IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 July 2026, accepted 7 August 2026, date of publication 11 August 2026, date of current version 17 August 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3722514

RESEARCH ARTICLE

A Novel Approximate Solution Method for Euclidean Steiner Tree Problem Based on Genetic Algorithm

LIPING ZHANG [1,2], TSUYOSHI MIGITA [3], AND NORIKAZU TAKAHASHI [3], (Member, IEEE)

[1]Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan
[2]School of Automobile and Rail Transit, Luoyang Polytechnic, Luoyang 471000, China
[3]Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan

Corresponding author: Norikazu Takahashi (takahashi@okayama-u.ac.jp)

This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP25K03196.

ABSTRACT A novel approximate solution method for the Euclidean Steiner tree problem is proposed and its performance is demonstrated through experiments using 195 benchmark problem instances from the OR-Library. Given a finite number of terminal points, the proposed method first generates non-terminal points around each terminal point and at the same location as the Steiner point for each triangle obtained by the Delaunay triangulation for all terminal points. It next runs a genetic algorithm to select a subset of the generated non-terminal points, aiming to minimize the total edge length of a minimum spanning tree for all the terminal points and the selected non-terminal points. It then optimizes the locations of the non-terminal points in the tree using Weiszfeld's method, while preserving the topology. It finally refines the tree by adding new non-terminal points, adding and removing edges, and optimizing the locations of all non-terminal points. The experimental results show that, among 150 benchmark problem instances with known optimal solutions, the proposed method successfully constructs a Euclidean Steiner tree for 61 instances. For the remaining 89 instances, it produces an approximate solution whose total edge lengths is less than 100.7% of the optimal. The experimental results also show that the proposed method obtains approximate solutions efficiently: within 1.5 seconds for instances with up to 100 terminal points and within 61 seconds for instances with up to 1,000 terminal points.

INDEX TERMS Combinatorial optimization, genetic algorithm, Fermat problem, Weiszfeld's method, Delaunay triangulation.

I. INTRODUCTION

The Euclidean Steiner Tree Problem (ESTP) [1] has been a significant challenge in the fields of combinatorial optimization and computational geometry. Given a finite set of terminal points in the two-dimensional Euclidean space, the goal of the ESTP is to construct a spanning tree that connects all terminal points, possibly incorporating additional non-terminal points known as the Steiner points, in such a way that the total edge length is minimized. Since shortest path problems encountered in various domains – such as electrical circuits, computer networks, telecommunication systems, power grids, and transportation systems – can be formulated as instances of the ESTP, it has been the focus of extensive research over the past several decades [2].

One of the most challenging aspects of the ESTP lies in determining both the number and locations of the Steiner points. However, it has been proved that the ESTP is NP-hard [3]. As a consequence, a large number of research efforts have been dedicated to devising heuristic algorithms and approximation methods for solving the ESTP [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Smith et al. [4] presented a heuristic algorithm based on a decomposition approach using the Delaunay triangulation [16], the Voronoi diagram, and the Minimum Spanning Tree (MST). Fampa and Anstreicher [5] proposed some techniques to address

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

several issues in Smith's branch-and-bound algorithm and demonstrated experimentally that they significantly improve performance. Van Laarhoven and Ohlmann [6] presented a heuristic that utilizes the Delaunay triangulation to generate candidate Steiner points for insertion, the MST to identify Steiner points to remove, and second-order cone programming to optimize the location of the remaining Steiner points. Hsu et al. [10] designed a novel biologically-inspired explore-and-fuse approach. Bereta [11] concerned with the Baldwin effect and Lamarckian evolution in a memetic algorithm. Pinto and Maculan [12] presented two heuristics for the ESTP in higher dimensions and showed that those heuristics return relatively good solutions. Cao et al. [14] proposed a binary particle swarm optimization algorithm for the ESTP, and applied it to the problem of establishing a communication network connecting 144 cities in China. Bereta [15] presented a genetic algorithm for the ESTP and pointed out that significant improvement can be achieved through proper initialization of the initial population, which depends on the appropriate sorting of vertices.

Recently, Zhang et al. [13], the authors of the present paper, proposed an approximate solution method for the ESTP based on a genetic algorithm. This method first generates non-terminal points around each terminal point using a simple rule, next runs a genetic algorithm to select a subset of the generated non-terminal points, aiming to minimize the total edge length of an MST for all terminal points and the selected non-terminal points, and finally optimizes the locations of the non-terminal points in the MST using Weiszfeld's method [17], [18], while preserving the topology. Experimental results show that this simple method can find good approximate solutions, but they tend to have fewer non-terminal points than the corresponding Euclidean Steiner Trees (ESTs).

Although the ESTP is NP-hard, a number of researchers have investigated exact algorithms over the past several decades [19], [20], [21], [22], [23], [24]. Winter [19] proposed an exact algorithm called GeoSteiner, which is considerably faster than earlier exact algorithms. Winter and Zachariasen [20] designed an exact algorithm which is based on but much faster than GeoSteiner. Van Laarhoven and Anstreicher [21] presented geometric conditions that can be used to eliminate candidate topologies for the ESTs in d-dimensional Euclidean space with $d \geq 2$, which is an extension of the results of Winter and Zachariasen [20]. Juhl et al. [22] described algorithmic enhancements that improves the performance of the GeoSteiner software package, and presented a computational study performed on the largest problem instances from the 2000-study. Brazil et al. [23] introduced an exact algorithm for the Euclidean k-Steiner tree problem, which minimizes the total edge length of the tree connecting given terminal points and at most k additional Steiner points, allowing them to have degree four. They extended the basic framework of GeoSteiner's generation algorithm so that degree-four Steiner points are allowed, derived some new results restricting the structural and geometric properties of optimal k-Steiner trees, and developed efficient topological pruning methods based on those results. Lee et al. [24] designed a novel pruning test to allow more extensive elimination of sub-optimal topologies during the generation phase of the exact algorithm for the Euclidean k-Steiner tree problem [23].

As described above, substantial progress has been achieved in the development and acceleration of exact algorithms for the ESTP, owing to the dedicated efforts of many researchers. Nevertheless, due to the NP-hard nature of the problem, such acceleration is inherently limited. Therefore, further enhancement of both the efficiency and accuracy of approximation algorithms remains a critical area of ongoing research.

In this paper, we propose a novel approximate solution method for the ESTP, which builds upon the authors' previous work [13] but newly incorporates techniques such as an exact algorithm for three terminal points and the Delaunay triangulation. The proposed method first generates non-terminal points not only around each terminal point but also at the same location as the Steiner point for each triangle obtained by the Delaunay triangulation for all terminal points. It next runs a genetic algorithm to select a subset of the generated non-terminal points, and optimizes the locations of the selected non-terminal points, aiming to minimize the total edge length of an MST for all the terminal points and the selected non-terminal points, in the same way as the previous work. It finally repeats the following two procedures: i) selecting two edges that share a terminal point and generating a new non-terminal point at the same location as the Steiner point for the three endpoints of the two edges, ii) removing the two edges and adding three edges connecting the new non-terminal point to the three endpoints. This step is introduced to overcome the weakness of the previous work described above.

The effectiveness of the proposed method is evaluated through experiments conducted on 195 problem instances from the OR-Library [25]. The experimental results show that, among the 150 instances with known optimal solutions, the proposed method constructs ESTs in 61 instances, approximate solutions whose total edge lengths are less than 100.1% of the optimal in 83 instances, and approximate solutions whose total edge lengths are between 100.1% and 100.7% of the optimal in the remaining six instances. This indicates the proposed method outperforms the authors' previous method and an existing evolutionary algorithm [11]. The experimental results also show that the proposed method obtains approximate solutions efficiently: within 1.5 seconds for instances with up to 100 terminal points and within 61 seconds for instances with up to 1,000 terminal points.

II. EUCLIDEAN STEINER TREE PROBLEM AND SOME KNOWN RESULTS

A. PROBLEM STATEMENT

Let $G = (V, E)$ be an undirected, connected, and weighted graph. If a subgraph of G is a tree that includes all vertices

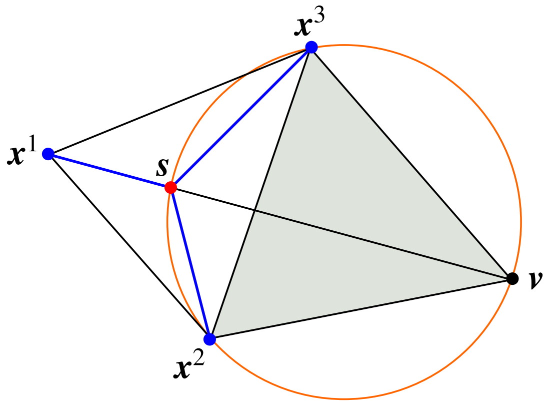


FIGURE 1. How to find the unique Steiner point for three terminal points.

in G then it is called a spanning tree of G. If the sum of the weights on all edges in a spanning tree of G is not greater than any other spanning trees of G then it is called a Minimum Spanning Tree (MST) of G.

Suppose we are given a finite set of terminal points $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n\} \subset \mathbb{R}^2$. A Euclidean MST of X is an MST of the complete graph K_n of which the vertex set is X and the weight $w_{ij} (= w_{ji})$ on the edge $\{\boldsymbol{x}^i, \boldsymbol{x}^j\}$ is defined by the Euclidean distance between $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$, that is, $w_{ij} = \|\boldsymbol{x}^i - \boldsymbol{x}^j\|$. The Euclidean Steiner Tree Problem (ESTP) is to find a finite set $S = \{\boldsymbol{s}^1, \boldsymbol{s}^2, \dots, \boldsymbol{s}^{n'}\} \subset \mathbb{R}^2$ of non-terminal points such that the sum of the weights of all edges (or the total edge length) of a Euclidean MST of $X \cup S$ is minimized. Such a tree is called a Euclidean Steiner Tree (EST), and the non-terminal points in S are called the Steiner points.

It is well known that every EST has the following properties (see Gilbert and Pollak [1] for details).

1) The angle between any two adjacent edges is not less than $2\pi/3$ radians.
2) No more than three edges are incident to any vertex.
3) The number of edges incident to any Steiner point is three, and the angle between them is $2\pi/3$ radians.
4) The number of Steiner points is not greater than the number of terminal points minus two.
5) All Steiner points are included in the convex hull of the terminal points.

B. STEINER POINT FOR THREE TERMINAL POINTS

The ESTP with $n = 3$ is widely known as the Fermat problem, and has been completely solved by many authors [1], [2], [26], [27]. One of their solution methods is illustrated in Fig. 1. We explain briefly how to find the unique Steiner point $\boldsymbol{s}$ (if it exists) using this method.

Let $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3\}$ be a given set of three terminal points which are distinct from each other. We first check whether the triangle formed by $\boldsymbol{x}^1$, $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$ has an internal angle greater than or equal to $2\pi/3$ radians, that is, there exist i, j and k such that $\{i, j, k\} = \{1, 2, 3\}$ and

$$\frac{(\boldsymbol{x}^j - \boldsymbol{x}^i)^\top (\boldsymbol{x}^k - \boldsymbol{x}^i)}{\|\boldsymbol{x}^j - \boldsymbol{x}^i\| \|\boldsymbol{x}^k - \boldsymbol{x}^i\|} \leq -\frac{1}{2}. \tag{1}$$

Algorithm 1 Three-Point Algorithm

Input: $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3\} \subset \mathbb{R}^2$ where $\boldsymbol{x}^1$, $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$ are distinct from each other.
Output: Steiner point $\boldsymbol{s} \in \mathbb{R}^2$ or "None"
1: If there exist i, j and k such that $\{i, j, k\} = \{1, 2, 3\}$ and (1) is satisfied then return "None" and stop.
2: Set $\boldsymbol{p} \leftarrow \boldsymbol{R}(\pi/2)(\boldsymbol{x}^3 - \boldsymbol{x}^2)$ and $\boldsymbol{q} \leftarrow \boldsymbol{R}(\pi/3)(\boldsymbol{x}^3 - \boldsymbol{x}^2)$ where $\boldsymbol{R}(\theta)$ is the rotation matrix defined by (2).
3: If $\boldsymbol{p}^\top(\boldsymbol{x}^1 - \boldsymbol{x}^2)$ and $\boldsymbol{p}^\top \boldsymbol{q}$ have different signs then set $\boldsymbol{v} \leftarrow \boldsymbol{x}^2 + \boldsymbol{q}$. Otherwise set $\boldsymbol{v} \leftarrow \boldsymbol{x}^2 + \boldsymbol{R}(-\pi/3)(\boldsymbol{x}^3 - \boldsymbol{x}^2)$.
4: Set $\boldsymbol{s} \leftarrow \boldsymbol{v} + t(\boldsymbol{x}^1 - \boldsymbol{v})$ where t is given by (3).
5: Return $\boldsymbol{s}$ and stop.

