

Deterministic Deployment of Wireless Sensor Networks*

M. Brazil[†]

C. J. Ras[†]

D. A. Thomas[‡]

Abstract—We propose a new heuristic for deterministic deployment of wireless sensor networks when 1-connectivity and minimum cost are the two competing objectives. Given a set of data sources and a base station, our aim is to introduce the minimum number of relays to the network so that every sensor is connected to the base station via some multi-hop path. We assume that the data sources and base station lie in a plane, and that every sensor and relay has the same fixed communication radius. Our heuristic is based on the GEOSTEINER algorithms for the Steiner minimal tree problem, and proves to be much more accurate than the current best heuristics for the 1-connected deployment problem, especially in the case of sparse data source distributions.

Keywords: wireless sensor networks, connectivity, deterministic deployment, Steiner trees

1 Introduction

A wireless sensor network (WSN) consists of small sensing devices that can be readily deployed in diverse environments to form a distributed wireless network for collecting information in a robust and autonomous manner. Although early research was mainly motivated by potential military uses, there are now many other important applications such as fault detection, environmental habitat monitoring, irrigation and terrain monitoring (see, eg, [1], [9]). As an example of the latter, many lives have recently been lost in Australia due to bush fires. An early warning system is critical in preventing small fires from becoming disastrous infernos. Deploying smart sensors in strategically selected areas can lead to early detection and an increased likelihood of success in fire extinguishing efforts. There are also many medical applications including monitors and implantable devices as well as smart sensors for pollution control and climate control in large buildings.

A sensor network can be deployed in two ways: with deterministic placement, where a particular quality of service can be guaranteed; or with random placement, where

sensors are scattered possibly from an aircraft. Although many consider random placement to be the ultimate long term goal, it is currently infeasible in most situations as the individual sensors are generally too expensive for this level of redundancy (in many cases costing thousands of dollars each), and, under current technologies, often need to be carefully set up by hand. Note also that the deterministic case, where we can control placement of the nodes, provides a lower bound on the number of nodes needed to cover the area and hence is useful for the random model where the density of the sensors is a significant factor in performance. Randomly deployed WSNs may also be augmented (for any of a number of reasons) by deterministically deploying additional sensors or relays. For these reasons this paper will focus on deterministic deployment only.

The topic of this paper is the most fundamental objective of deployment of wireless sensor networks, namely 1-connectivity. Although it is possible to interconnect a WSN in many ways, we define 1-connectivity as the existence of at least one multi-hop path between every sensor in the network and the base station. This is the most basic requirement for the functioning of the network and can primarily be achieved in two ways: through power level adjustment (we will not consider this option here, but see [11] as a starting point for further study), or by deploying extra sensors or relays (see for instance [4]). The competing objective when deploying relays for connectivity is cost, which is predominantly determined by the number of added relays. Consequently our aim will be to deploy the minimum number of relays needed in order to ensure connectivity. We make the assumption that every sensor and relay in the network has the same communication radius, and by scaling we may assume that this radius is 1 unit.

These afore-mentioned conditions and assumptions lead to the following model for the 1-connected deterministic deployment problem. Represent all data sources (fixed sensors) and the base station by nodes embedded in the Euclidean plane. The objective is to embed the minimum number of additional nodes (relays) to ensure that there exists a tree interconnecting all nodes, where the length of every edge in the tree is at most 1 unit. This abstract version of the 1-connected deployment problem is referred to as the *minimum Steiner point tree* (MSPT) problem

*This research was supported by an ARC Discovery Grant.

[†]Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia; brazil@unimelb.edu.au.

[‡]Department of Mechanical Engineering, University of Melbourne; doreen.thomas@unimelb.edu.au.

in the literature, and an optimal solution is called a *minimum Steiner point tree*.

The MSPT problem was first described by Sarrafzadeh and Wong in [12], where they showed that it is NP-complete. Consequently a fair amount of research has been directed towards finding good heuristics. In [8] the minimum spanning tree (MST) heuristic was introduced (note that there they refer to the MSPT problem as the *Steiner tree problem with minimum number of Steiner points and bounded edge length*, or STP-MSPBEL). This heuristic simply subdivides all edges of an MST that are longer than one unit, resulting in an approximate MSPT solution within polynomial time. Mandoiu and Zelikovsky [10] prove that, in any metric space, the performance ratio of the MST heuristic is always one less than the maximum possible degree of a minimum-degree MST spanning points from the space. This gives an approximation ratio of four in the Euclidean plane and three in the rectilinear plane. Chen et al. [3] provide an improved approximation scheme, partly based on the MST heuristic, which has a performance ratio of three in the Euclidean plane.

