International Journal of Innovative Computing, Information and Control
Volume **15**, Number **2**, April **2019**
 ISSN 1349-4198
pp. 551–567

A SWARM-BASED META-HEURISTIC FOR RELAY NODES PLACEMENT IN WIRELESS SENSOR NETWORKS

Yihan Xu, Yutong Xiao and Qiuya Sun

College of Information Science and Technology
Nanjing Forestry University
No. 159, Longpan Road, Nanjing 210037, P. R. China
xuyihan@njfu.edu.cn; { 810840441; 917294933 }@qq.com

Received June 2018; revised October 2018

Abstract. *A Wireless Sensor Network (WSN) is a network consisting of wireless sensor nodes. There are usually requirements that need to be met when deploying a WSN, one being the placement of nodes. Due to placement requirements and limited node transmission range, a network might be partitioned initially. Therefore, additional relay nodes are added to the network to form an interconnected network. In this paper, the Minimum Relay nodes Placement (MRP) problem in WSNs is addressed. This problem addresses the placement of relay nodes: the minimum number needed and where the nodes should be placed. The problem is formulated as a Steiner Tree Problem with Minimum Steiner Points and Bounded Edge Length (STP-MSPBEL) problem which is NP-hard. In this paper, we present a variable dimension meta-heuristic based on Particle Swarm Optimization (PSO) called Multi-Space PSO (MSPSO) to address the problem. We tested MSPSO using randomly generated instances of the STP-MSPBEL problem of varying sizes and found that MSPSO is effective in addressing the STP-MSPBEL problem.*
Keywords: Relay nodes placement, Swarm-based meta-heuristic, Particle swarm optimization, Variable dimension, Steiner tree problem

1. **Introduction.** Wireless sensor nodes are programmable sensors that are usually battery-powered and capable of simple data processing and communicating with each other through wireless links to form a Wireless Sensor Network (WSN). A WSN eliminates the need for expensive and troublesome network cabling and makes placement cheap and flexible. There are usually requirements that need to be met when deploying a WSN. One such requirement is the placement of nodes. For example, sensing nodes must be placed at certain locations to allow them to gather data from the data sources, and a base station node might need to be placed in a special control room. Due to the placement requirements and limited node transmission range, a network might be partitioned initially. To form an interconnected network, additional nodes called relay nodes are necessary. While wireless sensor nodes are generally regarded as inexpensive devices, deploying a network with a huge number of nodes or a network to cover a large geographic area still incurs considerable cost; therefore, proper planning needs to be done prior to network deployment to minimize cost while satisfying the other requirements.

WSNs can be categorized based on their structure. Some networks are single-tiered, while others are multi-tiered/hierarchical. Single-tiered networks consist entirely of a single type of nodes. In multi-tiered/hierarchical networks, there are different types of nodes. Some nodes have basic capabilities while others have enhanced capabilities such as increased transmission range and faster data processing speed. The Minimum Relay nodes

DOI: 10.24507/ijicic.15.02.551

Placement (MRP) problem can be categorized according to the dimension of the space nodes to be placed. In the two-dimensional problem, nodes are to be placed in the two-dimensional Euclidean space $R2$ (plane) while in the three-dimensional problem, nodes are to be placed in the three-dimensional Euclidean space $R3$ (space). The MRP problem can be further categorized to unconstrained and constrained placement. In unconstrained placement, relay nodes can be placed anywhere in space. In contrast, in constrained placement, relay nodes are to be placed only at certain locations. Both versions of the problem find their uses in different applications.

In this paper, we consider the unconstrained MRP problem in two-dimensional WSNs and attempt to solve the problem with a novel meta-heuristic based on Particle Swarm Optimization (PSO) [1]. There are two contributions from our paper as follows.

1) A variable-dimension meta-heuristic based on PSO called the Multi Space PSO (MS-PSO) is proposed for use in problems where candidate solutions could be of different lengths (having different numbers of dimensions).
2) The MRP problem in two-dimensional WSNs is modeled as a Steiner Tree Problem with Minimum Steiner Points and Bounded Edge Length (STP-MSPBEL) problem and addressed by first transforming it into a different problem that we refer to as the dual problem.

This paper is organized as follows. First, related works are discussed in Section 2. In Section 3, we describe the problem that we attempt to solve and our proposed optimization method. In Section 4, our proposed optimization method is used to solve several randomly generated instances of the STP-MSPBEL problem to demonstrate its effectiveness. Finally, we offer conclusion in Section 5.

2. **Related Works.** In this paper, the Minimum Relay nodes Placement (MRP) problem in two-dimensional WSNs is approached. We formulate the problem as a Steiner Tree Problem with Minimum Steiner Points and a Bounded Edge Length (STP-MSPBEL) problem. The STP-MSPBEL problem is less well studied compared to the regular/vanilla Steiner Tree Problem (STP). Research studies in the literature that have approached the STP-MSPBEL problem and other similar problems can be categorized into two main categories. In the first category, works are very similar to one another as they are all based on the Minimum Spanning Tree (MST) heuristic. In the second category, meta-heuristics are used to address the problem.

Several works belonging to the first category: in [2], the MST heuristic was proposed for the STP-MSPBEL problem, and it was shown that it has a lower bound or worst-case performance of 5. However, the authors of [3] showed that the algorithm in [2] is actually a 4-approximation algorithm. In [4], the Terminal Steiner Tree with Bounded Edge Length (TSTBEL) problem, which is similar to the STP-MSPBEL problem, was introduced. In [5], the authors proposed two heuristics for the STP-MSPBEL problem. One has a performance ratio of 3, while the other has a performance ratio of 2.5. The 3-approximate algorithm is similar to the MST heuristic with the exception that degree-3 Steiner points are added to the tree for every subset of three terminals a, b, and c if there exists a point s within the three terminals prior to applying the MST heuristic. In [6], the 1-connected and 2-connected MRP problems were formulated. The 1-connected MRP problem (MRP-1) is similar to the TSTBEL problem. The authors employed the minimum disc cover approximation scheme [7] to initially find an approximate minimum disc cover set and then interconnect the centers of the disks using the 2.5-approximate STP-MSPBEL heuristic. In [8], the single-tiered and two-tiered relay node placement problems were discussed. The single-tiered version of the problem is essentially a more generalized version of the STP-MSPBEL problem in that relay nodes and terminal points have different transmission ranges. The MST heuristic was proposed for use in the single-tiered relay node placement problem. The authors proved that the MST heuristic is actually a polynomial-time 7-approximation algorithm for the problem. Two-tiered sensor relay placement was considered in [9,10], in which constant-factor approximation algorithms for several versions of the problem are proposed. In [11], the authors provide an extensive survey on optimization algorithms for cloud environments based on three popular meta-heuristic techniques: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithms (GAs).