If this is the case, there is no Steiner point. Otherwise, we find a point $\boldsymbol{v}$ such that i) $\boldsymbol{x}^2, \boldsymbol{x}^3$ and $\boldsymbol{v}$ form an equilateral triangle and ii) the line segment connecting $\boldsymbol{v}$ and $\boldsymbol{x}^1$ intersects the line passing through $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$. By the first condition, $\boldsymbol{v}$ must be either $\boldsymbol{v}^1 = \boldsymbol{x}^2 + \boldsymbol{R}(\pi/3)(\boldsymbol{x}^3 - \boldsymbol{x}^2)$ or $\boldsymbol{v}^2 = \boldsymbol{x}^2 + \boldsymbol{R}(-\pi/3)(\boldsymbol{x}^3 - \boldsymbol{x}^2)$ where $\boldsymbol{R}(\theta)$ is the rotation matrix defined by

$$\boldsymbol{R}(\theta) := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{2}$$

By the second condition, the inner products $\boldsymbol{p}^\top(\boldsymbol{x}^1 - \boldsymbol{x}^2)$ and $\boldsymbol{p}^\top(\boldsymbol{v} - \boldsymbol{x}^2)$ must have different signs, where $\boldsymbol{p}$ is a normal vector of the line passing through $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$. Using this condition, we choose either $\boldsymbol{v}^1$ or $\boldsymbol{v}^2$ to be $\boldsymbol{v}$. We then draw a circle passing through $\boldsymbol{v}$, $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$, and find the intersection $\boldsymbol{s}\ (\neq \boldsymbol{v})$ of the line segment connecting $\boldsymbol{v}$ and $\boldsymbol{x}^1$ and the circle, which is given by $\boldsymbol{s} = \boldsymbol{v} + t(\boldsymbol{x}^1 - \boldsymbol{v})$ where

$$t = \frac{2(\boldsymbol{x}^1 - \boldsymbol{v})^\top(\boldsymbol{x}^2 + \boldsymbol{x}^3 - 2\boldsymbol{v})}{3\|\boldsymbol{x}^1 - \boldsymbol{v}\|^2}. \tag{3}$$

The above-mentioned procedure to find the unique Steiner point (if it exists) for a given set $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3\}$ of three terminal points is formally stated in Algorithm 1.

C. WEISZFELD'S METHOD

As an extension of the Fermat problem, we consider the geometric median problem, which is also known as the Fermat-Weber problem [28]. Let $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n\} \subset \mathbb{R}^2$ be a given set of terminal points. The Fermat-Weber problem is formulated as follows:

$$\text{minimize} \ \ f(\boldsymbol{y}) := \sum_{i=1}^{n} \|\boldsymbol{y} - \boldsymbol{x}^i\|. \tag{4}$$

An optimal solution of this problem is called a geometric median of X. It is well known that the optimal solution is uniquely determined unless all of the n terminal points lie on the same line [29].

Weiszfeld [17], [18] proposed an iterative method for solving the problem (4), which repeatedly updates the value of $\boldsymbol{y}$ according to the rule:

$$\boldsymbol{y} \leftarrow \frac{1}{\sum_{i=1}^{n} \frac{1}{\|\boldsymbol{y} - \boldsymbol{x}^i\|}} \sum_{i=1}^{n} \frac{\boldsymbol{x}^i}{\|\boldsymbol{y} - \boldsymbol{x}^i\|}.$$

Let the value of $\boldsymbol{y}$ after k updates be denoted as $\boldsymbol{y}[k]$. It has been proved that if $\boldsymbol{y}[k] \notin X$ for all $k \geq 0$ then $f(\boldsymbol{y}[k+1]) \leq f(\boldsymbol{y}[k])$ for all $k \geq 0$ and the equality holds if and only if $\boldsymbol{y}[k]$ is an optimal solution of (4) [17], [28], [30]. It has also been proved that if $\boldsymbol{y}[k] \notin X$ for all $k \geq 0$ then the sequence $\{\boldsymbol{y}[k]\}_{k=0}^{\infty}$ converges to an optimal solution of (4) [31].

Weiszfeld's method plays an important role in the approximate solution method for ESTPs proposed in this paper.

III. APPROXIMATE SOLUTION METHOD FOR EUCLIDEAN STEINER TREE PROBLEM

A. OUTLINE

We propose a novel approximate solution method for solving the ESTPs. The proposed method is based on the authors' previous work [13], but newly incorporates techniques such as the three-point algorithm described in the previous section and the Delaunay triangulation [16], [32]. The Delaunay triangulation of a set of terminal points is defined as the unique triangulation so that the circumsphere of each triangle has no terminal points in its interior [33]. Since it possesses important properties related to generation of non-terminal points, the Delaunay triangulation has been used in many heuristic algorithms for solving the ESTPs [6], [34].

The proposed method consists of four main stages. In the first stage, initial non-terminal points are generated around each terminal point, and at the same locations as the Steiner points for triangles obtained by performing the Delaunay triangulation for the set of all terminal points (see Fig. 2(a)). In the second stage, a subset of the initial non-terminal points are selected using the genetic algorithm in the authors' previous work [13], aiming to minimize the total edge length of a Euclidean MST for all terminal points and the selected non-terminal points (see Fig. 2(b)). In the third stage, the locations of the non-terminal points selected in the second stage are optimized in the same way as in the authors' previous work [13] so that the total edge length of the spanning tree is minimized (see Fig. 2(c)). In the last stage, for each pair of edges sharing a common vertex corresponding to a terminal point, a new non-terminal point is generated at the same location as the Steiner point for the three end points of the two edges to further reduce the total edge length (see Fig. 2(d)). We call this process refinement of a tree.

In what follows, we describe each stage in more detail.

B. GENERATION OF INITIAL NON-TERMINAL POINTS

In the first stage, the proposed method generates initial non-terminal points using two rules. One is the same as in the authors' previous work [13]. For each terminal point $\boldsymbol{x}^i$, the radius r^i is determined first by

$$r^i = C \min_{j \neq i} \|\boldsymbol{x}^i - \boldsymbol{x}^j\| \tag{5}$$

where C is a positive constant less than 0.5, and then q non-terminal points denoted by $\boldsymbol{y}^{q(i-1)+k}$ ($k = 1, 2, \ldots, q$) are generated as follows:

$$\boldsymbol{y}^{q(i-1)+k} = \boldsymbol{x}^i + r^i \boldsymbol{R}\left(\frac{2\pi(k-1)}{q}\right)\boldsymbol{e}_1, \quad k = 1, 2, \ldots, q$$

where $\boldsymbol{R}(\theta)$ is the rotation matrix defined by (2) and $\boldsymbol{e}_1 := (1, 0)^\top$. The other is based on the Delaunay triangulation and the three-point algorithm. First the Delaunay triangulation is performed for the set of all terminal points. Then Algorithm 1 is run for each of the triangles obtained by the Delaunay triangulation. If Algorithm 1 returns the Steiner point $\boldsymbol{s}$ then a non-terminal point is generated at the same location as $\boldsymbol{s}$.

Figure 2(a) shows an example to illustrate how initial non-terminal points are generated in the proposed method. The blue circles represent given terminal points, the blue lines a Euclidean MST of the terminal points, the gray lines the edges of triangles obtained by the Delaunay triangulation, and the red circles the initial non-terminal points generated using the method described above with $C = 0.3$ and $q = 7$.

C. SELECTION OF NON-TERMINAL POINTS

Let $Y = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^m\}$ be the set of all non-terminal points generated in the first stage. In the second stage, the proposed method uses the Genetic Algorithm (GA) developed in the authors' previous work [13] to find a subset Y_1 of Y such that the total edge length of a Euclidean MST for the points in the set $X \cup Y_1$ is as short as possible.

GAs are random search methods inspired by the principles of natural selection and genetic inheritance observed in biological systems [35], [36]. They first initialize a population of candidate solutions. Each candidate solution is represented as a sequence of 0's and 1's, which is called a chromosome. They next evaluate each candidate solution using a fitness function, and update the population of candidate solutions using some genetic operations such as selection, crossover and mutation. This process is repeated until a termination criterion is satisfied.

In the first step of our GA, the initial population of n_c chromosomes of length m is generated randomly. To be more precise, each gene of a chromosome is set to 1 with probability $p_w \in (0, 1)$ and 0 with probability $1 - p_w$, independently of other genes. A chromosome $\boldsymbol{b} = (b_1, b_2, \ldots, b_m) \in \{0, 1\}^m$ is an encoding of a subset Y^* of Y, where $b_i = 1$ indicates that $\boldsymbol{y}^i$ is included in Y^*, and $b_i = 0$ otherwise.

In the second step, the fitness of each chromosome $\boldsymbol{b}$ in the initial population is evaluated. To do so, a Euclidean MST for X, the set of all terminal points, and the set of non-terminal points represented by $\boldsymbol{b}$ is obtained using Prim's algorithm [37]. The fitness of $\boldsymbol{b}$ is then evaluated by the inverse of the total edge length of the tree. However, it is clear that if the Euclidean MST contains a non-terminal point with degree one or two then the total edge length can be reduced by removing the non-terminal point. Therefore, in this case, the fitness of $\boldsymbol{b}$ is evaluated after transforming it into a better chromosome $\boldsymbol{b}^*$. To be more precise, the following two operations are done alternatively until the Euclidean MST

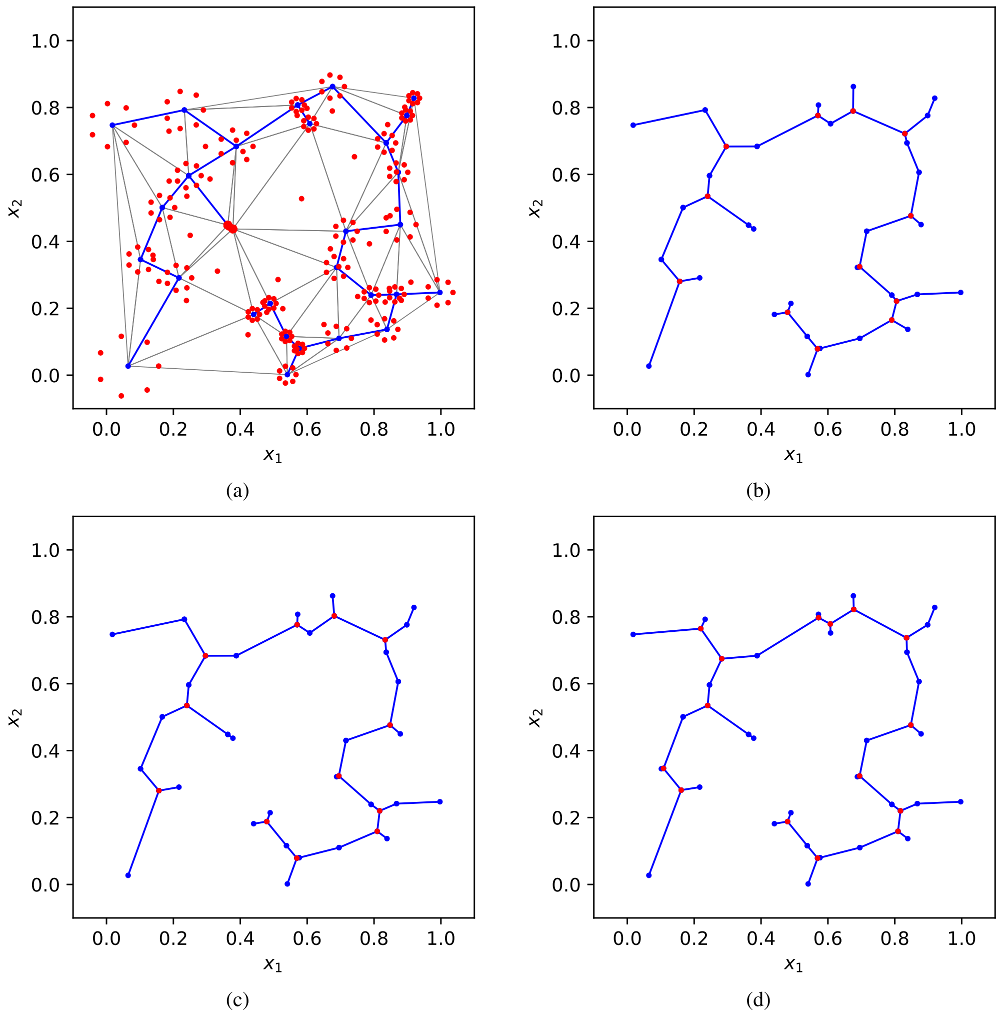


FIGURE 2. Four main stages of the proposed method: (a) generation of non-terminal points, (b) selection of non-terminal points using a genetic algorithm, (c) optimization of the locations of the selected non-terminal points, and (d) refinement of the tree. Blue and red circles represent terminal and non-terminal points, respectively. Blue lines represent the edges of the tree.

does not have such non-terminal points: i) construction of the Euclidean MST and ii) removal of a non-terminal point with degree less than three. The procedure for evaluation of a chromosome is formally stated in Algorithm 2.

