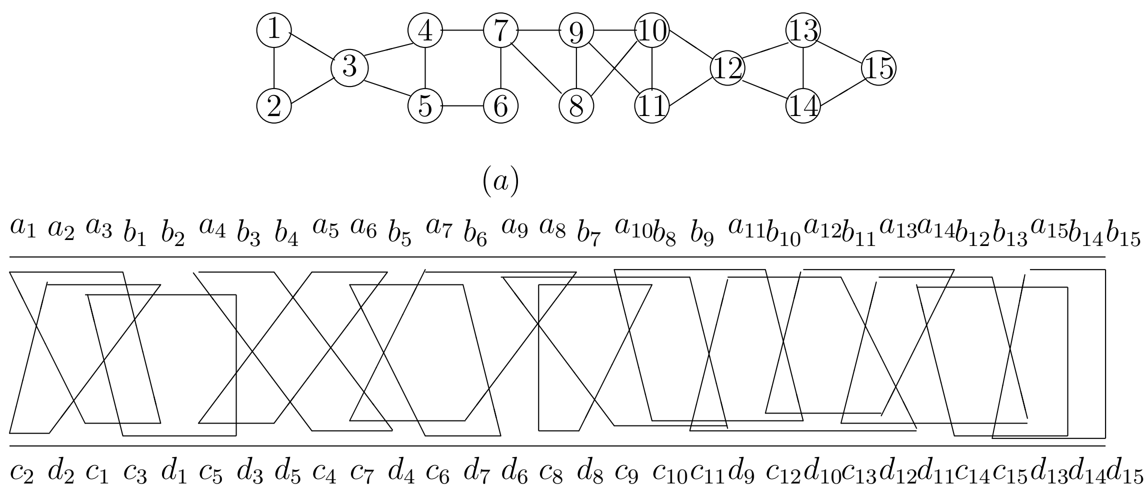
1. Introduction

A trapezoid T_i is defined by fore corner points $[a_i, b_i, c_i, d_i]$, where $a_i < b_i$ and $c_i < d_i$ with a_i, b_i lying on the top channel and c_i, d_i lying on the bottom channel of the trapezoid diagram (see Figure 1(b)). An undirected graph $G = (V, E)$ is called a *trapezoid graph* if it can be represented by a trapezoid diagram such that each vertex v_i in V corresponds to a trapezoid T_i and $(v_i, v_j) \in E$ if and only if the trapezoids T_i and T_j corresponding to the vertices

*Corresponding author E-mail: madhumangal@lycos.com

v_i and v_j intersect in the trapezoid diagram. Figure 1(a) and Figure 1(b) represent a trapezoid graph and corresponding trapezoid diagram. The class of trapezoid graphs includes two well known classes of intersection graphs: the permutation

graphs and the interval graphs [9]. The permutation graphs are obtained in the case where $a_i = b_i$ and $c_i = d_i$ for all i, and the interval graphs are obtained in the case where $a_i = c_i$ and $b_i = d_i$ for all i. In addition to this we assume that these n trapezoids are in increasing order of their right corner points on the top channel *i.e.*, for two trapezoids T_i and T_j, b_i is on the left of b_j iff $i < j$. The trapezoid graphs were first studied in [4, 5]. Trapezoid graphs can be recognized in $O(n^2)$ time [13]. These graphs are superclass of interval graphs and permutation graphs and subclass of cocomparability graphs [12]. There are so many works on Steiner tree in different type of graphs



(b)

Figure 1: A trapezoid graph and its corresponding trapezoid diagram.