Works belonging to the second category: in [12], a stochastic algorithm based on the simulated annealing [13] meta-heuristic was proposed to solve the STP problem. As with any other meta-heuristic, there is no guarantee that the optimal solution is ever found. However, the authors found an interesting aspect, which is that the time needed to compute a good solution is much less than that required by the exact algorithm by Cockayne and Hewgill. In [14], a swarm-based meta-heuristic emulating the concept of a fish swarm searching for food is proposed to solve the STP problem. The authors encode particles as trees. However, we found that the problem solved is not really the STP problem but the Multiple-Destination Routing (MDR) problem. Similar to [14], in [15], the authors proposed solving the MDR problem with PSO. The authors introduced mutation to introduce new tree structures to the particle population. The MDR problem was approached instead of the STP problem. In [16], a method adopted in cognitive radio wireless sensor networks was proposed.

In an instance of the STP-MSPBEL problem, candidate solutions may have different lengths. In the classical PSO, all candidate solutions for an instance of the problem to be solved have the same length. There are few works in the literature on solving problems where candidate solutions can be of different lengths with PSO. However, the idea of applying PSO to solving these problems has been proposed before in other problems. In [17], a variable-dimension optimization approach based on PSO was proposed to tackle the Unit Commitment Problem (UCP). In [18], Dimension-Adaptive PSO (DA-PSO) was proposed and demonstrated to solve the Weibull mixture model density estimation problem.

3. MSPSO to Address the STP-MSPBEL Problem.

3.1. Problem formulation.

In the MRP problem in two-dimensional wireless sensor networks, a set of nodes (their locations in two-dimensional Euclidean space) consisting of both base station and sensing nodes is given. This constitutes the requirement of the placement of nodes during a WSN deployment. Sensing nodes are to collect data from sites and send them to base station nodes. However, due to placement requirements, a sensing node could be placed far from its base station node. As a result, relay nodes might be needed to help relay packets. The following assumptions are made for our study.

1) There is only one base station node in the entire network.
2) Base station and sensing nodes can relay data packets.
3) All nodes have the same maximum transmission range.
4) Two nodes can communicate if they are within transmission range of each other.

These assumptions are reasonable as we only consider the case of one single base station in the sensing area. Meanwhile, for convenience initial deploying node randomly, all the nodes are equipped with the same transceiver, therefore they have the same maximum transmission range and only when they are within the transmission range of each other, the communication can occur. We refer to both base station and sensing nodes simply as demand points for the remainder of this paper. Based on these assumptions, the MRP problem can then be formulated as a Steiner Tree Problem with Minimum Steiner Points and Bounded Edge Length (STP-MSPBEL) [2] problem. In the STP-MSPBEL problem, given a set V of points and a constant R, one interconnects nodes in V with a graph such that 1) the number of points added (called Steiner points) to interconnect all points in the graph (demand and Steiner points) is minimal, and 2) the length of the edge between any two points is bounded by the constant R. In the STP-MSPBEL problem, there is an edge between two points a and b in the graph (i.e., a and b are interconnected) if and only if the Euclidean distance between point a and point b is less than R, i.e., $d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \leq R$, where $(x_a - y_a)$ and $(x_b - y_b)$ are the coordinates of points a and b respectively. In the MRP problem, the value of R in the STP-MSPBEL problem corresponds to the maximum transmission range of the nodes. The STP-MSPBEL problem is NP-hard, and the decision version of the problem was proven to be NP-complete in [2].

3.2. **Multi-Space Particle Swarm Optimization (MSPSO).** The Steiner Tree Problem with Minimum Steiner Points and Bounded Edge Length (STP-MSPBEL) is a variable-dimension problem. To solve such a problem, we propose the use of a variable-dimension meta-heuristic based on Particle Swarm Optimization (PSO). We name the proposed method Multi-Space PSO (MSPSO).

MSPSO is extended from PSO. The difference between MSPSO and the classical PSO is that, in the classical PSO, the search space is of a fixed number of dimensions, while in MSPSO, the search space is the "universe" and consists of different search spaces of different numbers of dimensions. In this paper, MSPSO is applied to solving the STP-MSPBEL problem; hence, the entire process is called MSPSO-STP-MSPBEL. To aid in understanding, an overview of the entire MSPSO-STP-MSPBEL process is shown in Figure 1.

