An improved 2-approximation algorithm for the rectilinear Steiner tree problem with minimal number of Steiner points

Jianglin LI[a], Wencheng WANG[a,*], Yinhua CHEN[a], Tongquan ZHANG[b]
[a]School of Mathematics and Computer Science, Yunnan Minzu University, Kunming, 650504, Yunnan, China
[b]School of Applied Technology, Yunnan Minzu University, Kunming, 650504, Yunnan, China
[*]Corresponding author's email: wencheng@ymu.edu.cn

supported by the Youth Project of the Department of Science and Technology of Yunnan Province (Grant No. 202401AU070027).

Introduction

RSTP-MSP problem: Given a set X of n terminal points in the L1-plane and a positive constant L, it is asked to construct a Steiner tree T interconnecting of these n terminal points, with the length of each edge in that tree T is no more than L.

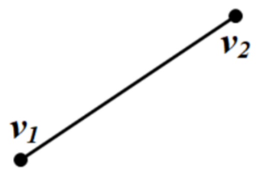


Figure 1. The shortest path on the Euclidean plane

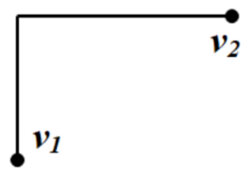


Figure 2. The shortest path on the L1-plane

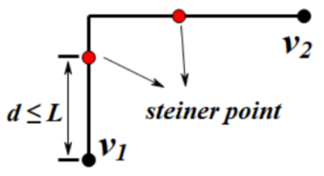
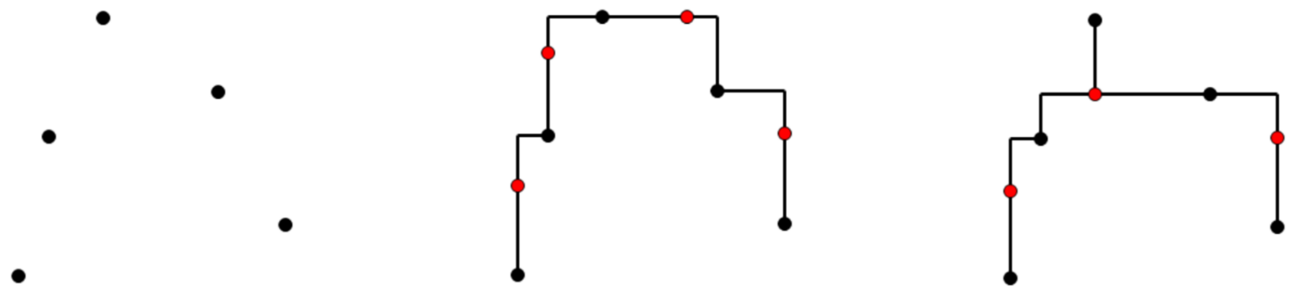


Figure 3. Steinerized of edges with a length greater than L[1]



(a) The terminal set X in the L1 plane **(b)** Steinerized minimum spanning tree **(c)** The optimal Steiner tree for X

Figure 4. The optimal Steiner tree for a given set of points X in the L1-plane

Research objectives

- Reduce time complexity by technique of leaf-deletion.

Algorithm

Strategy

(1) Using the sweep-line algorithm [2] to construct a rectilinear minimum spanning tree T on the set X, and sorting all the edges in tree T in non-decreasing order of their length.

(2) Construct a forest $F_a \subseteq T$ with the length of each edge e in Fa satisfying $L < w(e) \leq 2L$.

(3) Using the leaf-deletion technique to construct a rectilinear Steiner Tree T_s such that the length of each edge in T_s is no more than L.

Algorithm : $\mathcal{A}_1$

Input: A set $X = \{v_1, v_2, \cdots, v_n\}$ of n terminal points on the L_1 plane and a positive constant L.

Output: A rectilinear Steiner Tree T_s.

1: Use the sweep-line algorithm to obtain the minimum spanning tree T_{MST}, and sort all edges in T_{MST} according to their length and weight.

2: Add edges in T_{MST} with a side length equal to or less than L to T_s, where $T_s = (X_s, E_s)$, $X_s = X, E_s = \varnothing$.

3: Place the edges in T_{MST} with a length greater than L but less than or equal to $2L$ into forest F, where $F = \varnothing$. We construct the Samsung Steiner point by selecting vertex u_1 with a degree of 1 in F, checking the terminal point u_0 connected to it, and then checking the edges connected to U_0. After checking all the points, place the unchanged edges and newly generated points and edges into T_s.

4: Steinerized the edges in T_{MST} with a length greater than $2L$ and place them in T_s

5: **return** A rectilinear Steiner tree T_s.

Figure 5. Algorithm $\mathcal{A}_1$

Complexity analysis

Time complexity

Clearly, for step 1 of Algorithm $\mathcal{A}_1$, the construction of the rectilinear minimum spanning tree T using the sweep-line algorithm[2] and sorting all the edges in tree T in non-decreasing order of their length can be implemented in time *O(nlogn)*.