are available in literature. This generalized problem can be reduced to the node weighted Steiner tree problem, for which algorithms with performance guarantees of $O(\log n)$ are known. Khuller et al. [11] have designed an approximation algorithms with small constant factors for this problem. Drake et al. [7] proposed no polynomial time approximation algorithm for the terminal Steiner tree problem has a performance ratio less then $(1 - O(1)) \ln n$ unless NP has slightly superpolynomial time algorithms. Promel [17] has designed an RNC approximation algorithm for the Steiner tree problem in graph with performance ratio 5/3 and RNC approximation algorithms for the Steiner tree problem in networks with performance ratio $5/3+\varepsilon$ for all $\varepsilon > 0$. Finding the minimum Steiner set of an arbitrary graph is known to be NP-complete [10]. Polynomial time algorithms are reported in the literature for some special classes of graphs such as strongly chordal graphs and distance heredity graphs [6, 18]. An $O(n^3)$ [3] time algorithm for this problem in permutation graphs was first given using a dynamic programming approach for the cardinality case and the result was improved to $O(m+n\log n)$ for the non-negative weights by reducing the problems into shortest path problem in a general network, and finally to $O(n+m)$ time using a new dynamic programming scheme. Mondal et al. [15] have designed an optimal algorithm to find Steiner tree on permutation graphs. Pal et al. [16] presents a linear time algorithm for the k-connected Steiner subgraph problem on an interval graph.

A path of a graph G is an alternating sequence of distinct vertices and edges, beginning and ending with vertices. The length of a path is the number of edges in the path. A path from vertex u to v is *shortest path* if there is no other path from u to v with length less then this.

For a given subset T of V, called a set of target vertices, a set $S \subset V$ is said to be a *Steiner set* for T in G if

(i) S is a subset of $V - T$, *i.e.*, $S \subseteq V - T$,

(ii) the subgraph induced by $S \cup T$ in G is connected.

The Steiner set S is said to be *minimum cardinality* Steiner set, if the cardinality of S is minimum. A spanning tree of a connected subgraph induced by $S \cup T$ in G is called a *Steiner tree*. The minimum cardinality Steiner set problem is the problem of finding the minimum number of vertices to connect a given set of target vertices T.

In this paper, an algorithm is presented to compute a minimum cardinality Steiner set and Steiner tree on trapezoid graphs. The proposed algorithm takes $O(m + n\sqrt{\log C})$ time.

2. Preliminaries

In this section, we present some definitions and results. These results are found useful in developing the proposed algorithm.

Lemma 1. *[2] Two vertices i and j of a trapezoid graph are not adjacent iff* *either (i)* $b_i < a_j$ *and* $d_i < c_j$ *or (ii)* $b_j < a_i$ *and* $d_j < c_i$.

Lemma 2. *[8] Let* G *be a trapezoid graph and* u, v *be two adjacent vertices of* G*. If* $u < w < v$*, then* w *is adjacent to at least one of* u *or* v.

Define the following term for a trapezoid graph $G = (V, E)$.

Let $N(u) = \{v : v \in V \text{ and } (v, u) \in E\}$ be the set of vertices which are adjacent to u.

$LN(u) = \{v : v \in N(u) \text{ and } v < u\}$ be the set of vertices which are less then u and adjacent to u, called left adjacent to u.

$RN(u) = \{v : v \in N(u) \text{ and } v > u\}$ be the set of vertices which are greater then u and adjacent to u, called right adjacent to u.

i.e., $N(u) = LN(u) \cup RN(u)$.

A Trapezoid T_j is said to be right adjacent to T_i if

(i) $a_i < a_j < b_i$ and $c_i < c_j < d_i$ or

(ii) $a_i < a_j < b_i$ and $c_j > d_i$ or

(iii) $a_j > b_i$ and $c_i < c_j < d_i$.

Conversely, T_i is called the left adjacent to T_j.

The possible cases when T_i is left adjacent to T_j or T_j is right adjacent to T_i are shown in Figure 2.