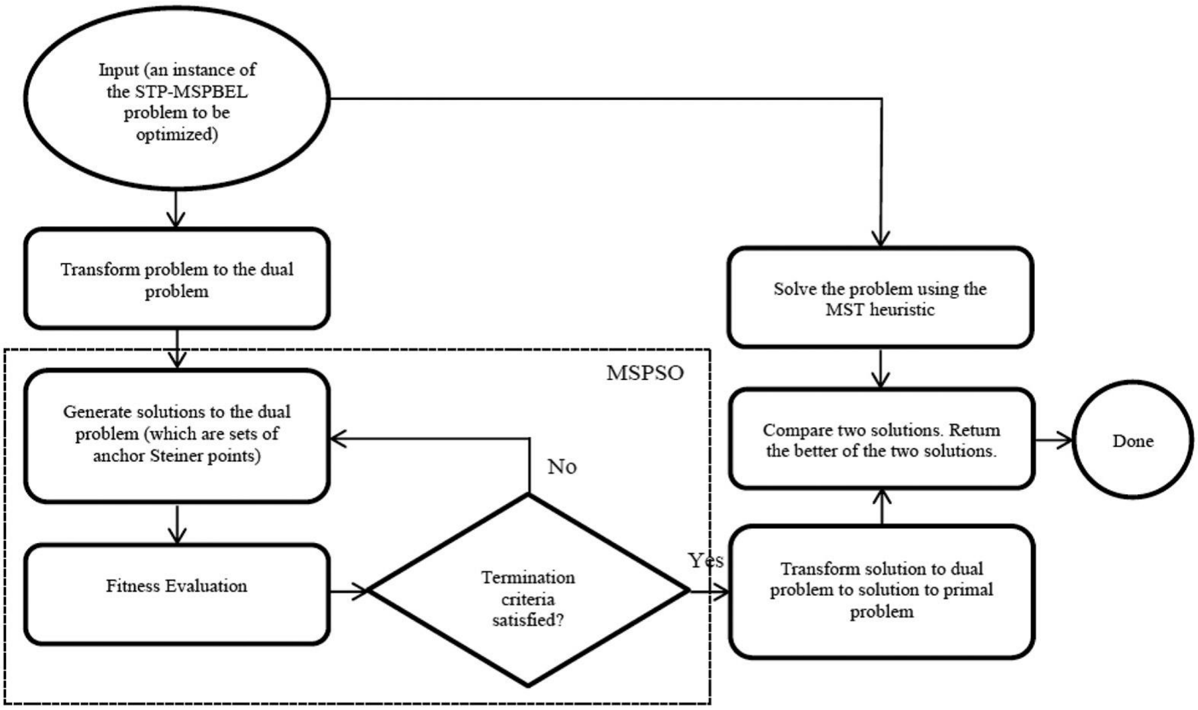


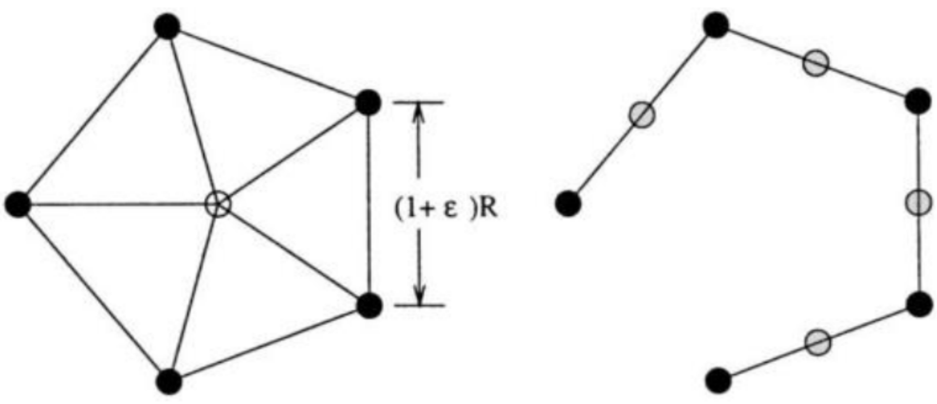
Figure 1. Overview of the MSPSO-STP-MSPBEL process

The MSPSO-STP-MSPBEL proceeds as follows. Given an instance of STP-MSPBEL problem, we first try to locate the degree-3 or higher Steiner points. For the sake of convenience, we refer to degree-3 or higher Steiner points simply as degree-3+ Steiner points. To locate the degree-3+ Steiner points, the notion of anchor Steiner points is used. Anchor Steiner points are Steiner points, and together with demand points, they are used as input to the Minimum Spanning Tree (MST) construction process to construct a candidate solution to the STP-MSPBEL input problem instance. In order words, anchor Steiner points are candidates for degree-3+ anchor Steiner points. When the anchor Steiner points are located, the remaining non-anchor/regular Steiner points can easily be determined by Steinerizing the edges in the MST formed by the demand points and anchor Steiner points. We name them anchor Steiner points because they serve as anchors to allow other regular Steiner points of degree-2 to be determined.

The degree of a node is the number of edges incident with that node. In the classical STP problem, Steiner points are of degree 3. According to [3], in the STP-MSPBEL problem, Steiner points can be of degree 2 or 3. However, in Theorem 3.1 below, we state that, for the STP-MSPBEL problem, Steiner points can have a degree of at most 5.

Theorem 3.1. *In the STP-MSPBEL problem, Steiner points can have a degree of at most 5.*

Proof: Before we prove that Steiner points can have a degree of at most 5 in the STP-MSPBEL problem, we first analyze a similar problem in [19]. Figure 2 was used in [20] as an example to prove that the Steinerized MST has a worst-case performance of 4. In the figure, the black colored circles are the demand points. In Figure 2(a), the empty circle is the Steiner point of the optimal solution, and ε is a small positive real number such that the distance from the center to each vertex is within R. In Figure 2(b), the empty circles are the Steiner points of the Steinerized MST. As mentioned in [19], the tree in Figure 2(a) is the optimal solution; it must be a valid solution. We see that the Steiner points in the optimal tree have a degree of 5. In this paper, Figure 3 is used to prove that there are no Steiner points of degree 6 in the STP-MSPBEL problem. Figure 3 presents a regular hexagon. In this figure, the white colored circles are the demand points while the gray colored circle is the Steiner point. Initially, we suppose that there is a Steiner point of degree 6 as shown in Figure 3(a). However, in a regular hexagon, the distance between any two vertexes is the same. In that case, the Steiner point is redundant, and thus we arrive at a contradiction. By removing the Steiner point, we could have Figure



(a) (b)

FIGURE 2. An input instance to the STP-MSPBEL to prove that the MST heuristic has a worst-case performance of 4 in [20]

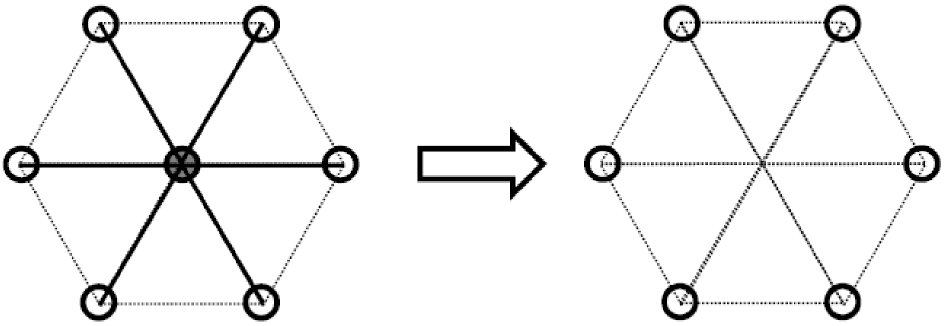

(a) (b)

FIGURE 3. A Steiner point in STP-MSPBEL has a degree of at most 5.

3(b). Therefore, we can prove that, in STP-MSPBEL, Steiner points have a degree of at most 5.

The minimum number of anchor Steiner points ($n_{\min}$) needed to steinerize an edge of length l is given in Equation (1). In MSPSO, degree-3+ anchor Steiner points are searched for using anchor Steiner points. A particular solution (a set of anchor Steiner points) is only a candidate for the optimal solution. The ideal candidate solution is such that anchor Steiner points are coincident with the degree-3+ anchor Steiner points in the optimal solution. In a candidate solution to an instance of the STP-MSPBEL problem, the total number of anchor Steiner points is given in Equation (2).

$$n_{\min} = \begin{cases} \dfrac{l(e)}{R} - 1 & \text{if } R < l(e) \\ 0 & \text{if } R \geq l(e) \end{cases} \tag{1}$$

$$n_{\textit{steiner points}} = n_{\textit{anchor steiner points}} + n_{\textit{regular steiner points}} \tag{2}$$

By determining the solution to an instance of the STP-MSPBEL problem by first locating the degree-3+ anchor Steiner points, we in fact have converted this problem to another problem, which is referred to as the dual problem.

Definition 3.1. The primal problem (the STP-MSPBEL problem)

Given a set V of terminal points and a constant R, an interconnected graph with the following properties is sought: 1) the number of added points initially not in set V (called anchor Steiner points) should be minimal, and 2) the length of the edge between any two points is bounded by R.

Definition 3.2. The dual problem

Given a set V of terminal points, the graph with minimum cost that is determined by Equation (3) is sought. Extra nodes not in set V (which we call anchor Steiner points in this paper) may be added to the graph to reduce cost.

$$cost = \sum_{e \in E} \left(\frac{l(e)}{R} + 1 \right) + n_{\textit{anchor steiner points}} \tag{3}$$

where E is the set of edges in the MST spanned by the demand points and anchor Steiner points, $l(e)$ is the length of an edge $e \in E$, and $n_{\textit{anchor steiner points}}$ is the number of anchor Steiner points in the tree.

Now that the notion of anchor Steiner points is introduced, we state in Theorem 3.2 that the fitness of a candidate solution is varied only by anchor Steiner points.