In the third step, the population of chromosomes is updated n_g times using three standard genetic operations: selection, crossover and mutation. Let P be the population of chromosomes of the current generation. It should be noted that, strictly speaking, P is a multiset and thus can contain multiple instances of the same chromosome. Let p_s, p_c and p_m be positive numbers satisfying $p_s + p_c + p_m = 1$. Let p_f be a positive number less than one. The population P' of chromosomes of the next generation, which is also a multiset, is obtained as follows. First, P' is initialized as $P' = \{\boldsymbol{b}^{\text{best}}\}$ where $\boldsymbol{b}^{\text{best}}$ is a chromosome in P with the highest fitness value. Second, one genetic operation, randomly selected from selection, crossover and mutation with probabilities p_s, p_c and p_m, respectively, is applied to P, and the resulting one or two chromosomes are added to P'. This process is repeated until $|P'|$ reaches or exceeds n_c where $|P'|$ denotes the cardinality of P'. Then the fitness of each chromosome in P' is evaluated as in the second step. Note that some of the chromosomes may be transformed into better ones during the evaluation.

The three genetic operations used in our GA are as follows. Selection selects one chromosome from P using the tournament selection rule, that is, randomly selects K chromosomes from P with equal probability, and then selects one chromosome with the highest fitness value from the K chromosomes. Crossover selects two chromosomes from P using the tournament selection rule, and then generates two new chromosomes by applying the single-point crossover operation. Mutation selects one chromosome from P using the tournament selection rule and, for each gene of the chromosome, changes its value (from 0 to 1 or from 1 to 0) with probability p_f.

Algorithm 2 Evaluation of Chromosome

Input: $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$, $Y = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^m\} \subset \mathbb{R}^2$ and $\boldsymbol{b} = (b_1, b_2, \ldots, b_m) \in \{0, 1\}^m$

Output: $\boldsymbol{b}^* = (b_1^*, b_2^*, \ldots, b_m^*) \in \{0, 1\}^m$, $T = (X \cup Y^*, E^*)$ and $v \in \mathbb{R}_{++}$

1: Set $\boldsymbol{b}^* \leftarrow \boldsymbol{b}$.
2: Set $Y^* \leftarrow \{\boldsymbol{y}^i \in Y \mid b_i^* = 1\}$.
3: Find a Euclidean minimum spanning tree $T = (X \cup Y^*, E^*)$ of $X \cup Y^*$ using Prim's algorithm.
4: Search for an index $i \in \{1, 2, \ldots, m\}$ such that $b_i^* = 1$ and the degree of $\boldsymbol{y}^i$ in T is less than three. If such an index does not exist, go to Step 6.
5: Set $b_i^* \leftarrow 0$ and go to Step 2.
6: Find the total edge length ℓ of T and set $v \leftarrow 1/\ell$.
7: Return $\boldsymbol{b}^*$, T and v, and then stop.

In the last step of our GA, one chromosome with the highest fitness value is selected from the population of chromosomes of the final generation.

The above-mentioned procedure to select non-terminal points is formally stated in Algorithm 3. Note that the tree returned by Algorithm 3 does not include any non-terminal points of degree one or two, because such non-terminal points are removed during the evaluation process in Step 12.

Figure 2(b) shows the result of the selection of non-terminal points from those in Fig. 2(a). The blue and red circles represent the terminal points and the selected non-terminal points, respectively. The blue lines represent the Euclidean MST of the terminal points and the selected non-terminal points.

D. OPTIMIZATION OF LOCATIONS OF NON-TERMINAL POINTS

Let $T_1 = (X \cup Y_1, E_1)$ be the Euclidean MST corresponding to the chromosome selected by the GA. If T_1 contains non-terminal points, their locations are not optimal in general. So, in the third stage, the proposed method optimizes the locations of all non-terminal points to minimize the total edge length of T_1, while preserving the topology, with the exception of vertex merging.

We assume without loss of generality that $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{m_1}$ are non-terminal points contained in T_1. The proposed method updates the locations of the non-terminal points sequentially using Weiszfeld's method as

$$\boldsymbol{y}^i \leftarrow \frac{\sum_{j \in N_1(i)} \frac{\boldsymbol{x}^j}{\|\boldsymbol{x}^j - \boldsymbol{y}^i\|} + \sum_{j \in N_2(i)} \frac{\boldsymbol{y}^j}{\|\boldsymbol{y}^j - \boldsymbol{y}^i\|}}{\sum_{j \in N_1(i)} \frac{1}{\|\boldsymbol{x}^j - \boldsymbol{y}^i\|} + \sum_{j \in N_2(i)} \frac{1}{\|\boldsymbol{y}^j - \boldsymbol{y}^i\|}} \tag{6}$$

until the movement of all non-terminal points is sufficiently small, where $N_1(i) \subseteq \{1, 2, \ldots, n\}$ is the set of indices of all terminal points adjacent to $\boldsymbol{y}^i$, and $N_2(i) \subseteq \{1, 2, \ldots, m_1\}$ is the set of indices of all non-terminal points adjacent to $\boldsymbol{y}^i$. It has been proved under a mild condition that, by using Weiszfeld's method, the total edge length of the tree decreases monotonically and all non-terminal points converges [28].

Algorithm 3 Selection of Non-Terminal Points

Input: $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$, $Y = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^m\} \subset \mathbb{R}^2$, $n_c, n_g, K \in \mathbb{Z}_{++}$, $p_w, p_s, p_c, p_m, p_f \in (0, 1)$

Output: $T_1 = (X \cup Y_1, E_1)$ where $Y_1 \subseteq Y$

1: Set $t \leftarrow 0$.
2: Generate n_c binary sequences $\boldsymbol{b}^1, \boldsymbol{b}^2, \ldots, \boldsymbol{b}^{n_c}$ of length m in such a way that b_i^k, the i-th bit of the k-th sequence, is set to 1 with probability p_w and 0 with probability $1 - p_w$, and set $P \leftarrow \{\boldsymbol{b}^1, \boldsymbol{b}^2, \ldots, \boldsymbol{b}^{n_c}\}$.
3: Evaluate sequences in P using Algorithm 2. Let $\boldsymbol{b}^{\text{best}} \in P$ be a sequence with the highest fitness value.
4: Set $t \leftarrow t + 1$ and Set $P' \leftarrow \{\boldsymbol{b}^{\text{best}}\}$.
5: Select K sequences randomly from P with equal probability, and then select one from the K sequences with the highest fitness value. Let $\boldsymbol{b}^{k_1}$ be the sequence.
6: Select one from the three genetic operations: *selection*, *crossover* and *mutation*, randomly with probability p_s, p_c and p_m, respectively. If *selection* is selected, go to Step 7. If *crossover* is selected, go to Step 8. If *mutation* is selected, go to Step 9.
7: Add $\boldsymbol{b}^{k_1}$ to P', and go to Step 10.
8: Select K sequences randomly from P with equal probability, and then select one from the K sequences with the highest fitness value. Let $\boldsymbol{b}^{k_2}$ be the sequence. Apply the single-point crossover operation to $\boldsymbol{b}^{k_1}$ and $\boldsymbol{b}^{k_2}$, add the generated two sequences to P', and go to Step 10.
9: For $i = 1, 2, \ldots, m$, set $b_i^{k_1} \leftarrow 1 - b_i^{k_1}$ with probability p_f, and then add $\boldsymbol{b}^{k_1}$ to P'.
10: If $|P'| < n_c$ then go to Step 5.
11: If $t < n_g$ then set $P \leftarrow P'$ and go to Step 3.
12: Evaluate sequences in P' using Algorithm 2. Let $\boldsymbol{b}^{\text{best}} \in P'$ be a sequence with the highest fitness value, and $T_1 = (X \cup Y_1, E_1)$ be the tree corresponding to $\boldsymbol{b}^{\text{best}}$.
13: Return T_1 and stop.

Note that it sometimes occurs that a non-terminal point approaches a terminal point or two non-terminal points approach each other. In these cases, the proposed method merges two points into one.

The above-mentioned procedure to optimize the locations of non-terminal points is formally stated in Algorithm 4. Note that the tree returned by Algorithm 4 does not contain any non-terminal points of degree one or two. This can be justified as follows. If $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$ in Step 7 are not adjacent then the degree of $\boldsymbol{y}^i$ after the merge is no less than the degrees of $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$ before the merge. If $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$ are adjacent then no vertex can be adjacent to both of them; otherwise, $T_1 = (X \cup Y_1, E_1)$ would not be a tree. In this case, the degree of $\boldsymbol{y}^i$ after the merge is no less than the degrees of $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$ before the merge.

Figure 2(c) shows the result of the optimization of the locations of the non-terminal points in Fig. 2(b). The number of non-terminal points in Fig. 2(c) is equal to that in Fig. 2(b) in this case. However, their locations are optimized, and thus

Algorithm 4 Optimization of Locations of Non-Terminal Points

Input: $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n\}$, $Y_1 = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^{m_1}\} \subset \mathbb{R}^2$, $T_1 = (X \cup Y_1, E_1)$, $\epsilon \in \mathbb{R}_{++}$

Output: Updated T_1

1: Let L be the length of the diagonal line of the smallest axis-aligned rectangle containing all points in $X \cup Y_1$.

2: Set $U \leftarrow \{(i, j) \mid \boldsymbol{x}^i \in X, \boldsymbol{y}^j \in Y_1, \|\boldsymbol{x}^i - \boldsymbol{y}^j\| < \epsilon L\}$.

3: If $U = \emptyset$ then go to Step 5.

4: Select a pair $(i, j) \in U$. Replace $\boldsymbol{y}^j$ with $\boldsymbol{x}^i$ in all edges $\{\boldsymbol{y}^j, \cdot\} \in E_1$, remove duplicate edges and the self-loop $\{\boldsymbol{x}^i, \boldsymbol{x}^i\}$ from E_1, remove all pairs $(\cdot, j)$ from U, and remove $\boldsymbol{y}^j$ from Y_1. Go to Step 3.

5: Set $V \leftarrow \{(i, j) \mid i < j, \boldsymbol{y}^i \in Y_1, \boldsymbol{y}^j \in Y_1, \|\boldsymbol{y}^i - \boldsymbol{y}^j\| < \epsilon L\}$.

6: If $V = \emptyset$ then go to Step 8.

7: Select a pair $(i, j) \in V$. Replace $\boldsymbol{y}^j$ with $\boldsymbol{y}^i$ in all edges $\{\boldsymbol{y}^j, \cdot\} \in E_1$, remove duplicate edges and the self-loop $\{\boldsymbol{y}^i, \boldsymbol{y}^i\}$ from E_1, remove all pairs $(\cdot, j)$ and $(j, \cdot)$ from V, and remove $\boldsymbol{y}^j$ from Y_1. Go to Step 6.

8: If $Y_1 = \emptyset$ then return $T_1 = (X, E_1)$ and stop. Otherwise update the values of all members of Y_1 sequentially using Weiszfeld's method (6).

9: If $\|\boldsymbol{y}^i - \boldsymbol{y}^{i,\mathrm{old}}\| \geq \epsilon L$ for some $\boldsymbol{y}^i \in Y_1$, where $\boldsymbol{y}^{i,\mathrm{old}}$ is the value of $\boldsymbol{y}^i$ before update, then go to Step 2. Otherwise return $T_1 = (X \cup Y_1, E_1)$ and stop.

the angles between edges incident to the same non-terminal point are approximately equal to $2\pi/3$ radians.

E. REFINEMENT OF TREE

Trees after the optimization of the locations of non-terminal points tend to have fewer non-terminal points than the corresponding ESTs, and hence their total edge lengths are not sufficiently close to the optimal. This is a weakness of the authors' previous method [13].

To overcome this weakness, the proposed method adds new non-terminal points to the tree and optimize their locations in the fourth stage. To be more precise, the proposed method first searches for a pair (e_1, e_2) of adjacent edges that share a terminal point and for which the triangle formed by the three endpoints of e_1 and e_2 has no internal angle greater than or equal to $2\pi/3$ radians. It next generates a new non-terminal point $\boldsymbol{s}$ using Algorithm 1 for the three endpoints. It then adds new edges connecting $\boldsymbol{s}$ with the three endpoints and remove e_1 and e_2. This process is repeated until there is no such pair of adjacent edges.

The proposed method finally runs Algorithm 4 again because the locations of non-terminal points are not optimal in general.