Obviously, steps 2 and 4 in Algorithm $\mathcal{A}_1$ have a time complexity of at most *O(n)*. In the step 3 of Algorithm $\mathcal{A}_1$, the leaf-deletion technique takes *O(1)* time for each leaf vertex, and we need to check at most n leaf vertices, then the execution of the final while loops of Algorithm $\mathcal{A}_1$ can also be implemented within *O(n)* time.

In summary, we can obtain Algorithm $\mathcal{A}_1$ can be implemented in time *O(nlogn)*.

Algorithm complexity

Du et al.[3] used the method of constructing a "3-star" using a complete graph to obtain a Steiner tree algorithm with an approximate ratio of 2. We also obtained a 2-approximation algorithm $\mathcal{A}_1$ using technique of leaf-deletion.

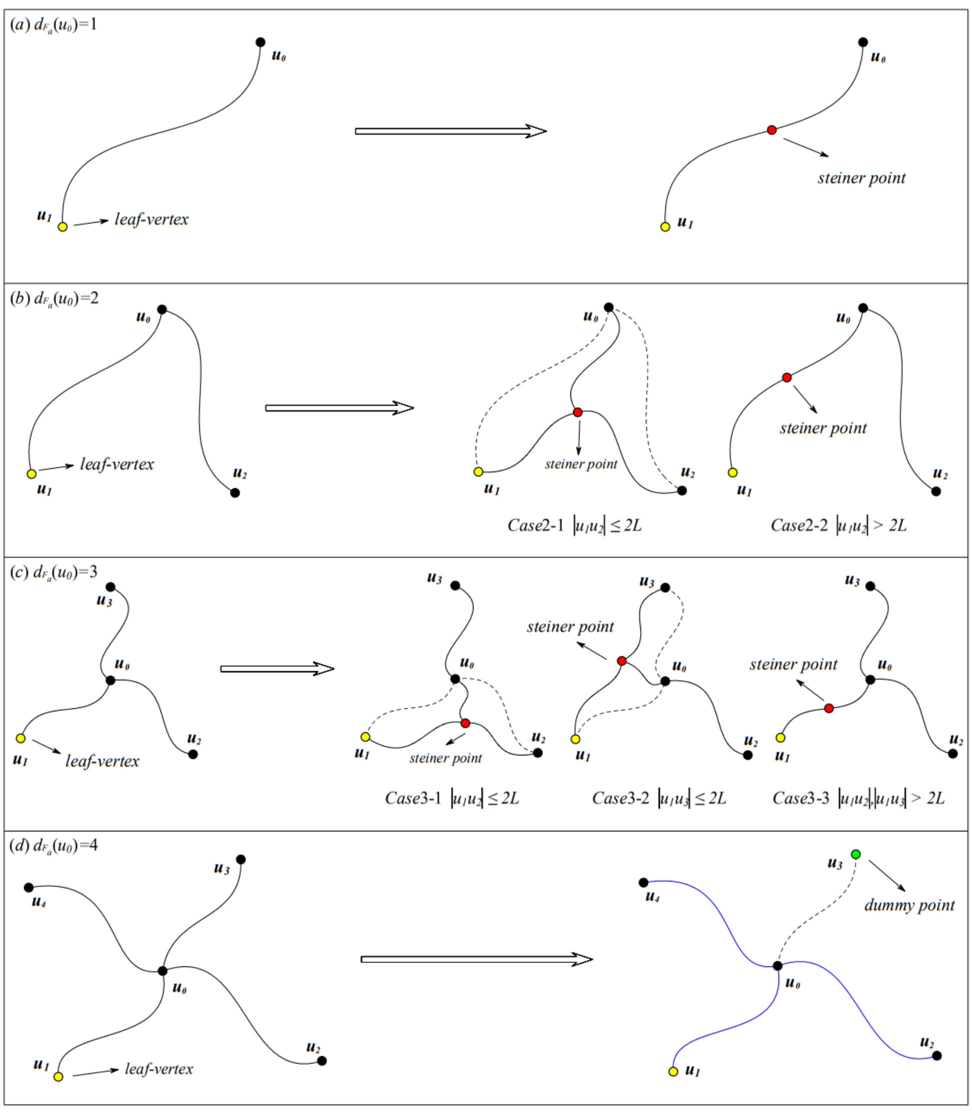


Figure 6. The technique of leaf-deletion

Conclusions

- The leaf deletion technique has been proposed, and it can construct a "3-star" within *O(n)* time.
- A 2-approximation algorithm $\mathcal{A}_1$ has been proposed, which can solve the RSTP-MSP problem in *O(nlogn)* time.

Acknowledgement

Thanks for the support by The Youth Project of the Department of Science and Technology of Yunnan Province (Grant No. 202401AU070027).

References

[1] Lu B, Gu J, Hu X, et al. Wire segmenting for buffer insertion based on RSTP-MSP[J]. Theoretical computer science, 2001, 262(1-2): 257-267.
[2] Zhou H, Shenoy N, Nicholls W. Efficient minimum spanning tree construction without Delaunay triangulation [C] //Proceedings of the 2001 Asia and South Pacific Design Automation Conference. 2001:192-197.
[3] Du D, Hu X. Steiner tree problems in computer communication networks[M]. World Scientific, 2008.