Theorem 3.2. *A candidate solution's fitness is varied only by anchor Steiner points.*

Proof: Regular Steiner points are determined by Steinerizing edges in the minimum spanning tree formed by demand points and anchor Steiner points. Because demand points are fixed (their quantity and their position are given), only the anchor Steiner points (their quantity and their position) can affect the minimum spanning tree formed and thus affect the fitness of a candidate solution.

In MSPSO-STP-MSPBEL, a particle is encoded as a set of points in two-dimensional Euclidean space, and a point is represented with a pair of numbers that corresponds to their coordinates in the two-dimensional Euclidean space. For example, a candidate solution $\{(3,4),(6,8)\}$ represents that there are two anchor Steiner points and that they are located at coordinates $(3,4)$ and $(6,8)$, respectively. Another candidate solution $\{(2,5),(6,7),(4,6)\}$ specifies that there are three anchor Steiner points and that they are located at coordinates $(2,5)$, $(6,7)$, and $(4,6)$, respectively.

A candidate solution to an instance of the STP-MSPBEL problem could have between zero and infinite anchor Steiner points. To avoid an infinite search, the range limit of the number of dimensions of the spaces is first determined. The maximum value of this range is set according to the following observations: the MST heuristic can be used to get an approximate to the optimal solution to a given instance of the STP-MSPBEL problem. We try to solve the dual problem to solve the primal problem. In the dual problem, we are to locate degree-3+ anchor Steiner points in the primal problem using anchor Steiner points. Because $n_{steiner\ points} \geq n_{anchor\ steiner\ points}$, there is no point in allowing a candidate solution to the dual problem to have more anchor Steiner points than the estimate of the number of anchor Steiner points in the primal problem (obtained using the MST heuristic) since $n_{steiner\ points} = n_{anchor\ steiner\ points} + n_{regular\ steiner\ points}$; the resulting candidate solution to the primal problem will have more anchor Steiner points than the solution obtained using the MST heuristic. Because $n_{steiner\ points}$ obtained from using the MST heuristic will be quite high, this will result in a long runtime since candidate solutions might initially have long lengths. That is because candidate solutions are searched for in unnecessary spaces of high dimension. To help reduce the runtime of our optimization method, the following point is used: in the Steiner Tree Problem (STP), a full Steiner topology/tree has at most $k = N - 2$ anchor Steiner points, where N is the number of demand points in an input instance. Because anchor Steiner points are used to search for these points, it is intuitive that we limit the number of dimensions allowed for a particular space to at most $N-2$ anchor Steiner points. The minimum of this range is set to 0. The range of the number of dimensions of a particular space ($n_{range\ of\ a\ particular\ space}$) is thus as given in Equation (4).

$$n_{range\ of\ a\ particular\ space} = [0, \min(N-2, n_{MST\ heuristic\ on\ the\ demand\ points})] \tag{4}$$

where N is number of demand points in an input instance, $n_{MST\ heuristic\ on\ the\ demand\ points}$ is number of Steiner points determined by the MST heuristic on the demand points.

1) Initialization of Particles

After the range of the number of dimensions of spaces is determined, we generate particles that randomly vary in the number of dimensions. Then, for each particle in the swarm, a number r within the range of dimensions of the spaces (Equation (4)) is randomly generated. Each particle then generates r anchor Steiner points. We state in Theorem 3.3 below that anchor Steiner points should be located inside the convex hull formed by the demand points. Hence, anchor Steiner points are generated randomly such that they are located within the convex hull formed by the demand points. The velocity of the particles is set to 0 or initialized randomly.

Theorem 3.3. *All Steiner points (anchor and non-anchor) should be located within the convex hull formed by the demand points.*

Proof: There are three points in a triangle; if a fourth point is outside the triangle formed by the three points, a shorter minimum spanning tree could be formed by placing the fourth point along the edge of the triangle or inside the triangle. One method is to place the fourth point along one of the edges of the triangle. Meanwhile, this can be easily extended to other given numbers of points.

2) Fitness Evaluation

Because we address the primal problem using the dual problem, evaluation of a particle's fitness becomes complex. To help understand how this process works, we first review the course of action of a particle as shown in Figure 4. Initially, a particle consists of an empty set. During the initialization phase, each particle generates a random number r. Each particle then randomly generates r Steiner points and adds them to the set. After an iteration, the fitness values of the particles are evaluated. The cost function to be used for evaluation of particles is as shown in Equation (2). By emulating/mimicking the MST construction process and recording the number of points needed to Steinerize an edge in the MST in the process, we can determine the total number of anchor Steiner points needed for the primal problem. The detailed algorithm for determining the total number of anchor Steiner points needed for the primal problem is presented in Algorithm 1.

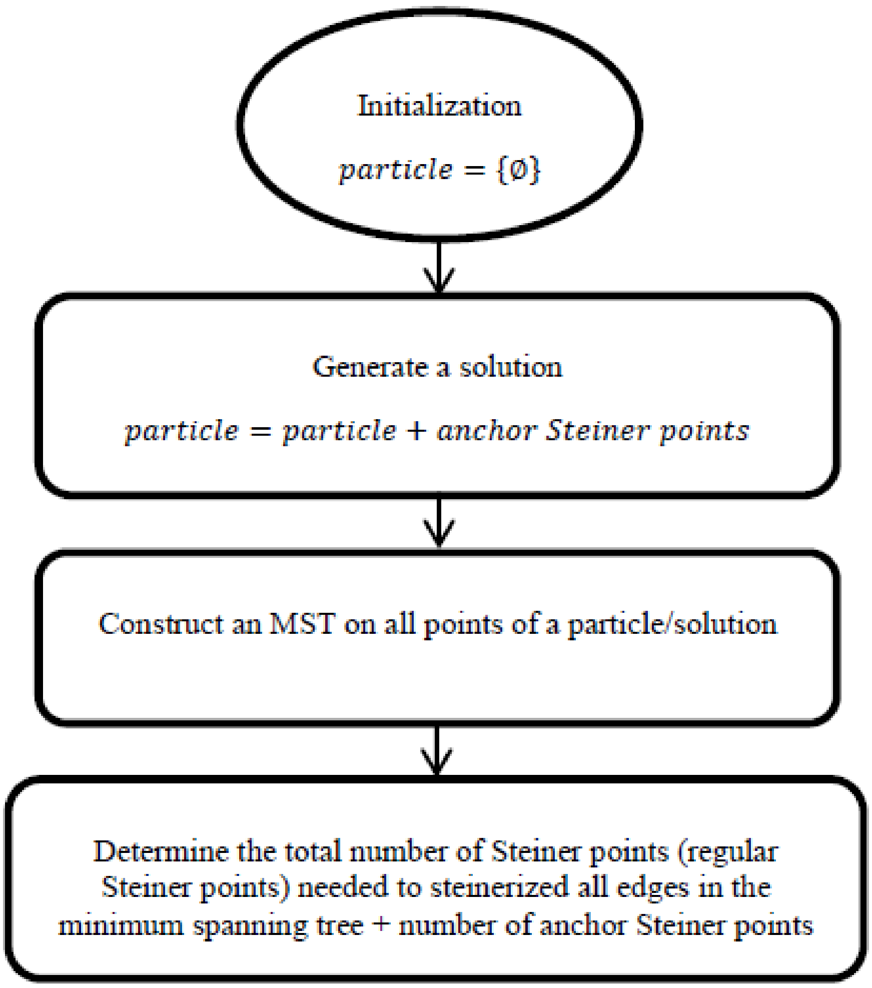


FIGURE 4. The course of action of a particular particle

Given: 1) demand points
2) anchor Steiner points
Output: The total number of anchor Steiner points in a candidate solution for the primal problem