The above-mentioned procedure to refine the tree is formally stated in Algorithm 5. Note that the tree returned by Algorithm 5 does not contain any non-terminal points of degree one or two. This can be justified as follows. The new non-terminal point $\boldsymbol{s}$ introduced in Step 2 has degree three, and the degrees of all other non-terminal points remain unchanged after the insertion of $\boldsymbol{s}$. Moreover, the tree returned by Algorithm 4 also contains no non-terminal points of degree one or two.

Algorithm 5 Refinement of Tree

Input: $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n\}$, $Y_2 = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^{m_2}\} \subset \mathbb{R}^2$, $T_2 = (X \cup Y_2, E_2)$, $\epsilon \in \mathbb{R}_{++}$

Output: Updated T_2

1: Search for a pair (e_1, e_2) of distinct edges in E_2 such that they share one common vertex in X and, for the three vertices incident to e_1 or/and e_2, Algorithm 1 returns a point $\boldsymbol{s} \in \mathbb{R}^2$. If such a pair does not exist, go to Step 3.

2: Add $\boldsymbol{s}$ to Y_2, add three edges connecting $\boldsymbol{s}$ to the three vertices incident to e_1 or/and e_2 to E_2, and remove e_1 and e_2 from E_2. Go to Step 1.

3: Optimize the locations of non-terminal points in the tree $T_2 = (X \cup Y_2, E_2)$ using Algorithm 4, and return the resulting T_2.

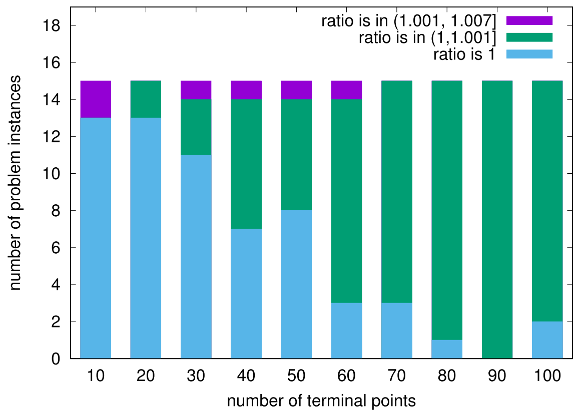


FIGURE 3. The number of problem instances for which the proposed method achieved a ratio equal to 1, in the interval (1, 1.001] and in the interval (1.001, 1.007].

Figure 2(d) shows the result of the refinement of the tree in Fig. 2(c). It is easily seen that Fig. 2(d) contains three more non-terminal points than Fig. 2(c).

IV. EXPERIMENTS

A. SETUP

In order to evaluate the performance of the proposed method, we implemented it in Python language version 3.11 and C++ language,[1] and conducted experiments using 195 problem instances from the OR-Library [25], which are labeled from 10-1 to 1000-15. The number before the hyphen indicates the number of terminal points, while the number after the hyphen represents the index of the instance. The parameters in the proposed method were fixed as shown in Table 1. Those values were determined through preliminary

[1] Only Prim's algorithm for constructing the MST is written in C++ language and called from Python using ctypes library.

TABLE 1. The values of the parameters used in the experiments.

Symbol	Value	Description
C	0.3	positive constant to determine the radius r^i in (5)
q	7	number of non-terminal points generated around each terminal point
n_c	200	number of chromosomes in each generation of the genetic algorithm
n_g	100	number of generations in the genetic algorithm
p_w	$1/7$	probability that each gene is set to 1 in the initial population of the genetic algorithm
K	5	number of randomly selected chromosomes in the tournament selection rule
p_s	0.2	probability that selection is selected from three genetic operations
p_c	0.6	probability that crossover is selected from three genetic operations
p_m	0.2	probability that mutation is selected from three genetic operations
p_f	0.1	probability that the value of each gene is flipped in mutation operation
ϵ	10^{-9}	positive constant used in Algorithm 4

TABLE 2. Total edge lengths and the numbers of non-terminal points of ESTs and trees obtained by the proposed method (Part 1).

Instance	Total Edge Length				No. of Non-Terminal Points			
	EST	Proposed Method			EST	Proposed Method		
		Min.	Avg.	Max.		Min.	Avg.	Max.
10-1	2.020673795	**2.020673795** (1.00000)	**2.020673795** (1.00000)	**2.020673795** (1.00000)	4	4	4.0	4
10-2	1.606868229	**1.606868229** (1.00000)	**1.606868229** (1.00000)	**1.606868229** (1.00000)	2	2	2.0	2
10-3	2.228074322	**2.228074322** (1.00000)	**2.228074322** (1.00000)	**2.228074322** (1.00000)	2	2	2.0	2
10-4	1.798596253	**1.798596253** (1.00000)	**1.798596253** (1.00000)	**1.798596253** (1.00000)	2	2	2.0	2
10-5	1.694433309	**1.694433309** (1.00000)	**1.694433309** (1.00000)	**1.694433309** (1.00000)	3	3	3.0	3
10-6	2.309602568	**2.309602568** (1.00000)	**2.309602568** (1.00000)	**2.309602568** (1.00000)	4	4	4.0	4
10-7	2.233858599	**2.233858599** (1.00000)	**2.233858599** (1.00000)	**2.233858599** (1.00000)	4	4	4.0	4
10-8	2.177682904	**2.177682904** (1.00000)	**2.177682904** (1.00000)	**2.177682904** (1.00000)	3	3	3.0	3
10-9	1.968478249	1.980716968 (1.00622)	1.980716968 (1.00622)	1.980716968 (1.00622)	5	4	4.0	4
10-10	2.059331690	**2.059331690** (1.00000)	**2.059331690** (1.00000)	**2.059331690** (1.00000)	4	4	4.0	4
10-11	1.947322109	**1.947322109** (1.00000)	**1.947322109** (1.00000)	**1.947322109** (1.00000)	4	4	4.0	4
10-12	1.753123664	**1.753123664** (1.00000)	**1.753123664** (1.00000)	**1.753123664** (1.00000)	2	2	2.0	2
10-13	1.713886731	**1.713886731** (1.00000)	**1.713886731** (1.00000)	**1.713886731** (1.00000)	3	3	3.0	3
10-14	1.949652208	**1.949652208** (1.00000)	**1.949652208** (1.00000)	**1.949652208** (1.00000)	6	6	6.0	6
10-15	1.671645607	1.676122287 (1.00268)	1.676122287 (1.00268)	1.676122287 (1.00268)	4	4	4.0	4
20-1	3.071642745	3.072208195 (1.00018)	3.072208195 (1.00018)	3.072208195 (1.00018)	8	8	8.0	8
20-2	2.854631374	**2.854631374** (1.00000)	2.854698307 (1.00002)	2.855300706 (1.00023)	8	8	8.0	8
20-3	2.453091823	**2.453091823** (1.00000)	2.454843042 (1.00071)	2.455280847 (1.00089)	6	6	6.8	7
20-4	2.466116544	**2.466116544** (1.00000)	**2.466116544** (1.00000)	**2.466116544** (1.00000)	9	9	9.0	9
20-5	2.953547045	**2.953547045** (1.00000)	**2.953547045** (1.00000)	**2.953547045** (1.00000)	9	10	10.0	10
20-6	3.131569521	**3.131569521** (1.00000)	**3.131569521** (1.00000)	**3.131569521** (1.00000)	6	6	6.0	6
20-7	3.059300170	**3.059300170** (1.00000)	**3.059300170** (1.00000)	**3.059300170** (1.00000)	9	9	9.0	9
20-8	3.316986082	**3.316986082** (1.00000)	**3.316986082** (1.00000)	**3.316986082** (1.00000)	7	7	7.0	7
20-9	3.133634165	**3.133634165** (1.00000)	**3.133634165** (1.00000)	**3.133634165** (1.00000)	7	7	7.0	7
20-10	3.011872639	**3.011872639** (1.00000)	3.013441553 (1.00052)	3.014487496 (1.00087)	8	8	8.0	8
20-11	2.318052600	**2.318052600** (1.00000)	**2.318052600** (1.00000)	**2.318052600** (1.00000)	8	8	8.0	8
20-12	2.653745316	**2.653745316** (1.00000)	**2.653745316** (1.00000)	**2.653745316** (1.00000)	7	7	7.0	7
20-13	3.022848161	3.022882674 (1.00001)	3.022882674 (1.00001)	3.022882674 (1.00001)	10	9	9.0	9
20-14	2.933008589	**2.933008589** (1.00000)	**2.933008589** (1.00000)	**2.933008589** (1.00000)	6	6	6.0	6
20-15	2.791479480	**2.791479480** (1.00000)	**2.791479480** (1.00000)	**2.791479480** (1.00000)	6	6	6.0	6

experiments. For each instance, the Python program was run 10 times, and the minimum, average and maximum of the total edge lengths, the numbers of non-terminal points in the obtained trees, and the execution time were recorded. All experiments were performed on a computer with an Intel® Core™ i9-13900KF and 32GB RAM running Windows 11.

B. RESULTS

Tables 2–6 show the minimum, average and maximum of the total edge lengths and the numbers of non-terminal points in the obtained trees, as well as the total edge length and the number of Steiner points of the EST provided in the OR-Library, for each of the 150 instances with 10 to 100 terminal points. The numbers in parentheses are ratios to the total edge lengths of the ESTs. The bold numbers indicate that the total edge length of the tree constructed using the proposed method is equal to that of the EST.

We see from Tables 2–6 that the proposed method found an EST in 61 out of the 150 instances. The proposed method also achieved a ratio within the interval (1, 1.001] for 83 instances, and within the interval (1.001, 1.007] for six instances. The distributions of these three types of instances for 10 groups categorized by the number of terminal points are shown in Fig. 3. It can be observed from the figure that the number of instances for which the proposed method constructed an EST decreases as the number of terminal points increases. It can also be observed from the figure that the proposed method sometimes failed to achieve a ratio less than or equal to 1.001 in instances with 60 or fewer terminal points. In contrast, for instances with 70 or more terminal points,

TABLE 3. Total edge lengths and the numbers of non-terminal points of ESTs and trees obtained by the proposed method (Part 2).

Instance	Total Edge Length				No. of Non-Terminal Points			
	EST	Proposed Method			EST	Proposed Method		
		Min.	Avg.	Max.		Min.	Avg.	Max.
30-1	3.578760143	**3.578760143** (1.00000)	**3.578760143** (1.00000)	**3.578760143** (1.00000)	11	11	11.0	11
30-2	3.576654401	**3.576654401** (1.00000)	**3.576654401** (1.00000)	**3.576654401** (1.00000)	11	11	11.0	11
30-3	3.656897228	**3.656897228** (1.00000)	**3.656897228** (1.00000)	**3.656897228** (1.00000)	11	11	11.0	11
30-4	3.711412942	**3.711412942** (1.00000)	**3.711412942** (1.00000)	**3.711412942** (1.00000)	15	15	15.0	15
30-5	3.613844843	**3.613844843** (1.00000)	**3.613844843** (1.00000)	**3.613844843** (1.00000)	10	10	10.0	10
30-6	3.497442722	**3.497442722** (1.00000)	3.500002973 (1.00073)	3.502181867 (1.00136)	12	10	10.9	12
30-7	3.813681007	**3.813681007** (1.00000)	3.815479065 (1.00047)	3.815928579 (1.00059)	12	11	11.2	12
30-8	3.685799965	**3.685799965** (1.00000)	3.686281645 (1.00013)	3.689533205 (1.00101)	13	13	13.1	14
30-9	3.180977239	**3.180977239** (1.00000)	**3.180977239** (1.00000)	**3.180977239** (1.00000)	11	11	11.0	11
30-10	3.718992382	3.718992383 (1.00000)	3.718992383 (1.00000)	3.718992383 (1.00000)	11	12	12.0	12
30-11	3.590187768	**3.590187768** (1.00000)	**3.590187768** (1.00000)	**3.590187768** (1.00000)	9	9	9.0	9
30-12	3.423946991	3.427406461 (1.00101)	3.427413478 (1.00101)	3.427476626 (1.00103)	13	12	12.1	13
30-13	3.222445213	**3.222445213** (1.00000)	**3.222445213** (1.00000)	**3.222445213** (1.00000)	9	9	9.0	9
30-14	3.853249608	3.855672814 (1.00063)	3.857637537 (1.00114)	3.866729049 (1.00350)	15	15	15.2	16
30-15	3.771808282	3.772952792 (1.00030)	3.773208909 (1.00037)	3.773388195 (1.00042)	12	12	13.0	14
40-1	3.928354415	3.931451834 (1.00079)	3.931971269 (1.00092)	3.935466703 (1.00181)	19	18	18.0	18
40-2	4.066874017	**4.066874017** (1.00000)	4.067121256 (1.00006)	4.069048100 (1.00053)	18	17	18.8	20
40-3	4.384545715	**4.384545715** (1.00000)	4.387263565 (1.00062)	4.390060638 (1.00126)	15	12	13.3	15
40-4	3.853166642	**3.853166642** (1.00000)	**3.853166642** (1.00000)	**3.853166642** (1.00000)	16	16	16.0	16
40-5	4.543251968	**4.543251968** (1.00000)	4.545085653 (1.00040)	4.547836181 (1.00101)	13	13	13.0	13
40-6	4.415198331	4.420373938 (1.00117)	4.422122471 (1.00157)	4.423337337 (1.00184)	15	15	15.0	15
40-7	4.031922786	4.031922787 (1.00000)	4.037533609 (1.00139)	4.038157034 (1.00155)	16	15	15.2	17
40-8	4.273487039	**4.273487039** (1.00000)	4.273543793 (1.00001)	4.274054575 (1.00013)	13	13	13.0	13
40-9	4.622412940	4.622452980 (1.00001)	4.624809718 (1.00052)	4.625894785 (1.00075)	18	17	17.1	18
40-10	5.083206018	5.083206019 (1.00000)	5.086706153 (1.00069)	5.093752172 (1.00207)	19	19	19.1	20
40-11	4.139926929	**4.139926929** (1.00000)	4.140178921 (1.00006)	4.140376672 (1.00011)	14	14	14.0	14
40-12	3.907862365	3.909153257 (1.00033)	3.910750246 (1.00074)	3.911149493 (1.00084)	16	17	17.0	17
40-13	4.560496435	**4.560496435** (1.00000)	**4.560496435** (1.00000)	**4.560496435** (1.00000)	14	14	14.0	14
40-14	4.357807960	4.359022419 (1.00028)	4.359152195 (1.00031)	4.359517404 (1.00039)	19	17	17.2	19
40-15	4.507584671	4.507908941 (1.00007)	4.507908941 (1.00007)	4.507908941 (1.00007)	16	16	16.0	16