The MSPT problem may be seen as a variant of the classical Steiner tree problem, which asks for a shortest tree interconnecting a set of nodes where any number of additional nodes may be introduced. An optimal solution to this problem is called a Steiner minimal tree (SMT). As the minimum distance between the given nodes tends to infinity, an SMT with subdivided edges becomes an optimal solution to the MSPT problem. This leads us to the question: would the SMT approximation for the MSPT problem be a practical and accurate heuristic? Certainly we do not have effective algorithms for calculating SMT's for very large sets of nodes, in fact the problem is NP-hard. However, Warme, Winter, and Zachariassen ([13],[14]) have developed practical, fast and optimal SMT algorithms for up to thousands of points, namely the GEOSTEINER algorithms.

In this paper we define and analyze the *SMT heuristic* for MSPTs. We provide a small linear upper bound in terms of the number of original data sources (sensors) for the performance difference of the SMT heuristic, and show that this bound is best possible in the Euclidean plane.

2 Preliminaries

Consider a set $N \subseteq \mathbb{R}^2$. The *Steiner tree problem* asks for a shortest tree interconnecting N , where extra nodes $W \subset \mathbb{R}^2$ are introduced if they reduce the total length. Introducing degree one or degree two nodes will not reduce total length, henceforth for the Steiner tree problem we assume all added nodes are of degree at least three. The nodes in N are called *terminal points* and the nodes in W are called *Steiner points*.

In our discussions we distinguish between the concept of a *free node* and an *embedded node*. In other words any tree may be considered as a topological graph structure only, or as an embedded network. Embedded nodes are denoted by bold letters (as is common when representing vectors).

Two standard techniques for shortening an embedded tree are *splitting* and Steiner point *displacements*. To *split* a node v one disconnects two or more of the edges at v and connects them instead to a new Steiner point, connected to v by an extra edge. To *displace* a Steiner point one simply embeds it at any new point in the plane without changing the topology of the tree. If no shortening of a tree is possible when splitting or Steiner point displacements are allowed, then the tree is called a *Steiner tree*. Note that an SMT is always a Steiner tree. A *full Steiner tree* is a Steiner tree where every terminal is of degree one and every Steiner point is of degree three. A full Steiner tree has exactly $|N| - 2$ Steiner points and $2|N| - 3$ edges. A *cherry* of a full Steiner tree is the subtree induced by two terminals and their mutually adjacent Steiner point. Every full Steiner tree has at least two cherries. We refer the reader to [6] and [7] for more background on Steiner trees.

Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, we denote the edge e between them by $e = \mathbf{xy}$, and we use the standard notation $|e|$ to denote $\|\mathbf{x} - \mathbf{y}\|$. Any Steiner tree can be viewed as a candidate MSPT if we simply subdivide, or *bead*, edges that are longer than one unit. Formally, *beading* is the process whereby for every edge e , $\lceil |e| \rceil - 1$ equally spaced degree-two nodes lying on e are included (along with the elements of W) in the set U of extra MSPT nodes. In general, any tree can be viewed as an MSPT candidate if we partition its nodes into a set N of terminals and a set W of Steiner points of degree at least three, and then bead any edges that are too long. Consequently, when constructing an MSPT on a given set N , we are mainly concerned with finding the elements of W , i.e. the elements of U that have degree at least three; clearly degree one nodes will not occur in U and degree two nodes in U only arise from beading. Henceforth, degree two nodes in U will not be considered as part of the topology of the MSPT. All nodes in U will be referred to as *beads* and, specifically, the nodes in W will be called *Steiner beads*. The procedure of constructing an SMT in order to approximate an MSPT will be referred to as the *SMT heuristic*.

Let T be any tree with node-set partitioned into terminals N and Steiner beads W . Let $n = |N|$. Then T^* is the tree that results by splitting nodes of T until every terminal is of degree one and every Steiner bead is of degree three (i.e. T^* is a full Steiner tree). New nodes are not displaced from their original positions, in other words some zero edge-lengths may be introduced and the total length of T does not change. See Figure 1 as an example;

here \mathbf{t} is a degree-four terminal, \mathbf{s} is a Steiner point, and after splitting \mathbf{t} we have three zero-length edges (depicted by broken lines).