1. Assign every node v_i to different groups, $grp_i : grp_i \leftarrow v_i$, $0 < |V| - 1$
2. Combine groups that are interconnected into one resulting in a less number of n groups, $n \leq |V|$.
3. Initial $numNonAnchorSteinerPointsNeeded \leftarrow 0$.
4. ***While*** $(n > 1)$ // tree not spanning
   a. Let grp_0 be the source group: grp_{src}. Get the group among $n - 1$ other groups with the shortest distance to grp_{src}, call this group the destination group, grp_{dst}.
   b. Calculate the number of anchor Steiner points needed to steinerize the shortest edge between grp_{src} and grp_{dst}. Let this answer be k.
   c. $numNonAnchorSteinerPointsNeeded \leftarrow numNonAnchorSteinerPoints Needed + k$
   d. $n \leftarrow n - 1$ // there is now one group less since we combined grp_{src} and grp_{dst}
5. ***End while***
6. *Return* $numNonAnchorSteinerPointsNeeded$ + *number of anchor Steiner points*

Algorithm 1. Algorithm for determining the total number of anchor Steiner points in a candidate solution for the primal problem

3) Updating of the Space That Particles Belong to

After an iteration of the classical Particle Swarm Optimization (PSO) meta-heuristic, the best position obtained by the entire swarm (*gbest*) and particles' personal best locations (*pbest*) are updated as necessary. In MSPSO-STP-MSPBEL, we want the particles to behave similarly. In this section, we discuss how the number of anchor Steiner points a particle specifies is updated.

The number of anchor Steiner points in a particle is discrete/integer-valued. A concept similar to the one employed in the Jumping Frog Optimization (JFO) [20] designed for discrete optimization problems is adopted. A random number is generated in the range $[0, 1)$ and the action to take depends on the range the random number falls into. In the range $[0, w)$, a particle stays in the current space or explores the neighboring space. In the range $[w, w + c_1)$, a particle gets attracted to and moves towards the space *pbest* is located in. In the range $[w + c_1, 1)$ a particle gets attracted to and moves towards the space *gbest* is located in. The number of anchor Steiner points thus contributes to one dimension of the problem. In the $[0, w)$ range, another random variable in $[0, 1)$ is generated to see if it should remain in the current space or explore the neighboring space. If this variable falls within $[0, PROB\_EXPLORE\_NEIGHBORING\_SPACE)$, then the particle explores the space of one dimension lower or higher with equal probability; otherwise, the particle remains in the same space. The reason for the reuse of the same parameters is as follows: in the classical PSO, particles are attracted towards new positions in space in the directions of *pbest* and *gbest*, and in MSPSO, we want the particles to behave similarly with respect to the variable number of anchor Steiner points.

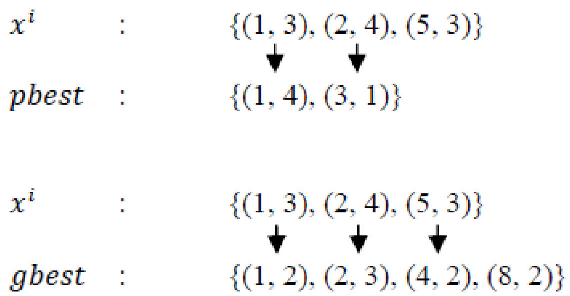


FIGURE 5. Component-wise updating of variables

4) Updating of Anchor Steiner Point Velocities and Positions

During the velocity update of particles, the current position v^i could have a different number of dimensions compared to *pbest* or *gbest*. However, particles from different spaces can share information with each other. Hence, we employ component-wise updating similar to that proposed in [17] for the unit commitment problem. Figure 5 shows the component-wise updating of variables. Component-wise updating is used because particles share information with each other only between identical dimensions. Particles do not anticipate or borrow information from alien or unknown space dimensions.

The velocity of a particle is updated in two steps. First, we update it towards *pbest*. Then in step two, it is updated toward *gbest*. The position of a particle is updated by simply adding its previous position to the updated velocity.

5) Transform Solution of the Dual Problem to Solution of the Primal Problem

The MSPSO meta-heuristic is only used to determine the anchor Steiner points. Anchor Steiner points are approximates of degree-3+ Steiner points in the optimal solution for the input instance of the STP-MSPBEL problem. The remaining non-anchor/regular Steiner points still need to be determined. The detailed algorithm to determine the non-anchor Steiner points is described formally in Algorithm 2.

4. **Results and Discussions.** We implemented MSPSO-STP-MSPBEL in C/C++ in Ubuntu 15.10. To demonstrate the feasibility of the proposed method, we generate random instances of the STP-MSPBEL problem, use our program to obtain approximate solutions to these instances, and compare our results with those obtained by the Minimum Spanning Tree (MST) heuristic and approximation algorithm. Random instances of the problem of varying input size (number of demand points) were generated with 2 instances for each input size. In all instances, we assume node placement in a square region of 1000 units by 1000 units. In Particle Swarm Optimization (PSO), the number of particles (swarm size) determines how many different agents are used to solve a given input instance. If more particles are used, the probability of finding a better solution increases. Due to the inherent complexity of the STP-MSPBEL problem, 2000 particles were used in all instances. To allow sufficient iterations for particles to search for a good solution, the MSPSO-STP-MSPBEL was programmed to terminate after 500 iterations. As PSO is based on the concept of iterative search, if a low value of number of iterations is used, the particles might not converge by the time the search process is terminated. In all instances, we assume that the transmission range of all nodes $R = 25$ units. The values of w, $c1$, and $c2$ are set as 0.2, 0.35, and 0.45, respectively. We run the program on a system with an Intel Core i5-5200U processor and 8 GB of RAM.

In general, we found that our program can address the generated instances with better results compared to those obtained by using the MST heuristic. For practical input instance size, we found that our program has a fast runtime. For instance, referring to Table

Given: 1) demand points
2) anchor Steiner points
3) the transmission range of nodes, R
Output: A minimal spanning tree involving the given nodes.