TABLE 4. Total edge lengths and the numbers of non-terminal points of ESTs and trees obtained by the proposed method (Part 3).

Instance	Total Edge Length				No. of Non-Terminal Points			
	EST	Proposed Method			EST	Proposed Method		
		Min.	Avg.	Max.		Min.	Avg.	Max.
50-1	4.836601423	**4.836601423** (1.00000)	4.836847670 (1.00005)	4.837422247 (1.00017)	20	20	20.0	20
50-2	4.948404618	4.948634893 (1.00005)	4.952211855 (1.00077)	4.964110748 (1.00317)	23	22	22.3	23
50-3	4.747169968	**4.747169968** (1.00000)	**4.747169968** (1.00000)	**4.747169968** (1.00000)	20	20	20.0	20
50-4	4.469074736	**4.469074736** (1.00000)	4.469074737 (1.00000)	4.469074737 (1.00000)	22	22	22.7	23
50-5	4.864825693	**4.864825693** (1.00000)	4.867175475 (1.00048)	4.871440250 (1.00136)	22	21	21.9	22
50-6	4.923458590	4.923504042 (1.00001)	4.923555123 (1.00002)	4.923936519 (1.00010)	17	17	17.1	18
50-7	4.361318655	**4.361318655** (1.00000)	4.363285614 (1.00045)	4.366227650 (1.00113)	21	21	21.0	21
50-8	4.702746993	4.702943680 (1.00004)	4.704034790 (1.00027)	4.705539913 (1.00059)	19	19	19.2	20
50-9	4.676073800	4.676921160 (1.00018)	4.680608917 (1.00097)	4.683813527 (1.00166)	18	18	18.0	18
50-10	4.627790992	**4.627790992** (1.00000)	4.627955988 (1.00004)	4.628673878 (1.00019)	16	16	16.3	17
50-11	4.669385652	4.669612048 (1.00005)	4.670480403 (1.00023)	4.672395653 (1.00064)	20	20	20.2	22
50-12	4.673221536	4.673256048 (1.00001)	4.674511005 (1.00028)	4.675185863 (1.00042)	20	19	19.8	20
50-13	4.656471039	**4.656471039** (1.00000)	4.659529207 (1.00066)	4.668295006 (1.00254)	20	20	20.0	20
50-14	4.709868528	**4.709868528** (1.00000)	4.710301439 (1.00009)	4.713237302 (1.00072)	16	16	16.0	16
50-15	4.607990921	4.612944594 (1.00108)	4.613117259 (1.00111)	4.614671247 (1.00145)	19	19	19.0	19
60-1	4.774045341	4.774357870 (1.00007)	4.778847038 (1.00101)	4.785042894 (1.00230)	27	25	25.6	26
60-2	4.812986967	4.813466301 (1.00010)	4.813482878 (1.00010)	4.813487023 (1.00010)	24	23	23.5	24
60-3	4.945878320	4.946485813 (1.00012)	4.948302083 (1.00049)	4.949340340 (1.00070)	24	24	24.0	24
60-4	4.846180502	4.846603670 (1.00009)	4.847874524 (1.00035)	4.848706585 (1.00052)	24	22	22.8	24
60-5	4.835551265	4.835989025 (1.00009)	4.836130742 (1.00012)	4.836916491 (1.00028)	24	24	24.0	24
60-6	5.250457500	**5.250457500** (1.00000)	5.252288669 (1.00035)	5.256031922 (1.00106)	21	21	21.2	22
60-7	5.214252011	5.214649496 (1.00008)	5.217881417 (1.00070)	5.227678614 (1.00257)	28	28	29.1	30
60-8	5.117320734	**5.117320734** (1.00000)	5.118102171 (1.00015)	5.120171240 (1.00056)	19	19	19.4	20
60-9	4.908680813	4.909011680 (1.00007)	4.913519137 (1.00099)	4.915773051 (1.00144)	24	23	23.9	24
60-10	5.058701860	5.060086377 (1.00027)	5.061979790 (1.00065)	5.066859256 (1.00161)	23	23	24.6	25
60-11	4.932758774	4.932758775 (1.00000)	4.933579314 (1.00017)	4.934786958 (1.00041)	26	27	27.4	28
60-12	5.292435162	5.293873455 (1.00027)	5.298334988 (1.00111)	5.302513874 (1.00190)	24	23	23.9	24
60-13	5.266382273	**5.266382273** (1.00000)	5.271029919 (1.00088)	5.277702035 (1.00215)	20	19	19.3	20
60-14	5.023550188	5.028855043 (1.00106)	5.035540766 (1.00239)	5.038993927 (1.00307)	28	25	27.3	29
60-15	4.967095781	4.967979951 (1.00018)	4.969195155 (1.00042)	4.971454626 (1.00088)	26	26	26.0	26

the proposed method consistently achieved a ratio less than or equal to 1.001. This demonstrates the capability of the proposed method to construct high-accuracy approximate solutions in a stable manner.

TABLE 5. Total edge lengths and the numbers of non-terminal points of ESTs and trees obtained by the proposed method (Part 4).

Instance	Total Edge Length				No. of Non-Terminal Points			
	EST	Proposed Method			EST	Proposed Method		
		Min.	Avg.	Max.		Min.	Avg.	Max.
70-1	5.430374506	**5.430374506** (1.00000)	5.432610545 (1.00041)	5.435600477 (1.00096)	30	29	29.3	30
70-2	5.327590160	5.332189611 (1.00086)	5.334366627 (1.00127)	5.339082790 (1.00216)	30	29	30.4	31
70-3	5.391160605	5.392670837 (1.00028)	5.395234792 (1.00076)	5.398766767 (1.00141)	29	26	26.6	28
70-4	5.498932382	5.499192660 (1.00005)	5.500650424 (1.00031)	5.506092998 (1.00130)	25	23	23.6	24
70-5	5.476696679	5.476707198 (1.00000)	5.478354185 (1.00030)	5.482774572 (1.00111)	26	26	26.0	26
70-6	5.533596306	5.534843219 (1.00023)	5.540738669 (1.00129)	5.547320331 (1.00248)	30	31	31.2	32
70-7	5.502831476	5.505586852 (1.00050)	5.509759178 (1.00126)	5.513566078 (1.00195)	29	29	29.5	31
70-8	5.480649271	**5.480649271** (1.00000)	5.485615927 (1.00091)	5.492551459 (1.00217)	23	23	24.0	25
70-9	5.472164303	**5.472164303** (1.00000)	5.474834235 (1.00049)	5.480950682 (1.00161)	24	23	24.3	25
70-10	5.520368980	5.520601261 (1.00004)	5.521371663 (1.00018)	5.522599480 (1.00040)	26	25	26.1	27
70-11	5.717338928	5.718330511 (1.00017)	5.718398968 (1.00019)	5.718760972 (1.00025)	26	25	25.2	27
70-12	5.522830316	5.522866210 (1.00001)	5.527965239 (1.00093)	5.532341733 (1.00172)	28	28	28.0	28
70-13	5.444450389	5.445973816 (1.00028)	5.448957464 (1.00083)	5.452932625 (1.00156)	26	26	27.1	28
70-14	5.352111315	5.353536043 (1.00027)	5.356469709 (1.00081)	5.359406352 (1.00136)	27	27	27.5	28
70-15	5.519824091	5.520108635 (1.00005)	5.525878120 (1.00110)	5.531514287 (1.00212)	31	31	31.8	33
80-1	6.257418035	6.262033514 (1.00074)	6.265137281 (1.00123)	6.268163311 (1.00172)	35	33	34.9	36
80-2	5.695396267	5.698124185 (1.00048)	5.702677985 (1.00128)	5.706890069 (1.00202)	33	33	33.0	33
80-3	5.872480074	5.875400668 (1.00050)	5.878672891 (1.00105)	5.886048516 (1.00231)	36	34	34.2	35
80-4	5.624164105	5.626873390 (1.00048)	5.628869632 (1.00084)	5.632819416 (1.00154)	29	31	32.2	33
80-5	5.754511555	5.754800103 (1.00005)	5.756990593 (1.00043)	5.762288709 (1.00135)	32	31	32.0	33
80-6	6.163252761	6.163996089 (1.00012)	6.169031626 (1.00094)	6.186982471 (1.00385)	30	29	30.1	31
80-7	6.030849951	6.031318987 (1.00008)	6.034652955 (1.00063)	6.036321324 (1.00091)	35	33	33.8	36
80-8	5.952855469	5.957641017 (1.00080)	5.965850456 (1.00218)	5.971906488 (1.00320)	31	30	31.9	34
80-9	6.107672887	6.110499664 (1.00046)	6.113410104 (1.00094)	6.116375070 (1.00142)	35	34	34.7	37
80-10	5.714735044	5.714735045 (1.00000)	5.716701203 (1.00034)	5.720492507 (1.00101)	30	29	30.7	32
80-11	5.764836056	5.765687576 (1.00015)	5.768612320 (1.00066)	5.772900447 (1.00140)	32	33	33.1	34
80-12	5.673138760	5.673364276 (1.00004)	5.677079709 (1.00069)	5.680117947 (1.00123)	30	30	30.1	31
80-13	5.968368124	5.970011855 (1.00028)	5.973040916 (1.00078)	5.978221286 (1.00165)	30	29	30.2	31
80-14	6.117819783	**6.117819783** (1.00000)	6.123155343 (1.00087)	6.137112607 (1.00315)	32	32	32.8	34
80-15	6.143383707	6.143772087 (1.00006)	6.147213532 (1.00062)	6.150418170 (1.00115)	30	31	31.1	32

TABLE 6. Total edge lengths and the numbers of non-terminal points of ESTs and trees obtained by the proposed method (Part 5).