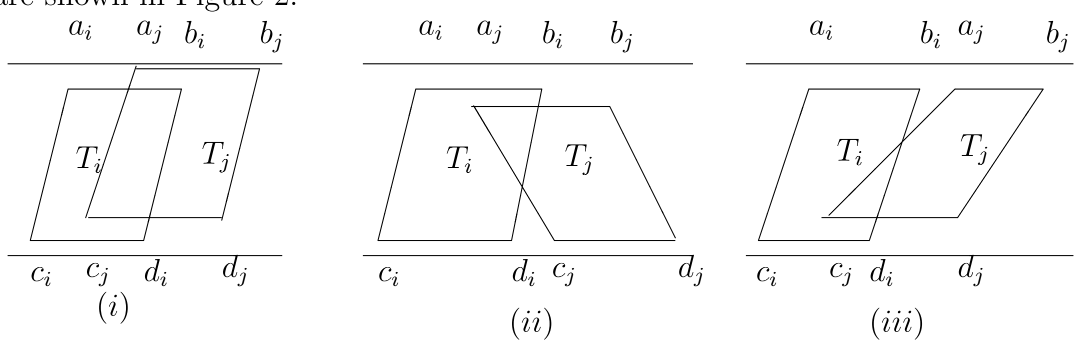


Figure 2: T_i is left adjacent to T_j

Given a trapezoid graph $G = (V, E)$ and a set of target vertices $T = \{x_1, x_2, \ldots, x_k\}$ of V, with $x_1 < x_2 < \cdots < x_k$, we note that the subgraph induced by T is not necessarily connected. In general, it contains some connected subgraphs and some isolated vertices. If x_1 and x_k belong to such two connected subgraphs we denote them by C_0 and C_1 respectively. If x_1 and x_k are isolated then $C_0 = \{x_1\}$ and $C_1 = \{x_k\}$. Four situations arise, (a)

none of C_0 and C_1 is a singleton, (b) C_0 is a singleton but not C_1, (c) C_1 is a singleton but not C_0 and (d) both C_0 and C_1 are singleton. If C_0 and C_1 are not singleton sets then we find two fictitious trapezoids T_s and T_t corresponding to C_0 and C_1 respectively. The four corner points of the trapezoid T_s are $min\{a_i, i \in C_0\}$, $max\{b_i, i \in C_0\}$, $min\{c_i, i \in C_0\}$ and $max\{d_i, i \in C_0\}$. Similarly, the four corner points of the trapezoid T_t are $min\{a_i, i \in C_1\}$, $max\{b_i, i \in C_1\}$, $min\{c_i, i \in C_1\}$ and $max\{d_i, i \in C_1\}$. Let s and t are the vertices corresponding to the trapezoids T_s and T_t respectively. We note that when C_0 is singleton then $s = x_1$ and when C_1 is singleton then $t = x_k$.

As the minimum cardinality Steiner set problem involves of finding the minimum number of vertices which connect a given set of target vertices, we construct an auxiliary graph $G' = (V', E')$, where $V' = V - LN(x_1) - RN(x_k)$ and $E' \subseteq E$ and also containing the fictitious trapezoids T_s and T_t (Figure 3).

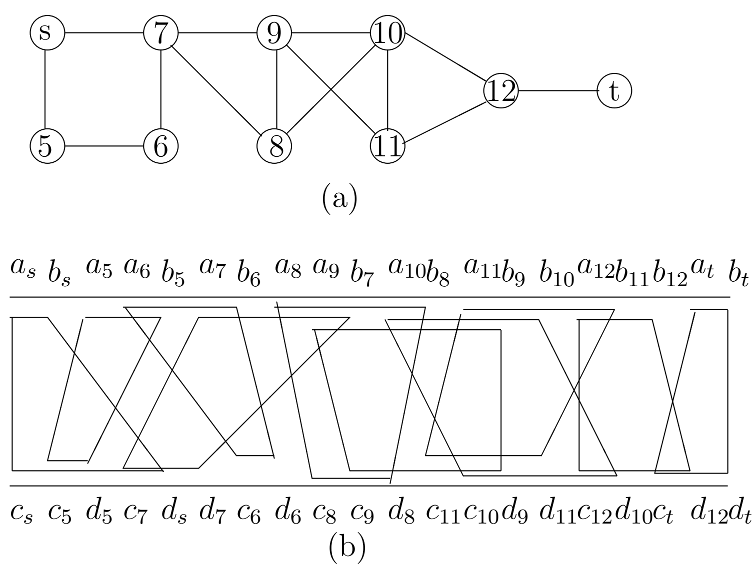


Figure 3: New trapezoid graph G' and its trapezoid diagram with respect to T.

Lemma 3. *If T_i is a left adjacent trapezoid to T_{x_1} in G then the trapezoid T_i is deleted from G, the reduced graph G' has no effect.*

Proof. Let T_k be a trapezoid in G and T_k is right adjacent to T_i. Since, T_k is right adjacent to T_i then T_k must adjacent to T_j, because T_j is right adjacent to T_i. But if T_i is deleted from G then T_k can not be deleted from G. Therefore, T_k becomes in G', *i.e.*, the auxiliary graph G' has no effect. □

Similar to the above result we have the following lemma.