1. Group nodes that are interconnected into a larger group.
2. ***While*** (number of groups, $(n > 1)$) // tree not spanning
   a. Select the first group as the starting group grp_{src}.
   b. Find the group closest in Euclidean distance to grp_{src} and name it the destination group grp_{dst}.
   c. Get the shortest edge between grp_{src} and grp_{dst}. Name the point of the edge in grp_{src} as pt_{src} and the point of the edge in grp_{dst} as pt_{dst}.
   d. ***While*** $(d(pt_{src}, pt_{dst}) > R)$
      i. $diff_x = pt_{dst} \cdot x - pt_{src} \cdot x$;
      ii. $diff_y = pt_{dst} \cdot y - pt_{src} \cdot y$;
      iii. $ratio = R \div d(pt_{src}, pt_{dst})$;
      iv. Add $(pt_{src} \cdot x + ratio \times diff_x,\ pt_{src} \cdot y + ratio \times diff_y)$ as a non-anchor Steiner point
   e. ***End while***
   f. $n \leftarrow n - 1$ // there is now one less group since we combined grp_{src} and grp_{dst}
3. ***End while***

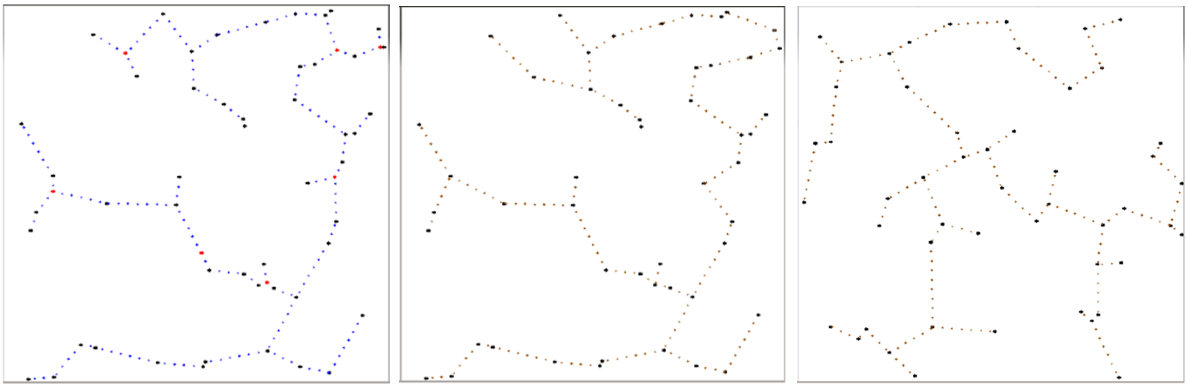
Algorithm 2. Algorithm for determining the non-anchor Steiner points

3, for an input instance size of 200 demand points, our program was able to obtain a runtime of just over 3 hrs. Runtime is an important criterion in this paper. If one algorithm takes an unacceptable amount of time to search for a good solution, it becomes useless. The lower runtime also indicates the lower computation complexity. From Tables 1, 2 and 3, we observed that the runtime accelerates with increasing input instance size. This is because the fitness evaluation process of a particle requires that the MST construction process be emulated/mimicked, and this takes (n^2) time, where n is the number of nodes. In our program, we employed the traditional method of updating the MST when anchor points are added, removed, or changed in position. If a more efficient method of updating the MST is used, such as dynamic MST construction, better runtime performance is possible. However, implementing such an efficient MST updating process algorithm is not the focus of this paper. Meanwhile, we compared our program with an approximation algorithm, in which we first select a critical node from the uncovered sensor nodes and then determine the location of the relay node based on the principle of preferring to cover the sensor node closer to the critical node. We found that our program has better performance compared to the approximation algorithm. Moreover, PSO is an iterative search technique, and thus one should not compare its runtime to simple heuristics such as the MST heuristic. For example, with no termination criteria, PSO continues to search for the optimal solution continuously, even if it has already found it.

4.1. **50 demand points.**

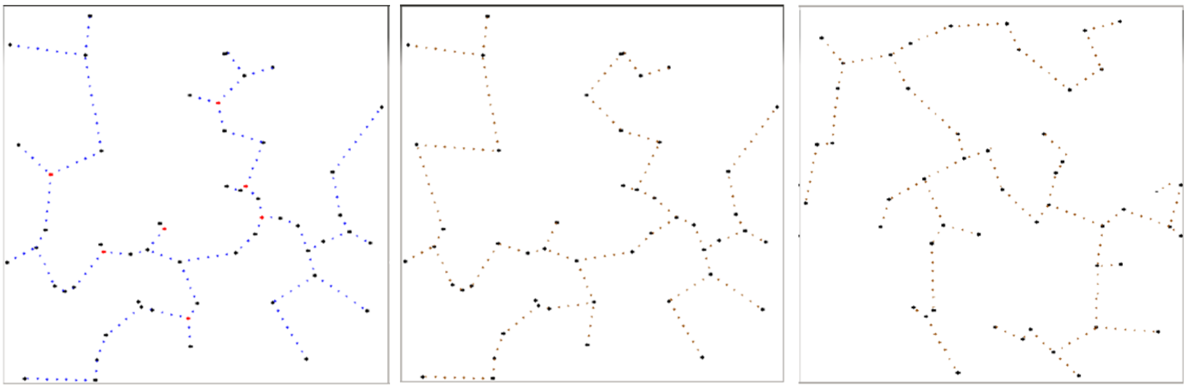
1) Instance #1

Figure 6(a) shows the result obtained by using our program, Figure 6(b) shows the result obtained by using the MST heuristic, and Figure 6(c) gives the result obtained by using the approximation algorithm. In these figures, black dots are demand points, red



(a) (b) (c)

FIGURE 6. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm



(a) (b) (c)

FIGURE 7. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm

dots are anchor Steiner points, blue dots are regular Steiner points, and brown dots are Steiner points obtained by using the MST heuristic.

2) Instance #2

Figure 7(a) shows the result obtained by using our program, Figure 7(b) shows the result obtained by using the MST heuristic, and Figure 7(c) gives the result obtained by using the approximation algorithm. In these figures, black dots are demand points, red dots are anchor Steiner points, blue dots are regular Steiner points, and brown dots are Steiner points obtained by using the MST heuristic.

3) Summary

Table 1 shows a summary of the three input instances with 50 demand points.

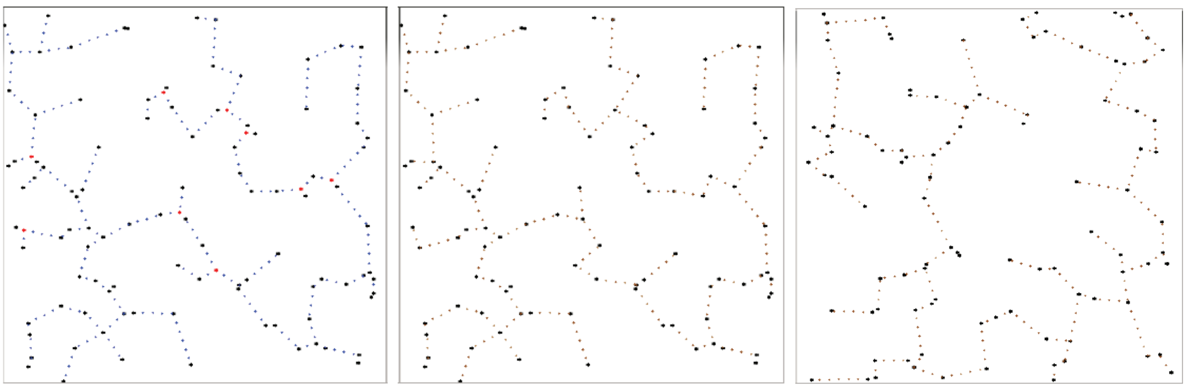
4.2. **100 demand points.**

1) Instance #1

Figure 8(a) shows the result obtained by using our program, Figure 8(b) shows the result obtained by using the MST heuristic, and Figure 8(c) gives the result obtained by using the approximation algorithm. In these figures, black dots are demand points, red