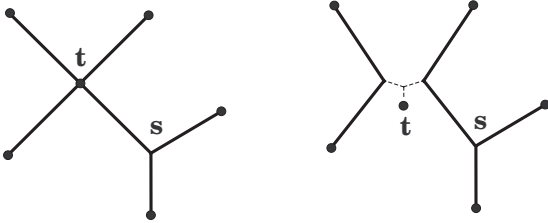


Figure 1: Conversion to a full Steiner tree

Let the edge-set of T^* be $E(T^*) = \{e_1, \dots, e_m\}$, where $m = 2n - 3$. Then the *bead count* of T is $\text{beads}(T) = |U| = n - 2 + \sum_{i=1}^m (\lceil |e_i| \rceil - 1) = 1 - n + \sum_{i=1}^m \lceil |e_i| \rceil$. In other words, by considering T^* rather than T we get a formula for $\text{beads}(T)$ that does not depend on the number of Steiner beads of T ; this formula works because every time a node is split (creating a new Steiner bead) we introduce a zero-length edge which in effect cancels the count of this Steiner bead. We can now reformulate the MSPT problem as follows: let N be a subset of \mathbb{R}^2 . Find a $W \subset \mathbb{R}^2$ and a tree T interconnecting $N \cup W$ such that every node in W is of degree at least three and $\text{beads}(T)$ is a minimum over all trees interconnecting N .

3 The Upper Bound

In this section we provide an upper bound for the performance difference of the SMT heuristic. Let N be a set of n terminals in the Euclidean plane. We use T_{opt} to denote an MSPT on N and T_S to denote an SMT on N . We need the following lemma before we prove our main result:

Lemma 1 *If i, k are real numbers then $\lceil i+k \rceil - \lceil i \rceil = \lceil k \rceil$ or $\lceil k \rceil - 1$ (equivalently $\lfloor k \rfloor$ or $\lfloor k \rfloor + 1$), with $\lceil i+k \rceil - \lceil i \rceil = k$ if k is an integer.*

Proof. Suppose that $\lceil i \rceil = i + \varepsilon_i$ and $\lceil k \rceil = k + \varepsilon_k$ where $0 \leq \varepsilon_i, \varepsilon_k < 1$. Then $\lceil i+k \rceil = \lceil i \rceil + \lceil k \rceil - \lfloor \varepsilon_i + \varepsilon_k \rfloor$, from which the result follows.

Suppose that $E(T_{\text{opt}}^*) = \{e_1, \dots, e_m\}$ and $E(T_S^*) = \{a_1, \dots, a_m\}$. Then $\sum_{i=1}^m |a_i| \leq \sum_{i=1}^m |e_i|$ since T_S is a shortest total length tree connecting N . We can therefore partition the set $\{1, \dots, m\}$ as follows: let $\{1, \dots, m\} = I \cup D$ such that $|e_i| = |a_i| + p_i$ for $i \in I$ and $|e_i| = |a_i| - p_i$ for $i \in D$. Here each p_i is a non-negative real number and the cardinality of D , but not I , may be zero. We further partition I into I_Z and I'_Z (where I_Z may be empty) such that $i \in I_Z$ if and only if $|a_i|$ is an integer. We similarly

partition D into D_Z and D'_Z . Note that $\sum_{i \in I} p_i \geq \sum_{i \in D} p_i$ - an inequality that is central to the next proof.

Proposition 2 $\text{beads}(T_S) - \text{beads}(T_{\text{opt}}) \leq 2n - 4 - j$, where j is the number of integer-length edges in $E(T_S^*)$.

Proof.

$$\begin{aligned} \text{beads}(T_S) - \text{beads}(T_{\text{opt}}) &= \left[1 - n + \sum_{i=1}^m \lceil |a_i| \rceil \right] - \left[1 - n + \sum_{i=1}^m \lceil |e_i| \rceil \right] \\ &= \sum_{i=1}^m \lceil |a_i| \rceil - \sum_{i=1}^m \lceil |e_i| \rceil \\ &= \sum_{i \in D} \{ \lceil |a_i| \rceil - \lfloor |a_i| - p_i \rfloor \} - \sum_{i \in I} \{ \lceil |a_i| + p_i \rceil - \lceil |a_i| \rceil \}. \end{aligned}$$