Instance	Total Edge Length				No. of Non-Terminal Points			
	EST	Proposed Method			EST	Proposed Method		
		Min.	Avg.	Max.		Min.	Avg.	Max.
90-1	6.056187009	6.058655956 (1.00041)	6.061875664 (1.00094)	6.068712029 (1.00207)	38	37	37.9	39
90-2	6.221350869	6.225003510 (1.00059)	6.227634399 (1.00101)	6.231118460 (1.00157)	39	38	39.8	41
90-3	6.460569259	6.464062846 (1.00054)	6.465903837 (1.00083)	6.467805329 (1.00112)	36	35	35.5	37
90-4	6.257681402	6.261895412 (1.00067)	6.263803109 (1.00098)	6.266960769 (1.00148)	36	37	38.1	39
90-5	6.389159126	6.389351003 (1.00003)	6.399997419 (1.00170)	6.408516376 (1.00303)	39	36	37.8	39
90-6	6.046532095	6.050677082 (1.00069)	6.054233868 (1.00127)	6.057991463 (1.00190)	38	35	36.1	38
90-7	6.249417094	6.249586637 (1.00003)	6.251148028 (1.00028)	6.257384524 (1.00127)	40	39	39.4	40
90-8	6.306456458	6.308761022 (1.00037)	6.310717845 (1.00068)	6.314849856 (1.00133)	38	35	37.1	38
90-9	5.941531189	5.941939615 (1.00007)	5.946049178 (1.00076)	5.949751477 (1.00138)	38	36	37.1	39
90-10	6.320064030	6.320064032 (1.00000)	6.325991268 (1.00094)	6.342893877 (1.00361)	36	35	36.4	38
90-11	6.280806724	6.280806725 (1.00000)	6.290902140 (1.00161)	6.297528757 (1.00266)	34	34	34.9	37
90-12	6.082185432	6.082185433 (1.00000)	6.086982404 (1.00079)	6.096223617 (1.00231)	40	40	41.8	43
90-13	6.305672180	6.306698600 (1.00016)	6.308295090 (1.00042)	6.311185388 (1.00087)	38	37	37.5	39
90-14	6.094139846	6.095352441 (1.00020)	6.099592342 (1.00089)	6.109540527 (1.00253)	38	37	37.9	39
90-15	6.249653001	6.251551700 (1.00030)	6.254001013 (1.00070)	6.257700140 (1.00129)	32	30	30.7	32
100-1	6.394256000	6.399787929 (1.00087)	6.404918880 (1.00167)	6.410283409 (1.00251)	44	41	42.6	44
100-2	6.594812128	**6.594812128** (1.00000)	6.604294256 (1.00144)	6.613350210 (1.00281)	43	41	41.9	43
100-3	6.531347066	6.533952161 (1.00040)	6.537623106 (1.00096)	6.545000945 (1.00209)	39	39	40.0	42
100-4	6.576977411	6.580053469 (1.00047)	6.586001360 (1.00137)	6.598656287 (1.00330)	44	42	42.9	44
100-5	6.674687782	6.676334051 (1.00025)	6.682592114 (1.00118)	6.692181066 (1.00262)	37	37	37.9	39
100-6	6.466368368	6.466765562 (1.00006)	6.471537960 (1.00080)	6.481458189 (1.00233)	39	38	39.2	41
100-7	6.987863494	6.991512264 (1.00052)	7.000391225 (1.00179)	7.018116792 (1.00433)	45	39	43.6	46
100-8	6.394971137	6.395667966 (1.00011)	6.397394457 (1.00038)	6.404588741 (1.00150)	38	39	39.5	40
100-9	6.914321113	6.918436042 (1.00060)	6.926653811 (1.00178)	6.933837424 (1.00282)	48	44	46.2	48
100-10	6.719510819	**6.719510819** (1.00000)	6.723663727 (1.00062)	6.728095771 (1.00128)	38	37	38.2	41
100-11	6.832950895	6.834669474 (1.00025)	6.838999941 (1.00089)	6.847837416 (1.00218)	40	40	41.6	43
100-12	6.670622612	6.671513927 (1.00013)	6.673986235 (1.00050)	6.680232486 (1.00144)	41	40	40.8	42
100-13	6.505252521	6.507172448 (1.00030)	6.513149492 (1.00121)	6.518100651 (1.00198)	37	38	38.7	41
100-14	6.882598532	6.887367090 (1.00069)	6.890544645 (1.00115)	6.894562667 (1.00174)	44	43	44.0	46
100-15	6.205148881	6.207522820 (1.00038)	6.218386613 (1.00213)	6.231280313 (1.00421)	34	34	35.4	37

Figure 4 illustrates the distribution of 150 ratios for each problem size. Across the range of problem sizes, the distribution of the ratios remains tightly concentrated around 1.0, with only slight upward deviations. The narrow

TABLE 7. Comparison of the best trees obtained by the proposed and previous methods for the six instances considered by Zhang et al. [13].

Instance	Total Edge Length			No. of Non-Terminal Points		
	EST	Proposed	Previous [13]	EST	Proposed	Previous [13]
10-1	2.020673795	**2.020673795** (1.00000)	**2.020673795** (1.00000)	4	4	4
20-2	2.854631374	**2.854631374** (1.00000)	**2.854631374** (1.00000)	8	8	8
30-1	3.578760143	**3.578760143** (1.00000)	3.579130322 (1.00010)	11	11	10
40-2	4.066874017	**4.066874017** (1.00000)	4.070599316 (1.00092)	18	18	13
50-2	4.948404618	4.948634893 (1.00005)	4.955753420 (1.00149)	23	23	16
100-2	6.594812128	**6.594812128** (1.00000)	6.620518385 (1.00390)	43	43	29

TABLE 8. Total edge lengths and the numbers of non-terminal points of trees obtained by the proposed method for 45 problem instances with 250, 500 and 1,000 terminal points.

Instance	Total Edge Length			No. of Non-Terminal Points		
	Proposed Method			Proposed Method		
	Min.	Avg.	Max.	Min.	Avg.	Max.
250-1	10.288418900	10.299510631	10.309213780	101	102.8	105
250-2	10.128332540	10.136147181	10.145236160	95	101.3	104
250-3	10.064799630	10.083314736	10.100964820	96	100.0	105
250-4	10.408614450	10.423517965	10.435723690	101	104.0	109
250-5	10.262396940	10.270141728	10.275168960	101	103.0	104
250-6	10.232150210	10.243945546	10.254156270	95	97.3	101
250-7	10.158168800	10.166415591	10.178159420	97	102.2	109
250-8	10.318193440	10.335429367	10.349132070	97	100.7	105
250-9	10.322272870	10.334523332	10.350497420	102	103.8	106
250-10	10.260885110	10.271598229	10.286946780	95	98.1	101
250-11	9.914059848	9.921682849	9.936570539	99	102.7	106
250-12	10.503926250	10.519120030	10.537809890	98	102.8	107
250-13	10.163733430	10.185418161	10.202119930	99	103.3	106
250-14	10.284218000	10.292899494	10.304236780	106	108.5	112
250-15	10.176258490	10.188717758	10.197336490	99	103.5	108
500-1	14.381593480	14.394538926	14.412784370	209	213.7	224
500-2	14.257607960	14.267747172	14.281985370	208	213.0	218
500-3	14.356809570	14.375667776	14.393122520	205	213.9	217
500-4	14.470996690	14.488584995	14.511249780	202	209.9	215
500-5	14.113535960	14.125606143	14.141623480	200	204.3	208
500-6	14.571264810	14.584637836	14.597613230	206	211.6	217
500-7	14.102737530	14.114990762	14.135856320	200	206.0	212
500-8	14.185616940	14.208688204	14.219518050	197	201.9	209
500-9	14.248837100	14.260134369	14.280587460	200	204.4	210
500-10	13.717971700	13.733722501	13.764719590	211	216.5	222
500-11	14.230566490	14.244915606	14.261522910	200	205.7	210
500-12	14.441895400	14.461779023	14.481063100	193	198.3	205
500-13	14.172924460	14.187766808	14.213338950	205	209.3	213
500-14	14.702894730	14.722181793	14.734105460	210	215.8	220
500-15	14.162407510	14.171429849	14.188727870	206	210.1	212
1000-1	20.327382000	20.340002036	20.356255310	413	418.4	426
1000-2	20.170384000	20.188228725	20.204977050	421	423.2	428
1000-3	20.027650080	20.059866574	20.089361260	388	397.6	410
1000-4	20.308249240	20.340698723	20.358157780	427	432.2	440
1000-5	20.152253950	20.169539888	20.178609890	389	400.4	409
1000-6	20.393349530	20.411117400	20.440364660	412	420.9	428
1000-7	20.364781960	20.385790767	20.406796220	408	415.3	425
1000-8	20.312047820	20.334670181	20.356712190	407	415.5	423
1000-9	20.186568560	20.215046656	20.263660420	419	429.2	438
1000-10	20.213213700	20.235882239	20.256815520	405	411.6	418
1000-11	20.407532820	20.430198869	20.444152380	418	423.8	433
1000-12	20.467451130	20.480335208	20.499110740	419	429.6	435
1000-13	20.077959360	20.093735836	20.109304570	408	414.8	420
1000-14	20.672876710	20.685710937	20.711322920	408	416.0	425
1000-15	20.268914750	20.285395354	20.302759070	413	420.1	434

interquartile ranges and short whiskers indicate low variability in the ratio, suggesting that the proposed method maintains stable performance even as the problem size increases.

Table 7 shows a comparison of the best trees obtained by the proposed and previous methods for the six instances considered in the international conference paper [13]. The bold numbers indicate that the total edge length of the tree is equal to that of the EST provided in the OR-Library. It is clearly seen from the table that the proposed method overcomes a key limitation of the previous method, which tends to produce a small number of non-terminal points, and yields better solutions for all of the six instances.

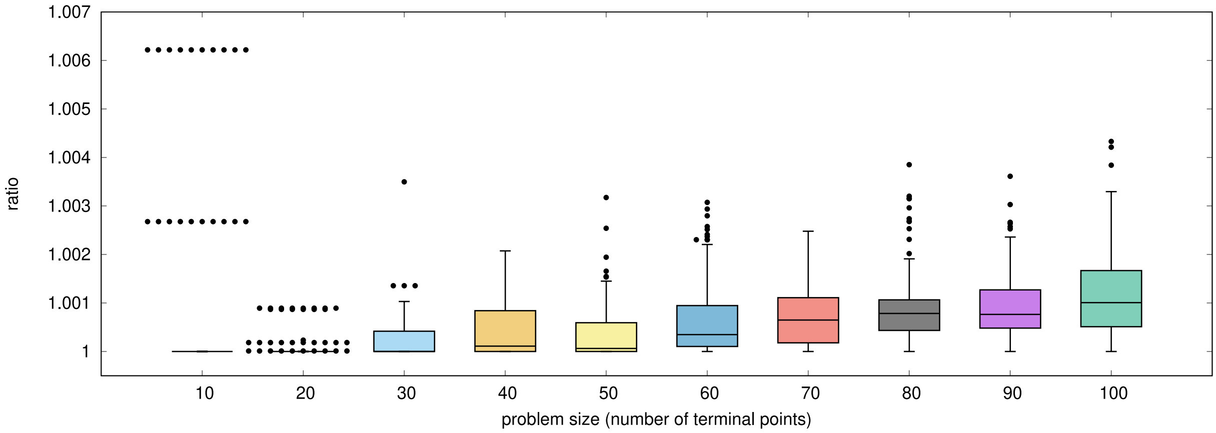


FIGURE 4. Boxplots of the ratios obtained by the proposed method for the 150 instances in the OR-Library with known optimal solutions.

TABLE 9. Execution time (in second) of the proposed method.

Instance	Minimum	Average	Maximum
10-1 to 10-15	0.29500	0.31140	0.40000
20-1 to 20-15	0.33900	0.35609	0.38600
30-1 to 30-15	0.39400	0.42370	0.64400
40-1 to 40-15	0.47300	0.51726	0.52300
50-1 to 50-15	0.54100	0.60045	0.74600
60-1 to 60-15	0.62400	0.71369	0.87600
70-1 to 70-15	0.73600	0.81714	0.98500
80-1 to 80-15	0.86500	0.94561	1.13900
90-1 to 90-15	0.92500	1.06140	1.28200
100-1 to 100-15	1.07300	1.18008	1.44700
250-1 to 250-15	3.99800	4.21385	4.55900
500-1 to 500-15	14.01400	15.27572	16.38600
1000-1 to 1000-15	53.94900	57.27826	60.94800

Table 8 shows the minimum, average and maximum of the total edge lengths and the numbers of non-terminal points in the trees obtained by the proposed method for each of the 45 instances with 250, 500 and 1,000 terminal points. We do not evaluate the quality of the trees using ratio because the optimal solutions are not available in the OR-Library. However, these results may serve as a benchmark for evaluating the performance of algorithms developed in the future.

Table 9 shows the execution times of the proposed method for the 195 problem instances. Each row of the table presents the minimum, average and maximum execution times obtained from 150 runs of the Python program, with 10 runs performed for each instance. It can be observed from the table that the proposed method is able to find an approximate solution in less than 1.5 seconds for the 15 instances with 100 terminal points. Even for the 15 instances with 1,000 terminal points, the proposed method is able to find an approximate solution in less than 61 seconds. Figure 5 illustrates the relationship between the problem size and the average execution time of the proposed method. The log-linear plot of the relationship produces a concave curve as shown in Fig. 5(a), while the log-log plot produces a convex curve as shown in Fig. 5(b). These observations indicate that the time complexity of the proposed method is sub-exponential. However, establishing this conclusion requires further theoretical analysis.