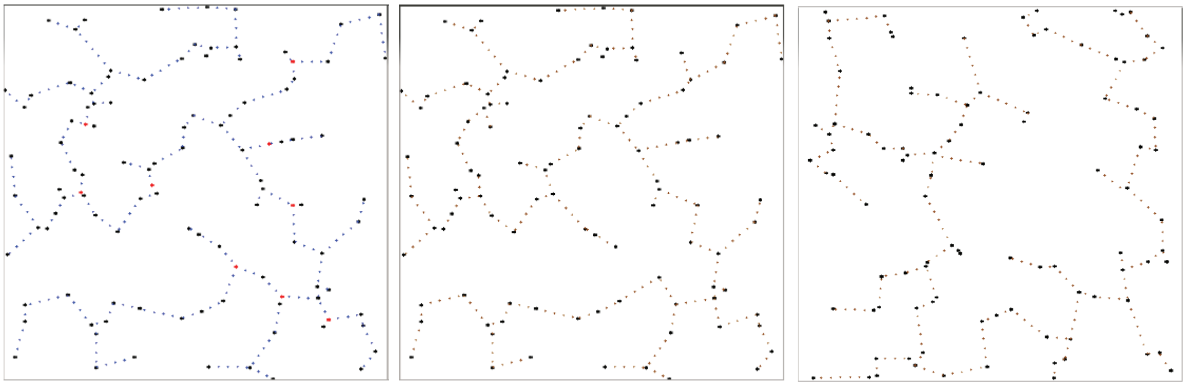
TABLE 1. Summary of three instances with 50 demand points

Instance	1	2
Number of relay nodes in the solution obtained by our program	159	164
Number of relay nodes in the solution obtained by the MST heuristic	166	173
Number of relay nodes in the solution obtained by the approximation algorithm	165	171
Improvement	4.22% and 3.63%, respectively to MST heuristic and approximation algorithm	5.2% and 4.09%, respectively to MST heuristic and approximation algorithm
Number of anchor Steiner points by our program	7	7
Runtime of our program	421.5 s	463.3 s
Runtime of the MST heuristic	474.7 s	522.7 s
Runtime of the approximation algorithm	465.2 s	499.1 s
Runtime improvement	11.2% and 9.39%, respectively to MST heuristic and approximation algorithm	11.36% and 7.17%, respectively to MST heuristic and approximation algorithm



(a) (b) (c)

FIGURE 8. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm



(a) (b) (c)

FIGURE 9. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm

TABLE 2. Summary of three instances with 100 demand points

Instance	1	2
Number of relay nodes in the solution obtained by our program	212	228
Number of relay nodes in the solution obtained by the MST heuristic	221	234
Number of relay nodes in the solution obtained by the approximation algorithm	219	230
Improvement	4.07% and 3.19%, respectively to MST heuristic and approximation algorithm	2.98% and 0.86%, respectively to MST heuristic and approximation algorithm
Number of anchor Steiner points by our program	9	9
Runtime of our program	2092.0 s	1962.2 s
Runtime of the MST heuristic	2346.6 s	2314.4 s
Runtime of the approximation algorithm	2241.7 s	2246.1 s
Runtime improvement	10.84% and 6.68%, respectively to MST heuristic and approximation algorithm	15.21% and 12.64%, respectively to MST heuristic and approximation algorithm

dots are anchor Steiner points, blue dots are regular Steiner points and brown dots are Steiner points obtained by using the MST heuristic.

2) Instance #2

Figure 9(a) shows the result obtained by using our program, Figure 9(b) shows the result obtained by using the MST heuristic, and Figure 9(c) gives the result obtained by using the approximation algorithm. In these figures, black dots are demand points, red dots are anchor Steiner points, blue dots are regular Steiner points and brown dots are Steiner points obtained by using the MST heuristic.

3) Summary

Table 2 shows a summary of the three input instances with 100 demand points.

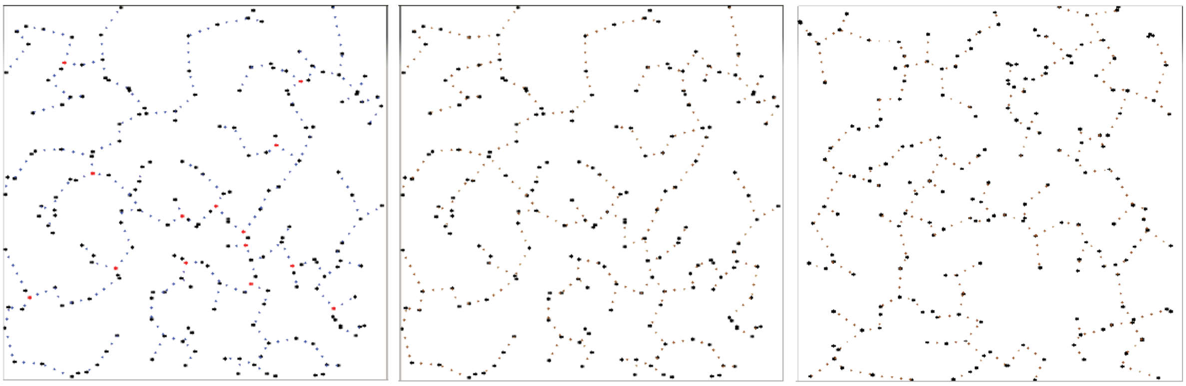
4.3. **200 demand points.**

1) Instance #1

Figure 10(a) shows the result obtained by using our program, Figure 10(b) shows the result obtained by using the MST heuristic, and Figure 10(c) gives the result obtained by using the approximation algorithm. In these figures, black dots are demand points, red dots are anchor Steiner points, blue dots are regular Steiner points and brown dots are Steiner points obtained by using the MST heuristic.