Therefore, if some $p_i \notin \mathbb{Z}$ for $i \in I$ then:

$$\begin{aligned} \text{beads}(T_S) - \text{beads}(T_{\text{opt}}) &\leq \sum_{i \in D_Z} \lfloor p_i \rfloor + \sum_{i \in D'_Z} (\lfloor p_i \rfloor + 1) - \sum_{i \in I_Z} \lfloor p_i \rfloor \\ &\quad - \sum_{i \in I'_Z} (\lfloor p_i \rfloor - 1) \\ &= |D'_Z| + |I'_Z| + \sum_{i \in D} \lfloor p_i \rfloor - \sum_{i \in I} \lfloor p_i \rfloor \\ &\leq m - j + \sum_{i \in I} p_i - \sum_{i \in I} \lfloor p_i \rfloor \\ &< m - j \\ &= 2n - 3 - j. \end{aligned}$$

Similarly, if $p_i \in \mathbb{Z}$ for all $i \in I$ then:

$$\begin{aligned} \text{beads}(T_S) - \text{beads}(T_{\text{opt}}) &\leq \sum_{i \in D_Z} \lfloor p_i \rfloor + \sum_{i \in D'_Z} (\lfloor p_i \rfloor + 1) - \sum_{i \in I} p_i \\ &\leq m - |I| - j \\ &< m - j. \end{aligned}$$

Corollary 3 $\text{beads}(T_S) - \text{beads}(T_{\text{opt}}) \leq 2n - c - 3$ where c is the number of full components of T_S .

Proof. Note that every terminal \mathbf{x} of degree $\text{deg}(\mathbf{x})$ is split $\text{deg}(\mathbf{x}) - 1$ times to produce T_S^* , i.e. each terminal \mathbf{x} produces $\text{deg}(\mathbf{x}) - 1$ zero-length edges after all splits. Clearly also $c = \sum_{\mathbf{x} \in N} \{ \text{deg}(\mathbf{x}) - 1 \}$.

Corollary 4 *If T_S has at most one edge with non-integer length then $\text{beads}(T_S) = \text{beads}(T_{\text{opt}})$.*

Du et al. in [3] and [5] provide an approximation for the MSPT problem that gives a performance ratio with upper bound of three in the Euclidean plane. Their algorithm is based on the MST heuristic and therefore runs in polynomial time. If we rewrite our performance difference to get the bounded ratio $\frac{\text{beads}(T_S)}{\text{beads}(T_{\text{opt}})} \leq 1 + \frac{2n-4}{\text{beads}(T_{\text{opt}})}$ we see that the performance ratio of the SMT heuristic has a smaller upper bound than the heuristic of Du et al. when $\text{beads}(T_{\text{opt}}) > n - 2$. Since $\text{beads}(T_{\text{opt}})$ increases as the minimum distance between any pair of terminals increases, we arrive at the intuitive fact that the performance of the SMT heuristic improves as the terminal configuration becomes more sparse. During this limiting process the upper bound of the ratio $\frac{\text{beads}(T_M)}{\text{beads}(T_S)}$, where T_M is an MST, tends towards the well-known Steiner ratio. This gives a limiting upper bound of $\frac{\text{beads}(T_M)}{\text{beads}(T_S)} \leq \frac{\sqrt{3}}{2}$ in the Euclidean plane, which serves as a comparison between the performances of the SMT heuristic and the standard MST heuristic.

We mention once again that the SMT heuristic does not run in polynomial time. However, for n up to a few thousand nodes the GEOSTEINER algorithms will produce solutions in reasonable running time. This makes the SMT heuristic a tool worthy of consideration for applications where optimization is required during an initialization process (such as deployment) and the cost benefit of a more accurate algorithm justifies a possible time delay.

4 Sharpness of the Upper Bound

The aim of this section is to show that the performance difference from Proposition 2 is best possible. We begin with a few definitions and preliminary results. Due to minimality of total length, any two adjacent edges of a Euclidean Steiner tree meet at an angle of at least 120° . This implies that the degree of any terminal is no more than 3, and the degree of any Steiner point is exactly 3. Let T be a full Steiner tree on a set of embedded terminals. To *sprout* new terminals from a given terminal \mathbf{t} of T , with incident edge e one replaces \mathbf{t} by a Steiner point \mathbf{s} and embeds two new terminals $\mathbf{t}_1, \mathbf{t}_2$ adjacent to \mathbf{s} such that the two new edges \mathbf{st}_1 and \mathbf{st}_2 each form 120° angles with e and with each other - see Figure 2. We denote by $L(T)$ the total Euclidean edge length of T . If N is a set of embedded terminals then T_S will denote a Euclidean SMT on N and T_{opt} will denote a Euclidean MSPT on N . As usual we let $n = |N|$. The next proposition shows that we can use sprouting to create full SMTs with any given topology. It is a fundamental result and is almost certainly known, but does not appear to have been explicitly written up in the literature before now.