Lemma 4. *If T_i is right adjacent to T_{x_k} in G then the trapezoid T_i is deleted from G, the reduced graph G' has no effect.*

Lemma 5. *If a trapezoid T_j is adjacent to at least one trapezoid of C_0 then T_j is adjacent to T_s.*

Proof. Let C_0 be a connected subgraph containing the target vertex x_1. Let T_s be the fictitious trapezoid corresponding the subgraph C_0. The four corner points of the trapezoid T_s are $min\{a_i, i \in C_0\}$, $max\{b_i, i \in C_0\}$, $min\{c_i, i \in C_0\}$ and $max\{d_i, i \in C_0\}$. So, T_s is the least region which includes all the members of C_O. If the trapezoid T_j is adjacent to at least one trapezoid of C_0 then that trapezoid of C_0 and T_j have a common region, *i.e.*, T_j and T_s have a common region. Therefore, T_j is adjacent to T_s. Hence the lemma. □

Similar to the above lemma we have the following result.

Lemma 6. *If a trapezoid T_j is adjacent to at least one trapezoid of C_1 then T_j is adjacent to T_t.*

To find the Steiner set we determine a shortest path between the vertices s and t containing maximum number of target vertices and let such shortest path be the subgraph $P' = (V_{P'}, E_{P'})$, where $V_{P'}$ and $E_{P'}$ respectively denote the set of vertices and edges. Let this path be $s \to v_1 \to v_2 \to \cdots \to v_{r-2} \to t$. We denote the path $v_1 \to v_2 \to \cdots \to v_{r-2}$ by the subgraph $P = (V_P, E_P)$. We now consider the set $V_P - T$ and denote it by S. It can be shown that this set S is a Steiner set.

For the graph of Figure 1, let $T = \{3, 4, 6, 8, 9, 13, 14\}$ then $x_1 = 3$, $x_k = 14$, $LN(x_1) = \{1, 2\}$, $RN(x_k) = \{15\}$, $C_0 = \{3, 4\}$ and $C_1 = \{13, 14\}$.

In the following lemma we prove that S is a Steiner set.

Lemma 7. *The set S is a Steiner set.*

Proof. Let P' is a shortest path between s and t in G'. Therefore, if x_p is a member of T which is not a member of P' such that $s < x_p < t$ then there always exist two adjacent vertices u and v of P' with $u < x_p < v$. By Lemma 2, x_p is connected with at least one vertex of u and v as u and v are connected. Therefore, each vertex x_p of T with $s < x_p < t$ is connected with at least one vertex of P'. Now s in G' is connected with v_1 of P'. Therefore v_1 is connected with s in G. Since s of G is in C_0 and C_1 is connected, it follows that $C_0 \cup P$ is a connected subgraph in G. Similarly, $C_1 \cup P$ is also a connected subgraph in G. Hence $C_0 \cup P \cup C_1$ is a connected subgraph of G. Again each member of T is either a vertex of one of the subgraphs C_0, P and C_1 or it is connected with some member of V_P. Hence the subgraph $T \cup V_P$ of G is connected. Now $S = V_P - T$ implies $T \cup V_P = T \cup S$. So, S is a Steiner set. Hence the lemma. □

Next we show that the Steiner set S contains minimum number of vertices.

Lemma 8. *The Steiner set S is minimum Steiner set.*

Proof. By Lemma 7, $S = V_P - T$ is a Steiner set. As V_P contains minimum number of vertices, cardinality of V_P is minimum. Since $S = V_P - T$ and T is fixed. Therefore, S is minimum. Hence the lemma. □

It may be noted that the subgraphs C_0 and C_1 are not necessarily tree, they may contain cycle. If they are not tree let T_0 and T_1 be the spanning trees corresponding to the subgraphs C_0 and C_1 respectively. It is shown in the following lemma that $T_0 \cup P \cup T_1$ is a tree.