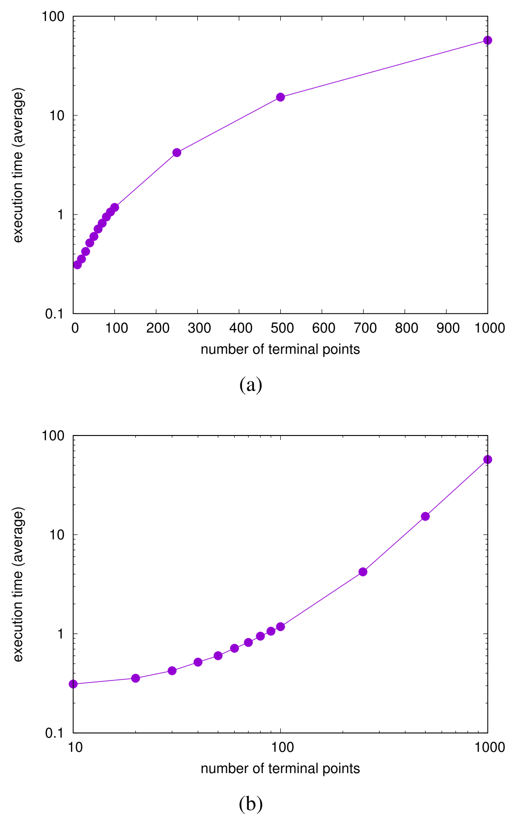
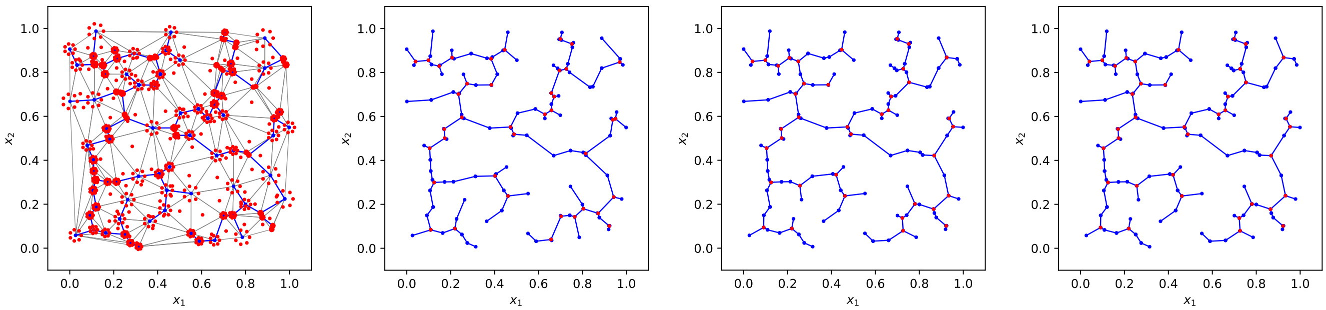


FIGURE 5. Relationship between the problem size and the average execution time of the proposed method, shown as (a) a log-linear plot and (b) a log-log plot.

C. DETAILED CASE ANALYSIS

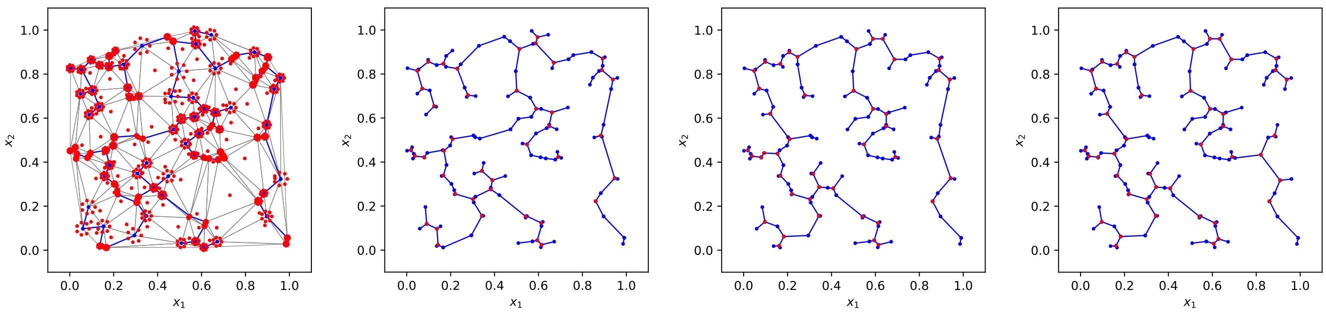
We select three problem instances from the OR-Library, and discuss the strengths and limitations of the proposed method.

The first instance is 100-10. Figure 6 shows (a) a Euclidean MST for the set of terminal points, (b) the tree obtained using



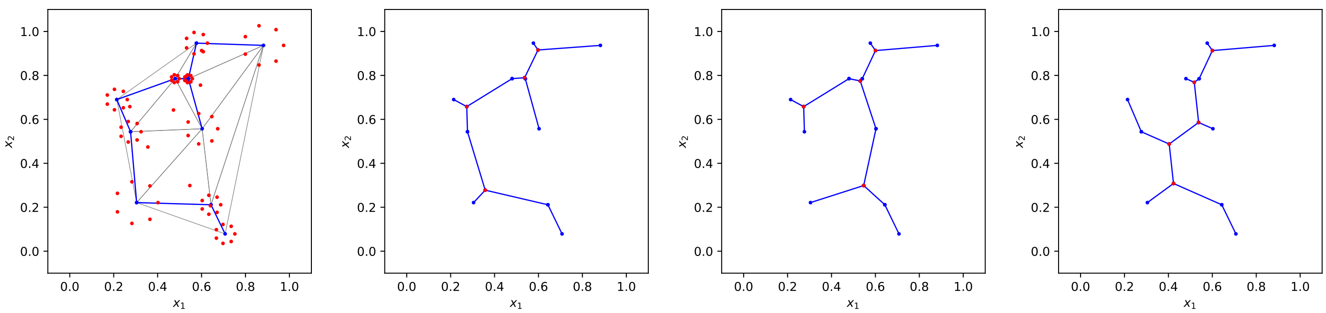
(a) (b) (c) (d)

FIGURE 6. Case analysis of Instance 100-10: (a) a Euclidean MST for the terminal points, (b) the tree obtained using the refinement of the Euclidean MST, (c) a tree obtained using the proposed method, and (d) the Euclidean Steiner tree provided in the OR-Library. The total edge lengths of the trees are (a) 6.953979314, (b) 6.816744444, (c) 6.719510819, and (d) 6.719510819. The number of non-terminal points are (a) 0, (b) 39, (c) 38, and (d) 38.



(a) (b) (c) (d)

FIGURE 7. Case analysis of Instance 100-1: (a) a Euclidean MST for the terminal points, (b) the tree obtained using the refinement of the Euclidean MST, (c) a tree obtained using the proposed method, and (d) the Euclidean Steiner tree provided in the OR-Library. The total edge lengths of the trees are (a) 6.608524624, (b) 6.437526688, (c) 6.399787929, and (d) 6.394256000. The number of non-terminal points are (a) 0, (b) 42, (c) 44, and (d) 44.



(a) (b) (c) (d)

FIGURE 8. Case analysis of Instance 10-9: (a) a Euclidean MST for the terminal points, (b) the tree obtained using the refinement of the Euclidean MST, (c) a tree obtained using the proposed method, and (d) the Euclidean Steiner tree provided in the OR-Library. The total edge lengths of the trees are (a) 2.018842093, (b) 1.981604778, (c) 1.980716968, and (d) 1.968478249. The number of non-terminal points are (a) 0, (b) 4, (c) 4, and (d) 5.

only the refinement of the Euclidean MST, (c) a tree obtained using the proposed method, and (d) the Steiner tree provided in the OR-Library. The total edge lengths of these four trees are (a) 6.953979314, (b) 6.816744444, (c) 6.719510819, and (d) 6.719510819. It can be observed that the trees in Figs. 6(b) and 6(c) satisfy the five conditions in Section II-A. However, the total edge lengths of these two trees are different. The proposed method is able to find the Steiner tree, while the refinement of the Euclidean MST alone is not sufficient to do it. This indicates that the first three stages of the proposed method, that is, generation of non-terminal points, selection of non-terminal points, and optimization of the locations of the selected non-terminal points, are certainly effective.

The second instance is 100-1. Figure 7 shows four trees as in Fig. 6. Their total edge lengths are (a) 6.608524624, (b) 6.437526688, (c) 6.399787929, and (d) 6.394256000. It can be observed that the trees in Figs. 7(b) and 7(c) satisfy the five conditions in Section II-A. The total edge length of the tree in Fig. 7(c) is shorter than that of the tree in Fig. 7(b), but longer than that of the Steiner tree

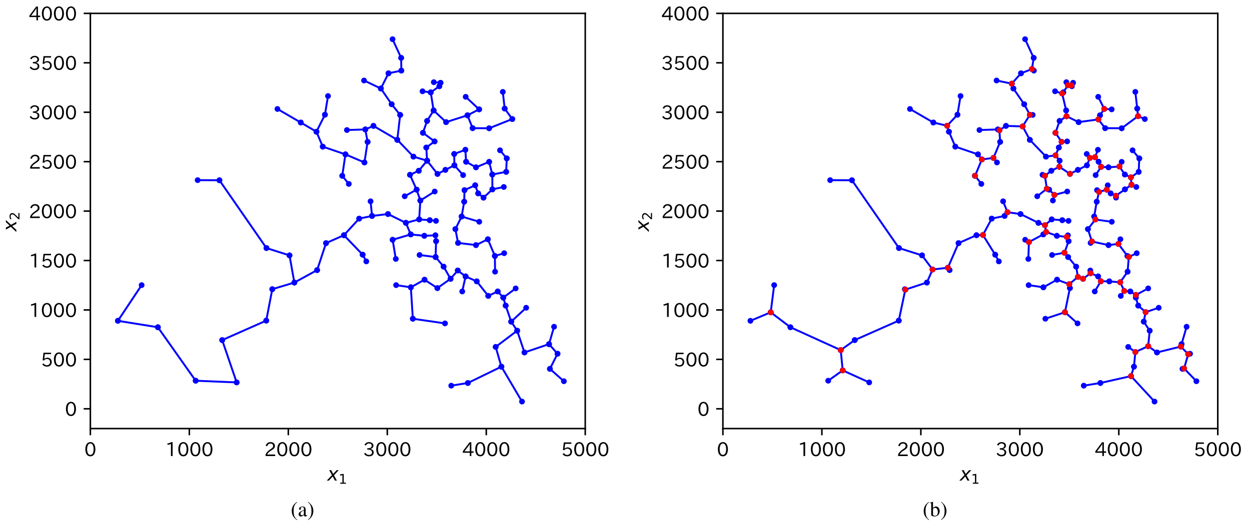


FIGURE 9. Results of the proposed method on the 144 terminal points considered by Cao et al. [14]: (a) a Euclidean MST for the terminal points and (b) a tree obtained by the proposed method. The number of non-terminal points in the latter is 66. The total edge lengths are (a) 26122.25135 and (b) 25108.80908.

in Fig. 7(d). This indicates that the first three stages of the proposed algorithm are effective, but the proposed method is not sufficient to find a Steiner tree.

The third instance is 10-9. Figure 8 shows four trees as in Figs. 6 and 7. Their total edge lengths are (a) 2.018842093, (b) 1.981604778, (c) 1.980716968, and (d) 1.968478249. The total edge length of the tree in Fig. 8(c) is shorter than that of the tree in Fig. 8(b), but longer than that of the Steiner tree in Fig. 8(d), as in the previous instance. This indicates that the first three stages of the proposed algorithm are effective, but not always sufficient to find a Steiner tree even in the case of 10 terminal points.

D. COMPARISON WITH OTHER METHODS

To further evaluate the performance of the proposed method, we compare it with the memetic algorithm developed by Bereta [11], the heuristics for the EST in higher dimensions proposed by Pinto and Maculan [12], and the Binary Particle Swarm Optimization (BPSO) algorithm developed by Cao et al. [14].

Bereta applied the memetic algorithm to the 150 instances from the OR-Library with known optimal solutions, and presented a graph illustrating the distributions of the minimum ratios for each problem size [11, Figure 9]. Comparing this graph with Fig. 4 and Tables 2–6, we can evaluate the performance of Bereta's algorithm and the proposed method. When considering the instances with 10 or 20 terminal points, Bereta's algorithm outperforms the proposed method because the former obtains optimal solutions for all of those instances, whereas the latter does not. For the instances with 30 or 40 terminal points, Bereta's algorithm and the proposed method exhibit comparable performance. For the instances with 50 to 100 terminal points, the proposed method clearly outperforms Bereta's algorithm. For example, in the instances with 80 terminal points, the minimum ratio obtained by the proposed method ranges from 1.00000 to 1.00080, whereas the minimum ratio obtained by Bereta's algorithm is greater than 1.0015 for all such instances.