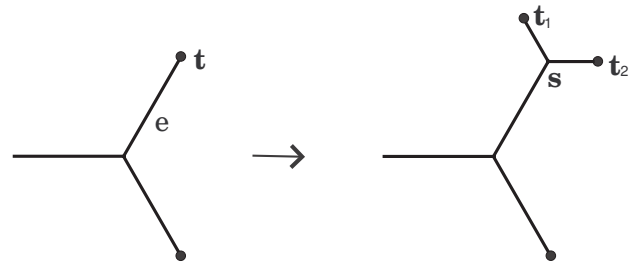


Figure 2: Sprouting new terminals

Proposition 5 *Given any full Steiner topology, there exists a set of embedded terminals N such that the SMT for N has the given topology and is unique. Furthermore, such trees can be explicitly constructed for any given topology.*

Proof. Let G_n be a full Steiner topology on n terminals. We will show how a suitable set of embedded terminals N_n can be constructed by induction on n , where the inductive step involves sprouting new terminals. Note that the construction is trivial if $n = 1, 2$ or 3 . The inductive claim is as follows.

Claim: For any full Steiner topology, G_i , on i terminals (with $i \geq 4$), there exists a set of embedded terminals N_i and a real number $f_i > 0$ such that

1. the SMT, T_i , for N_i has topology G_i , and
2. if T'_i is a Steiner tree for N_i such that the topology of T'_i is not G_i , then $L(T'_i) - L(T_i) \geq f_i$.

For the base case of the claim ($i = 4$), choose N_4 to be the four points with coordinates $(\pm 3/2, \pm \sqrt{3}/2)$. It is easily checked that the SMT T_4 for N_4 has Steiner points $(\pm 1/2, 0)$ and length 5 (see Figure 3). The shortest Steiner tree T'_4 with a different topology is full with Steiner points $(0, \pm(\sqrt{3}/2 - 1/\sqrt{3}))$ and length $L(T'_4) = 3\sqrt{3}$. So we can choose $f_4 = 3\sqrt{3} - 5 > 0$. Up to relabelling of the terminals, there is only one full topology for $i = 4$, so this completes the base case.

We now establish the inductive step for ($i = n$), where we assume that the claim holds for $i = n - 1$. Given a full Steiner topology G_n ($n > 4$), this topology contains at least one cherry. Replacing such a cherry by a single terminal t^* gives a full Steiner topology G_{n-1} on $n - 1$ terminals. By the inductive assumption there exists an embedded terminal set N_{n-1} with unique SMT T_{n-1} which has topology G_{n-1} and a corresponding constant $f_{n-1} > 0$. Let \mathbf{t} be the embedded terminal corresponding to t^* and create a new Steiner tree as follows.

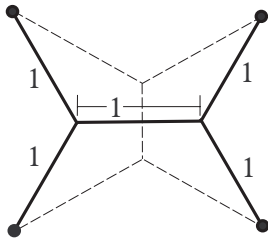


Figure 3: Base case

We sprout new terminals t_n and t_{n-1} from t , with t replaced by a Steiner point s , such that $|st_{n-1}| = |st_n| = f_{n-1}/4$. Let this new tree be T_n with embedded terminal set N_n . By construction, T_n has the correct topology G_n .

Let T'_n be any Steiner tree (but not necessarily an SMT) on N_n with topology *not* G_n . Suppose we collapse t_n and t_{n-1} into the point s , and consider the resulting topology G of this network. If $G = G_{n-1}$, then T'_n also has the same topology as T_n , which by convexity and the fact that T'_n is a Steiner tree implies that $T'_n = T_n$ (see Theorem 1.3 of [7]); this is a contradiction and hence $G \neq G_{n-1}$. It follows from this that, if we consider the network $T'_n \cup \{st_n\}$ (which interconnects N_{n-1}), we have $L(T'_n) + |st_n| \geq L(T_{n-1}) + f_{n-1}$. This implies that

$$\begin{aligned} L(T'_n) &\geq L(T_{n-1}) + (f_{n-1} - |st_n|) \\ &> L(T_{n-1}) + 3|st_n| = L(T_n) + |st_n|. \end{aligned}$$

Hence, we can choose $f_n = f_{n-1}/4 < L(T'_n) - L(T_n)$.