Lemma 9. *The subgraph $T_0 \cup P \cup T_1$ is a tree.*

Proof. Let T_0 and T_1 be the spanning trees of C_0 and C_1. Let the number of vertices of T_0, T_1 and P' be respectively p, q and r. The path P' is $s \to v_1 \to v_2 \to \cdots \to v_{r-2} \to t$. If v_1 is connected with s in G', then v_1 is connected with s in G. The path $v_1 \to v_2 \to \cdots \to v_{r-2}$ is P. Hence the tree T_0 is connected with the path P which again connected with the tree T_1 *i.e.*, $T_0 \cup P \cup T_1$ is connected. If T_0 and P are connected by more then one edge we consider only one such edge. Similarly, we consider only one such edge connecting P and T_1. Clearly, the set of vertices of T_0, T_1 and P are mutually disjoint. Therefore the number of vertices of $T_0 \cup P \cup T_1$ is $p + (r-2) + q$ and the number of edges in it is $(p-1) + (r-2-1) + (q-1) + 1 + 1$ *i.e.*, $(p + (r-2) + q) - 1$. Since $T_0 \cup P \cup T_1$ is connected and its number of edges is one less then its number of vertices, it is a tree. Hence the lemma. □

If all the vertices of T are member of $T_0 \cup P \cup T_1$, then $T_0 \cup P \cup T_1 = S \cup T$. So, the connected subgraph induced by $S \cup T$ is a Steiner tree.

If $T \not\subset T_0 \cup P \cup T_1$, then let $R = T-$(vertices of $(T_0 \cup P \cup T_1)$). By lemma 2, each vertex of R is connected to at least one vertex of P. Now we construct a subgraph $T_{P\cup R} = (V_{P\cup R}, E_{P\cup R})$ as follows.

The vertex set $V_{P\cup R}$ is taken as $V_P \cup R$. For each vertex $v \in R$, we find two consecutive vertices u and w of P such that $u < v < w$. Then by Lemma 2, at least one of (u, v) and $(u, v) \in E$. For each $v \in R$, we add one of these edges with E_P to form $E_{P\cup R}$.

From the construction of $T_{P\cup R}$ it is obvious that $T_{P\cup R}$ is a tree with minimum number of vertices $|V_{P\cup R}| = |P| + |R|$ and the number of edges $|E_{P\cup R}| = |E_P| + |R| = |P| - 1 + |R|$.

Thus if $S \cup T$=(vertices of $(T_0 \cup P \cup T_1)$), then the Steiner tree is $T^* = (T_0 \cup P \cup T_1)$, otherwise it is $T^* = (T_0 \cup T_{P\cup R} \cup T_1)$.

To find the shortest path between two vertices s and t containing maximum number of target vertices on G', the graph G' is converted to a digraph $\overrightarrow{G}''$. Then applying the algorithm of Ahuja et al. [1] the shortest path P' is to be determined. The conversion method is described below.

3. Shortest Distance Between two Given Vertices Through Some Specified Vertices

Let $G = (V, E)$ be an undirected graph and $T = \{x_1, x_2, \ldots, x_k\}$ be the set of specified vertices through which the shortest path is to be determined. This problem is solve into two stages stated below:

(i) Convert the undirected graph into a directed graph,

(ii) Convert the vertex weight to edge weight.

3.1 Conversion of undirected graph to directed graph

A relation R is said to be symmetric relation if any two elements x_i and x_j, $x_i R x_j$ holds then also $x_j R x_i$ holds. Using symmetric relation, undirected graph $G' = (V', E')$ is transformed to a directed graph $\overrightarrow{G}' = (V', \overrightarrow{E}')$. The edge set $\overrightarrow{E}'$ is constructed as follows: If $(u, v) \in E'$ then the order pairs (u, v) and (v, u)

are both the edges of $\overrightarrow{E}'$. Thus every undirected graph is a representation of some symmetric binary relation (on the set of its vertices). Furthermore, every undirected graph with m edges can be through of as a symmetric digraph with $2m$ directed edges. Figure 4 represents to a digraph $\overrightarrow{G}'$ corresponding to an undirected trapezoid graph G' of Figure 3.