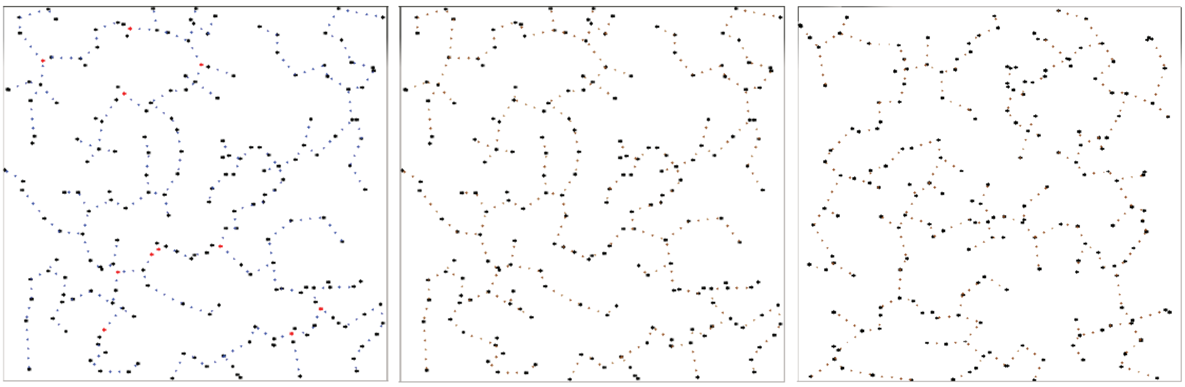
2) Instance #2

Figure 11(a) shows the result obtained by using our program, Figure 11(b) shows the result obtained by using the MST heuristic, and Figure 11(c) gives the result obtained by



(a) (b) (c)

FIGURE 10. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm



(a) (b) (c)

FIGURE 11. (color online) (a) Result obtained using our program; (b) result obtained using the MST heuristic; (c) result obtained using the approximation algorithm

Table 3. Summary of three instances with 200 demand points

Instance	1	2
Number of relay nodes in the solution obtained by our program	267	277
Number of relay nodes in the solution obtained by the MST heuristic	281	286
Number of relay nodes in the solution obtained by the approximation algorithm	278	280
Improvement	4.98% and 3.96%, respectively to MST heuristic and approximation algorithm	3.15% and 1.07%, respectively to MST heuristic and approximation algorithm
Number of anchor Steiner points by our program	14	11
Runtime of our program	9074.5 s	11060.3 s
Runtime of the MST heuristic	12456.4 s	13146.1 s
Runtime of the approximation algorithm	10234.1 s	11902.4 s
Runtime improvement	27.14% and 11.33%, respectively to MST heuristic and approximation algorithm	15.86% and 7.08%, respectively to MST heuristic and approximation algorithm

using the approximation algorithm. In these figures, black dots are demand points, red dots are anchor Steiner points, blue dots are regular Steiner points, and brown dots are Steiner points obtained by using the MST heuristic.

3) Summary

Table 3 shows a summary of the three input instances with 200 demand points.

5. **Conclusion.** In this paper, the single-tiered minimum relay nodes placement problem in two-dimensional Euclidean space is approached. The problem can be formulated as a Steiner Tree Problem with Minimum Steiner Points with Bounded Edge Length (STP-MSPBEL). In the literature, most algorithms proposed to solve the problem are based on the Minimum Spanning Tree (MST) heuristic. In this paper, a novel variable-dimension meta-heuristic based on Particle Swarm Optimization (PSO) was proposed to address the problem. The optimization method was put to the test for several randomly generated instances of the problem. We found that the method is effective in obtaining good approximate solutions to those instances of the problem. For the majority of cases, a better approximate solution was obtained compared to that obtained from the MST heuristic.

Acknowledgement. This work is partially supported by Introduction of High-level Talents and Overseas Returnee's Scientific Fund in Nanjing Forestry University (No: GXL015) and Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No: 16KJB510016). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of ICNN'95 – International Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.
[2] G. Lin and G. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, *Information Processing Letters*, vol.69, no.2, pp.53-57, 1999.
[3] D. Chen, D. Z. Du, X. D. Hu, G. H. Lin, L. Wang and G. Xue, Approximations for Steiner trees with minimum number of Steiner points, *Theoretical Computer Science*, vol.262, no.1, pp.83-99, 2001.
[4] Z. Lin, Terminal Steiner tree with bounded edge length, *Canadian Conference on Computational Geometry*, Carleton University, Ottawa, Canada, pp.121-123, 2007.
[5] X. Cheng, D. Z. Du, L. Wang and B. Xu, Relay sensor placement in wireless sensor networks, *Wireless Networks*, vol.14, no.3, pp.347-355, 2008.
[6] H. Liu, P. Wan and X. Jia, On optimal placement of relay nodes for reliable connectivity in wireless sensor networks, *Journal of Combinatorial Optimization*, vol.11, no.2, pp.249-260, 2006.
[7] D. S. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *Journal of the ACM*, vol.32, no.32, pp.130-136, 1985.
[8] E. L. Lloyd and G. Xue, Relay node placement in wireless sensor networks, *IEEE Radio and Wireless Symposium*, vol.56, pp.134-138, 2011.
[9] B. Hao, J. Tang and G. Xue, Fault-tolerant relay node placement in wireless sensor networks: Formulation and approximation, *The Workshop on High Performance Switching & Routing*, vol.56, pp.246-250, 2004.
[10] J. Tang, B. Hao and A. Sen, Relay node placement in large scale wireless sensor networks, *Computer Communications*, vol.29, no.4, pp.490-501, 2006.
[11] S. R. Shishira, A. Kandasamy and K. Chandrasekaran, Survey on meta heuristic optimization techniques in cloud computing, *International Conference on Advances in Computing, Communications and Informatics*, pp.1434-1440, 2016.
[12] F. C. Harris, A stochastic optimization algorithm for Steiner minimal trees, *Congr. Numer.*, vol.105, 1995.
[13] S. Kirkpatrick and M. P. Vecchi, Optimization by simulated annealing, *Science*, vol.220, no.4598, pp.671-680, 1983.
[14] X. Ma and Q. Liu, An artificial fish swarm algorithm for Steiner tree problem, *IEEE International Conference on Fuzzy Systems*, pp.59-63, 2009.
[15] X. Ma and Q. Liu, A particle swarm optimization for Steiner tree problem, *The 6th International Conference on Natural Computation (ICNC)*, vol.5, pp.2561-2565, 2010.
[16] Y. Xu, J. Zhu, Y. Ma, C. Liu and J. Tao, A novel joint channel assignment scheme for cognitive radio wireless sensor networks, *ICIC Express Letters, Part B: Applications*, vol.8, no.8, pp.1173-1180, 2017.
[17] V. S. Pappala and I. Erlich, A variable-dimension optimization approach to unit commitment problem, *IEEE Trans. Power Systems*, vol.25, no.3, pp.1696-1704, 2010.
[18] Y. Yan and L. A. Osadciw, Density estimation using a new dimension adaptive particle swarm optimization algorithm, *Swarm Intelligence*, vol.3, no.4, pp.275-301, 2009.
[19] D. Chen, D. Z. Du, X. D. Hu, G. H. Lin, L. Wang and G. Xue, Approximations for Steiner trees with minimum number of Steiner points, *Journal of Global Optimization*, vol.18, no.1, pp.17-33, 2000.
[20] F. J. M. García, J. A. M. Pérez and F. J. M. García, Jumping frogs optimization: A new swarm method for discrete optimization, *Documentos De Trabajo Del Deioc*, 2008.