The claim (and lemma) now follow. Furthermore, the iterative algorithm for constructing a suitable set of embedded terminals for any required Steiner topology is constructive with $f_i = (3\sqrt{3} - 5)/4^{i-4}$ for each $i \geq 4$.

The next proposition shows that the upper bound from Proposition 2 is sharp.

Proposition 6 *Let G_n be a full Steiner topology on n terminals. There exists an embedded set of terminals N in the Euclidean plane such that $\text{beads}(T_S) = \text{beads}(T_{\text{opt}}) + 2n - 4$ and T_S has topology G_n .*

Proof. We construct an SMT T_S with topology G_n by repeatedly sprouting terminals, starting from a full Steiner tree on three terminals called the *base*. By the previous proposition any full Steiner topology can be produced in this way. By making the edges of the base large enough, it is clear that we can construct T_S such that every edge-length has the form $a_i \pm \varepsilon_i$, where a_i is an integer of order at least two and ε_i has any predefined value between zero and one. T_S is then converted into an MSPT by a sequence of displacements (which we describe

below) of the Steiner points, where displacements do not change the original topology G_n .

In T_S , let s_0 be a Steiner point adjacent to a terminal t and two other nodes v_1, v_2 where edge-lengths are preselected as follows: $|ts_0| = a_1 - \varepsilon$, $|s_0v_1| = |s_0v_2| = b_1 + \varepsilon_1$ for large integers a_1, b_1 and $0 < \varepsilon, \varepsilon_1 < 1$. In the first step (Figure 4) we displace s_0 along the line through t and s_0 and in the direction of the vector ts_0 . We displace until $|ts_0| = a_1 - \varepsilon'$ and $|s_0v_1| = |s_0v_2| = b_1 - \varepsilon'_1$ for some $0 < \varepsilon', \varepsilon'_1 < 1$. Clearly this is possible as long as we preselect ε_1 to be small enough compared to ε .

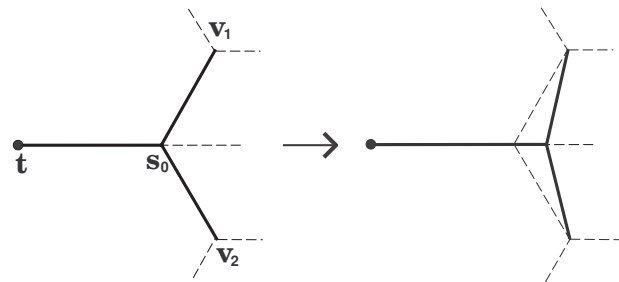


Figure 4: First step of the displacement sequence

We now displace all other Steiner points in a depth-first or breadth-first order rooted at t . Suppose that in the process we have reached the Steiner point s with parent s' and children u_1, u_2 . We displace s along the line through s and the point p and in the direction \overrightarrow{ps} , where p is the position s' had *before* its displacement; see Figure 5. If $|ss'| = a - \varepsilon_1$ then we preselect $|su_1| = |su_2| = b + \varepsilon_2$ for $0 < \varepsilon_2 < 1$. We select ε_2 small enough so that the displacement of s produces the lengths $|ss'| = a - \varepsilon'_1$ and $|su_1| = |su_2| = b - \varepsilon'_2$, for some $0 < \varepsilon'_1, \varepsilon'_2 < 1$. We continue this process until we have displaced all Steiner points. Call the resultant tree T . Note that the edges of T_S were preselected so that one edge has length $a_1 - \varepsilon$ and every other edge e_i has length $b_i + \varepsilon_i$. After all displacements the first edge has length $a_1 - \varepsilon'$ and every other e_i has length $b_i - \varepsilon'_i$. Clearly then $\text{beads}(T_S) = \text{beads}(T) + 2n - 4$ and T is an MSPT.