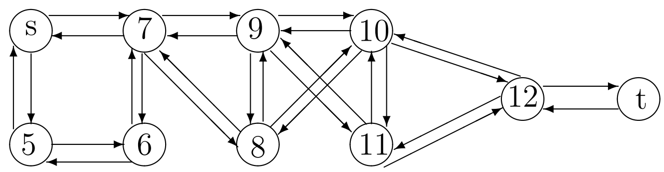


Figure 4: The digraph $\overrightarrow{G'}$ corresponding to the graph G'.

3.2 Conversion of vertex weight to edge weight

Suppose the graph $\overrightarrow{G}' = (V', \overrightarrow{E}')$ is weighted and the weights are assigned to the edges. Let $w(u, v)$ be the weight of the edge (u, v). Here we assume that the weight of each edge and each specified vertex of the corresponding directed graph are *unit* and each non-specified vertex is a large number, say, M. Let $w(i)$ be the weight of the vertex i. Therefore, $w(i)$=1, for $i \in T$ and $w(i)$=M, for $i \in V' - T$.

A network can be transformed into a network by replacing each vertex i by two vertices i' and i'' and introducing a new directed edge (i', i''). All edges previously incident on i are made to be incident on i' and all edges previously incident out of i are made to be incident of i''. This process is briefly illustrated in Figure 5. Again the digraph $\overrightarrow{G}'$ is transformed to a digraph $\overrightarrow{G}''$ by splitting all the vertices using the following method. Replacing each vertex i by two vertices i' and i'' and introducing a new directed edge (i', i''). Therefore, weight of each vertex i is transformed to the weight of edge (i', i''). The graph $\overrightarrow{G}''$ is a edge weighted directed graph of $\overrightarrow{G}'$. The basic difference between the digraph $\overrightarrow{G}'$ and $\overrightarrow{G}''$ is that a non-negative integer weight $w(i)$ is assigned to each vertex i in $\overrightarrow{G}'$ but there is no weight assigned to any vertex in $\overrightarrow{G}''$. The weight of each vertex i of $\overrightarrow{G}'$ has been converted to the weight of each edge

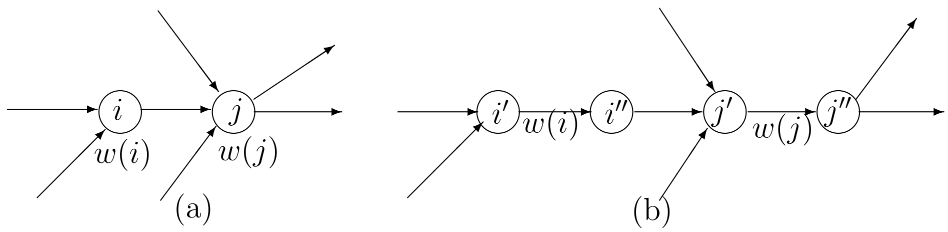


Figure 5: The replacement process of vertices i and j by the vertices i', i'', j' and j''.

(i', i'') of $\overrightarrow{G}''$. Figure 6 represents the edge weighted digraph $\overrightarrow{G}''$ corresponding to the digraph $\overrightarrow{G}'$ of Figure 4.

3.3 Weight of the edges

In Section 3.2, we construct a network, where weight of each vertex i is transformed to weight of edge (i', i''). Let $w(i, j)$ be the weight of the directed edge (i', i'') of the directed graph. Already, we assume that the weight of each edge of the digraph be 1. For simplicity, $w(i, j) = w(j, i) = 1$, for all i and j. Therefore, finally the digraph $\overrightarrow{G}''$ is the edge weighted digraph of $\overrightarrow{G}'$.
For the graph $\overrightarrow{G}'$ of Figure 4, let the set of specified vertices be $\{s, 6, 8, 9, t\}$. Then we put the weight of the vertices be $w(s) = 1$,$w(5) = M$, $w(6) = 1$, $w(7) = M$, $w(8) = 1$, $w(9) = 1$, $w(10) = M$, $w(11) = M$, $w(12) = M$ and $w(t) = 1$ and the weight of each edges be *unit*. The weight of the edges of the graph $\overrightarrow{G}''$ of Figure 6 are then become $w(5', 5'') = w(7', 7'') = w(10', 10'') = w(11', 11'') = w(12', 12'') = M$, $w(s', s'') = w(6', 6'') = w(8', 8'') = w(9', 9'') = w(t', t'') = 1$ and all other edges be *unit*.