Pinto and Maculan [12] proposed two heuristics which do not rely on the MST of the terminal points, and applied them to 75 instances from 10-1 to 50-15 in the OR-Library. Comparing the results [12, Table 9] with Tables 2–4, we can easily conclude that the proposed method significantly outperforms the heuristics in all aspects: the number of instances for which an optimal solution is obtained, the quality of the solution measured by the ratio, and the execution time. However, it should be noted that the heuristics were designed not for the ESTP in two dimensions but for higher dimensions.

Cao et al. [14] applied the BPSO algorithm to the problem of finding a communication network connecting 144 cities in China. The coordinates of the cities are given in their paper [14]. Although an approximate solution obtained by the BPSO algorithm is provided in [14], the total edge length of the solution is not shorter than a Euclidean MST for the set of terminal points, because non-terminal points are not used.[2] We thus compare an approximate solution obtained by the proposed method, which is shown in Fig. 9(b), with a Euclidean MST, which is shown in Fig. 9(a). The total edge length of the former is 25108.80908, while that of the latter is 26122.25135. The proposed method produced a tree whose total edge length is 96.12% of that of the Euclidean MST.

V. CONCLUSION

We presented a novel approximate solution method for the Euclidean Steiner tree problem, and demonstrated its effectiveness through experiments using 195 benchmark

[2] The BPSO algorithm may not have been properly used to solve the ESTP.

instances from the OR-Library. The proposed method is obtained by augmenting the authors' previous method [13], which utilizes a genetic algorithm and Weiszfeld's method, with initial non-terminal point generation based on the Delaunay triangulation, and tree refinement using a three-point algorithm.

Experimental results show that the proposed method outperforms the authors' previous method, the method using only the tree refinement, and existing evolutionary algorithms. To be more precise, among the 150 instances with known optimal solutions, the proposed method found the EST in 61 cases, a solution within 100.1% of the EST in 83 cases, and a solution within 100.7% of the EST in the remaining six cases. This indicates that the proposed method may be useful in many engineering problems.

However, the experimental results also show that the proposed method is not always sufficient to find the EST even for instances with a small number of terminal points. Improving the proposed method to more effectively solve the ESTP remains a significant challenge for future research. Furthermore, the proposed method is applicable only to two-dimensional ESTPs because it relies on the three-point algorithm. Extending the method to higher-dimensional ESTPs is another important avenue for future work.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

REFERENCES

[1] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," *SIAM J. Appl. Math.*, vol. 16, no. 1, pp. 1–29, Jan. 1968, doi: 10.1137/0116001.

[2] M. Brazil, R. L. Graham, D. A. Thomas, and M. Zachariasen, "On the history of the Euclidean Steiner tree problem," *Arch. for Hist. Exact Sci.*, vol. 68, no. 3, pp. 327–354, May 2014, doi: 10.1007/s00407-013-0127-z.

[3] M. R. Garey, R. L. Graham, and D. S. Johnson, "The complexity of computing Steiner minimal trees," *SIAM J. Appl. Math.*, vol. 32, no. 4, pp. 835–859, Jun. 1977, doi: 10.1137/0132072.

[4] J. M. Smith, D. T. Lee, and J. S. Liebman, "An O(n log n) heuristic for Steiner minimal tree problems on the Euclidean metric," *Networks*, vol. 11, no. 1, pp. 23–39, Mar. 1981, doi: 10.1002/net.3230110104.

[5] M. Fampa and K. M. Anstreicher, "An improved algorithm for computing Steiner minimal trees in Euclidean d-space," *Discrete Optim.*, vol. 5, no. 2, pp. 530–540, May 2008, doi: 10.1016/j.disopt.2007.08.006.

[6] J. W. Van Laarhoven and J. W. Ohlmann, "A randomized Delaunay triangulation heuristic for the Euclidean Steiner tree problem in $\mathbb{R}^d$," *J. Heuristics*, vol. 17, no. 4, pp. 353–372, Aug. 2011, doi: 10.1007/s10732-010-9137-z.

[7] Z.-X. Yang, X.-Y. Jia, J.-Y. Hao, and Y.-P. Gao, "Geometry-experiment algorithm for Steiner minimal tree problem," *J. Appl. Math.*, vol. 2013, pp. 1–10, Apr. 2013, doi: 10.1155/2013/367107.

[8] G.-Q. Dong, Z.-X. Yang, L. Song, K. Ye, and G.-S. Li, "The global shortest path visualization approach with obstructions," *J. Robot. Mechatronics*, vol. 27, no. 5, pp. 579–585, Oct. 2015, doi: 10.20965/jrm.2015.p0579.

[9] D. R. Dreyer and M. L. Overton, "Two heuristics for the Euclidean Steiner tree problem," *J. Global Optim.*, vol. 13, no. 1, pp. 95–106, Jan. 1998, doi: 10.1023/a:1008285504599.

[10] S. Hsu, F. I. S. Massolo, and L. P. Schaposnik, "A physarum-inspired approach to the Euclidean Steiner tree problem," *Sci. Rep.*, vol. 12, no. 1, pp. 14536:1–13, Aug. 2022, doi: 10.1038/s41598-022-18316-3.

[11] M. Bereta, "Baldwin effect and Lamarckian evolution in a memetic algorithm for Euclidean Steiner tree problem," *Memetic Comput.*, vol. 11, no. 1, pp. 35–52, Mar. 2019, doi: 10.1007/s12293-018-0256-7.

[12] R. V. Pinto and N. Maculan, "A new heuristic for the Euclidean Steiner tree problem in $\mathbb{R}^n$," *TOP*, vol. 31, no. 2, pp. 391–413, Jul. 2023, doi: 10.1007/s11750-022-00642-4.

[13] L. Zhang, T. Migita, and N. Takahashi, "A genetic algorithm approach to Euclidean Steiner tree problem," in *Proc. 7th Int. Conf. Robot Syst. Appl.*, Sep. 2024, pp. 71–75, doi: 10.1145/3702468.3702482.

[14] S. Cao, J. Chen, and J. Gou, "Binary particle swarm optimization algorithm for Euclidean Steiner tree," in *Proc. 4th Int. Conf. Comput. Sci. Blockchain*, 2024, pp. 417–422, doi: 10.1109/CCSB63463.2024.10735644.

[15] M. Bereta, "Evolutionary approach to the Euclidean Steiner tree problem in *n*-space," *Appl. Sci.*, vol. 15, no. 3, 2025, Art. no. 1413, doi: 10.3390/app15031413.

[16] B. Delaunay, "Sur la sphère vide. A la mémoire de Georges Voronoï," *Bulletin de l'Académie des Sciences de l'URSS*, no. 6, pp. 793–800, 1934.

[17] E. Weiszfeld, "Sur le point pour lequel la somme des distances de N points donnés est minimum," *Tohoku Math. J.*, vol. 43, pp. 355–386, 1937.

[18] E. Weiszfeld and F. Plastria, "On the point for which the sum of the distances to n given points is minimum," *Ann. Operations Res.*, vol. 167, no. 1, pp. 7–41, 2008, doi: 10.1007/s10479-008-0352-z.

[19] P. Winter, "An algorithm for the Steiner problem in the Euclidean plane," *Networks*, vol. 15, no. 3, pp. 323–345, Sep. 1985, doi: 10.1002/net.3230150305.

[20] P. Winter and M. Zachariasen, "Euclidean Steiner minimum trees: An improved exact algorithm," *Networks*, vol. 30, no. 3, pp. 149–166, Oct. 1997, doi: 10.1002/(sici)1097-0037(199710)30:3<149:aid-net1>3.0.co;2-l.

[21] J. W. Van Laarhoven and K. M. Anstreicher, "Geometric conditions for Euclidean Steiner trees in $\mathbb{R}^d$," *Comput. Geometry*, vol. 46, no. 5, pp. 520–531, Jul. 2013, doi: 10.1016/j.comgeo.2011.11.007.

[22] D. D. Juhl, D. M. Warme, P. Winter, and M. Zachariasen, "The GeoSteiner software package for computing Steiner trees in the plane: An updated computational study," *Math. Program. Comput.*, vol. 10, no. 4, pp. 487–532, 2018, doi: 10.1007/s12532-018-0135-8.

[23] M. Brazil, M. Hendriksen, J. Lee, M. S. Payne, C. Ras, and D. Thomas, "An exact algorithm for the Euclidean k-Steiner tree problem," *Comput. Geometry*, vol. 121, Aug. 2024, Art. no. 102099, doi: 10.1016/j.comgeo.2024.102099.

[24] J. Lee, M. Brazil, C. Ras, and D. Thomas. (2025). *An Improved Exact Algorithm for the Euclidean k-Steiner Tree Problem*. [Online]. Available: http://ssrn.com/abstract=5127184

[25] J. E. Beasley, "OR-library: Distributing test problems by electronic mail," *J. Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069–1072, Nov. 1990. [Online]. Available: https://people.brunel.ac.uk/%7Emastjjb/jeb/info.html

[26] P. Winter and J. MacGregor Smith, "Steiner minimal trees for three points with one convex polygonal obstacle," *Ann. Oper. Res.*, vol. 33, no. 7, pp. 577–599, Jul. 1991, doi: 10.1007/bf02067243.

[27] G. Soothill, "The Euclidean Steiner problem," Dept. of Mathematical Science, Durham University, Durham, U.K., Tech. Rep., 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:123785548

[28] A. Beck and S. Sabach, "Weiszfeld's method: Old and new results," *J. Optim. Theory Appl.*, vol. 164, no. 1, pp. 1–40, Jan. 2015, doi: 10.1007/s10957-014-0586-7.

[29] L. M. Ostresh, "On the convergence of a class of iterative methods for solving the Weber location problem," *Operations Res.*, vol. 26, no. 4, pp. 597–609, Aug. 1978, doi: 10.1287/opre.26.4.597.

[30] H. W. Kuhn and R. E. Kuenne, "An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics," *J. Regional Sci.*, vol. 4, no. 2, pp. 21–33, Dec. 1962, doi: 10.1111/j.1467-9787.1962.tb00902.x.

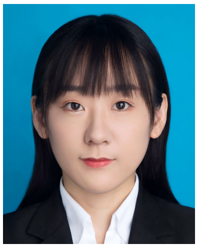
[31] H. W. Kuhn, "A note on Fermat's problem," *Math. Program.*, vol. 4, no. 1, pp. 98–107, Dec. 1973, doi: 10.1007/bf01584648.

[32] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *Int. J. Comput. Inf. Sci.*, vol. 9, no. 3, pp. 219–242, Jun. 1980, doi: 10.1007/bf00977785.

[33] S. Fortune, "Voronoi diagrams and Delaunay triangulations," in *Handbook of Discrete and Computational Geometry*. London, U.K.: Chapman & Hall, 2017, pp. 705–721.

[34] J. E. Beasley and F. Goffinet, "A Delaunay triangulation-based heuristic for the Euclidean Steiner problem," *Networks*, vol. 24, no. 4, pp. 215–224, Jul. 1994, doi: 10.1002/net.3230240404.

[35] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[37] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957, doi: 10.1002/j.1538-7305.1957.tb01515.x.



LIPING ZHANG received the B.E. and M.E. degrees in transportation engineering and vehicle operation engineering from Henan University of Science and Technology, Luoyang, China, in 2016 and 2019, respectively. She is currently pursuing the Ph.D. degree in computer science with Okayama University, Japan. Since 2019, she has been a Lecturer with the School of Automobile and Rail Transit, Luoyang Polytechnic, China. Her research interests include Euclidean Steiner tree problems and evolutionary computation.

TSUYOSHI MIGITA received the Ph.D. degree in information engineering from Hiroshima City University, Japan, in 2004. Since 2003, he has been an Assistant Professor with the Department of Computer Science, Okayama University, Japan. His research interests include computational geometry, numerical analysis, and computer vision.

NORIKAZU TAKAHASHI (Member, IEEE) received the B.E., M.E., and D.E. degrees in computer science and communication engineering from Kyushu University, Fukuoka, Japan, in 1991, 1993, and 1996, respectively. From 1996 to 1998, he was a Research Associate with Kyushu University. From 1998 to 2013, he was an Associate Professor with Kyushu University. Since 2013, he has been a Professor with the Department of Computer Science, Okayama University. His research interests include nonlinear systems, neural networks, machine learning, mathematical programming, multiagent systems, and graph theory. He is a Secretary of *Nonlinear Theory and Its Applications, IEICE*, and a member of the Editorial Board of *AEÜ–International Journal of Electronics and Communications*. He served as an Associate Editor for IEEE Transactions on Circuits and Systems—II: Express Briefs, from 2008 to 2009, and IEEE Transactions on Circuits and Systems—I: Regular Papers, from 2010 to 2012.