5 Conclusions and Future Work

In this paper we defined and analyzed the SMT heuristic for the deterministic deployment of sensor networks with a 1-connectivity objective. We find a provably sharp upper bound for the performance of the heuristic and we argue that the performance will improve as the data-source configuration becomes more sparse. In fact, there exists a lower bound on the minimum distance between pairs of data-sources which guarantees that the SMT heuristic has a better worst-case performance than the current best possible heuristics in the literature.

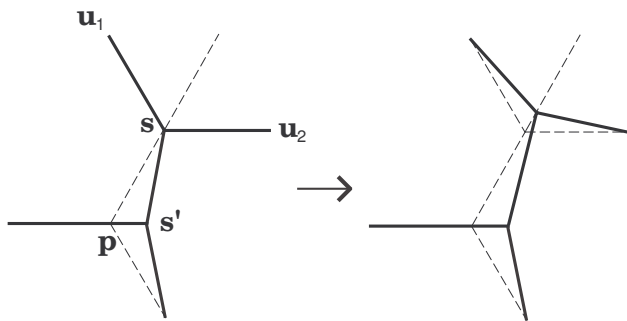


Figure 5: General step of the displacement sequence

Extensive simulations have been performed for the GEOSTEINER algorithm (see for instance [13]), so we will not repeat them here. These simulations show that one can calculate SMTs efficiently for up to about two thousand terminals. We therefore believe that the SMT heuristic will become a valuable tool for WSN deployment, especially in applications such as habitat and terrain monitoring where, indeed, data-sources distributions are often sparse and seldom contain more than a couple of thousand individual sources. Of course, any deterministically deployed WSN is unlikely to consist of an extremely large number of sensors, which means that the SMT heuristic is generally suitable for most of these scenarios.

Computational and theoretical results have led us to believe that it is possible to improve the performance of the SMT heuristic even more by small displacements of the Steiner points. One of the ways we wish to explore this possibility in the future is through extensive simulations.

Acknowledgement. We wish to thank Jamie Evans for providing many fruitful ideas during the development of this paper.

References

[1] T. Arampatzis, L. Lygeros, J. S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor networks," *Proc. 13th Mediterranean Conf. on Control and Automation* (2005), pp. 719–724.

[2] J.-H. Chang, R.-H. Jan, "An Efficient Relay Sensors Placing Algorithm for Connectivity in Wireless Sensor Networks," *Lecture Notes in Computer Science - Embedded and Ubiquitous Computing*, 4096 (2006), pp. 874–883.

[3] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, G. Xue, Approximations for Steiner trees with minimum number of Steiner points, *Theoretical Computer Science*, 262 (2001) pp. 83–99.

[4] X. Cheng, D.-Z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, *Wireless Networks*, 14 (2008), pp. 347–355.

[5] D.-Z. Du, X. Hu, Steiner tree problems in computer communication networks, *World Scientific Publishing Company*, 2008.

[6] E. N. Gilbert, H. O. Pollak, Steiner minimal trees, *SIAM J. Appl. Math.*, 16 (1968), pp. 1–29.

[7] F. K. Hwang, D. S. Richards, P. Winter, The Steiner Tree Problem, *Annals of Discrete Mathematics* 53, Elsevier Science Publishers B.V., Amsterdam, 1992.

[8] G.-H. Lin, G.L. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, *Inform. Process. Lett.*, 69 (1999), pp. 53–57.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)* (2002), pp. 88–97.

[10] I. I. Mandoiu, A. Z. Zelikovsky, A note on the MST heuristic for bounded edge-length Steiner trees with minimum number of Steiner points, *Inform. Process. Lett.*, 75 (2000), pp. 165–167.

[11] M. Naserian, K. Tepe, T. Mohammed, "On the Connectivity of Nodes in Wireless Ad Hoc and Sensor Networks," *Canadian Conference on Electrical and Computer Engineering* (2005), pp. 2073–2075.

[12] M. Sarrafzadeh, C.K. Wong, Bottleneck Steiner trees in the plane, *IEEE Trans. Comput.*, 41 (1992), pp. 370–374.

[13] D. M. Warme, P. Winter, M. Zachariasen. Exact Algorithms for Plane Steiner Tree Problems: A Computational Study. In D.-Z. Du, J. M. Smith, J. H. Rubinstein, editors, *Advances in Steiner Trees*, pp. 81–116, Kluwer Academic Publishers, Boston, 2000.

[14] P. Winter, M. Zachariasen, Euclidean Steiner Minimum Trees: An Improved Exact Algorithm, *Networks*, 30 (1997), pp. 149–166.