Now, finding the shortest path between two vertices s and t through some specified (target vertices) vertices on G' is equivalent to finding of shortest path between two vertices s' and t'' on $\overrightarrow{G}''$. The shortest path P' between s' and t'' can be obtained by using the algorithm of Ahuja et al. [1].

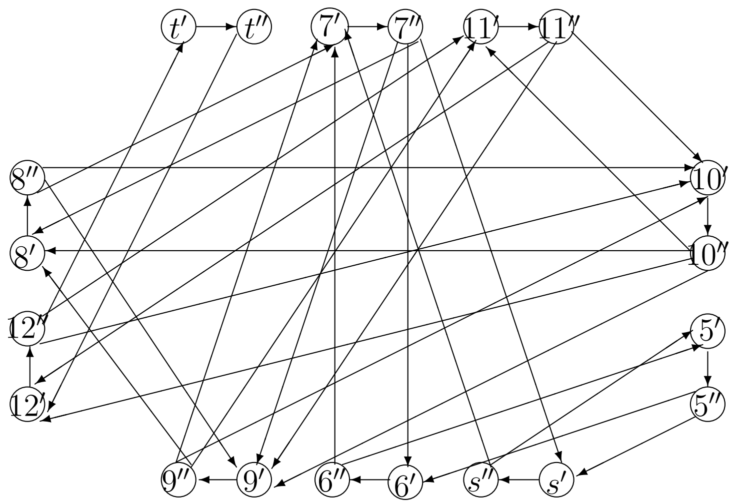


Figure 6: The edge weighted digraph $\overrightarrow{G''}$ corresponding to the digraph $\overrightarrow{G'}$.

4. Algorithm and its Complexity

To compute the Steiner set and Steiner tree we follow the following algorithm. The main steps of the algorithm are listed in Algorithm TSST.

ALGORITHM TSST

Input: A trapezoid graph G with trapezoid representation $T_i(a_i, b_i, c_i, d_i)$, $i = 1, 2, \ldots, n$ and a set of target vertices $T = \{x_1, x_2, \ldots, x_k\}$.

Output: The Steiner set S and Steiner tree T^*.

Step 1: Compute the vertex set $LN(x_1)$ and $RN(x_k)$.

Step 2: Construct an auxiliary graph $G' = (V', E')$, where $V' = V - LN(x_1) - RN(x_k)$ and $E' \subseteq E$.

Step 3: Compute the subgraph C_0 and C_1 of G.

Step 4: Compute the spanning trees T_0 and T_1 of C_0 and C_1 respectively.

Step 5: Construct two fictitious trapezoids T_s and T_t.

Step 6: Compute a shortest path $P' = (V_{P'}, E_{P'})$ from s' to t'' containing maximum number of target vertices on $\overrightarrow{G}''$.

Step 7: Compute $P = (V_P, E_P)$ from P'.

Step 8: Compute the Steiner set $S = V_P - T$.

Step 9: Compute the set $R = T -$(vertices of $T_0 \cup P \cup T_1$)).

Step 10: If $T \not\subset$ (vertices of $(T_0 \cup P \cup T_1)$) then compute $T_{P\cup R}$.

Step 11: If $T \subset$ (vertices of $(T_0 \cup P \cup T_1)$) then
$T^* = (T_0 \cup P \cup T_1)$
else
$T^* = (T_0 \cup T_{P\cup R} \cup T_1)$.

end TSST.

For the graph of Figure 1, the Seiner set $S = \{7, 10, 12\}$ or $\{7, 11, 12\}$ if we consider the set of target vertices $T = \{3, 4, 6, 8, 9, 13, 14\}$.

Theorem 1. *The minimum cardinality Steiner tree of a trapezoid graph with n vertices and m edges can be computed in $O(m + n\sqrt{\log C})$ time, where cost of each arc is a non-negative integer number bounded by C.*

Proof. The vertex set $LN(x_1)$ and $RN(x_k)$ con be computed in $O(n)$ time (Step 1). Construction of the auxiliary graph G' takes only $O(1)$ time (Step 2). The connected subgraph C_0 and C_1 of T can be computed in $O(n)$ time (Step 3). The spanning trees T_0 and T_1 can be computed in $O(n)$ time (Step 4). The fictitious trapezoids T_s and T_t can be computed in $O(n)$ time (Step 5). The shortest path between s' and t'' in $\overrightarrow{G}''$ can be obtained in Section 3. The shortest path between two vertices of a general graph is constructed in $O(m + n\sqrt{\log C})$ time [1]. Therefore, Step 6 takes by $O(m + n\sqrt{\log C})$ time. Step 8 can be computed in $O(n)$ time. Step 9 can be computed in $O(n)$ time. Inclusion of a set into another set can be checked in $O(n)$ time. Thus, Steps 10 and 11 can be computed in $O(n)$ time. Hence overall time complexity is $O(m + n\sqrt{\log C})$. Hence the theorem. □

References

[1] R. K. Ahuja, K. Mehlhorn, J. B. Orlin and R. E. Tarjan, Faster algorithm for the shortest path problem, *J. ACM*, **37** (1990) 213-223.

[2] D. Bera, M. Pal and T. K. Pal, An efficient algorithm to generate all maximal cliques on trapezoid graphs, *Intern. J. Computer Math.*, **79 (10)**(2002) 1057-1065.

[3] C. J. Colbourn and L. K. Stewant, Permutation graphs: connected domination and Steiner trees, *Discrete Math.*, **86** (1990) 145-164.

[4] D. G. Corneil and P. A. Kamula, Extension of permutation and interval graphs, *In: Proc. 18th Southeast Conf. on Combinatorics, Graph Theory and Computing, Congr. Numer*, (1987) 267-276.

[5] I. Dagan, M. C. Golumbic and R. Y. Pinter, Trapezoid graphs and their coloring, *Discrete Applied Mathematics*, **21** (1988) 35-46.

[6] A. D'Atri, and M. Mascarini, Distance hereditary graphs, Steiner trees and connected domination, *SIAM J. Comput.*, **17** (1988) 521-538.

[7] D. E. Drake, S. Hougardy, On approximation algorithms for the terminal Steiner tree problem, *Information Processing Letters*, **84** (2004) 15-18.

[8] P. K. Ghosh and M. Pal, An efficient algorithm to find the maximum matching on trapezoid graphs, to appear in $KSIAM$.

[9] M. C. Golumbic, *Algorithmic graph theory and prefect graphs*, Academic Press, New York, 1980.

[10] S. T. Hedetniemi, and R. C. Laskar (Eds.), Topics on domination, *Annals of Discrete Mathematics, North-Holland, Amsterdam*, **48** (1991) 145-164.

[11] S. Khuller and A. Zhu., The general Steiner tree-star problem, *Information Processing Letters*, **84** (2002) 215-220.

[12] Y. D. Liang, Dominations in trapezoid graphs, *Information Processing Letters*, **52** (1994) 309-315.

[13] T. Ma and J. P. Spinrad, An $O(n^2)$ algorithm for 2-chain problem on certain classes of perfect graphs, *In: Proc. 2nd ACM-SIAM Symp. on Discrete Algorithms*, 1991.

[14] S. Mondal, M. Pal and T. K. Pal, An optimal algorithm for solving all-pairs shortest paths on trapezoid graphs, *International Journal of Computational Engineering Science*, **3 (2)**(2002) 103-116.

[15] S. Mondal, M. Pal and T. K. Pal, Optimal sequential and parallel algorithms to compute a Steiner tree on permutation graphs, *International Journal of Computer Math.*, **80 (8)**(2003) 939-945.

[16] M. Pal and G. P. Bhattacharjee, A linear algorithm for k-connected Steiner subgraph problem on an interval graph, *Maeer's MIT Pune Journal*, **III (11)**(1994) 50-52.

[17] H. J. Promel, A new approximation algorithm for the Steiner tree problem with performance ratio 5/3, *Journal of Algorithms*, **36** (2000) 89-101.

[18] Y. L. White, H. C. Chen and C. Y. Lee, An $O(\log n)$ parallel algorithm for constructing a spanning tree on permutation graphs, *Information Processing Letters*, **56** (1995) 83-87.