On Minimum- and Maximum-Weight Minimum Spanning Trees with Neighborhoods

Reza Dorrigiv[1], Robert Fraser[2], Meng He[1], Shahin Kamali[2], Akitoshi Kawamura[3], Alejandro López-Ortiz[2], and Diego Seco[4]

[1] Dalhousie University, Halifax, Canada
[2] University of Waterloo, Waterloo, Canada
[3] University of Tokyo, Tokyo, Japan
[4] University of A Coruña, A Coruña, Spain

Abstract. We study optimization problems for the Euclidean minimum spanning tree (MST) on imprecise data. To model imprecision, we accept a set of disjoint disks in the plane as input. From each member of the set, one point must be selected, and the MST is computed over the set of selected points. We consider both minimizing and maximizing the weight of the MST over the input. The minimum weight version of the problem is known as the minimum spanning tree with neighborhoods (MSTN) problem, and the maximum weight version (MAX-MSTN) has not been studied previously to our knowledge. We provide deterministic and parameterized approximation algorithms for the MAX-MSTN problem, and a parameterized algorithm for the MSTN problem. Additionally, we present hardness of approximation proofs for both settings.

1 Introduction

We consider geometric problems dealing with imprecise data. In this setting, each point of the input is provided as a *region of uncertainty*, i.e., a geometric object such as a line, disk, set of points, etc., and the exact position of the point may be anywhere in the object. Each object is understood to represent the set of possible positions for the corresponding point. In our work, we study the Euclidean minimum spanning tree (MST) problem. Given a tree T, we define its weight $w(T)$ to be the sum of the weights of the edges in T. For a set of fixed points P in Euclidean space, the weight of an edge is the distance between the endpoints, and we write $mst(P)$ for the weight of the MST on P. Thus, $mst(P) = \min w(T)$, where the minimum is taken over all spanning trees T on P.

Given a set of disjoint disks as input, we wish to determine the minimum and maximum weight MSTs possible when a point is fixed in each disk. The minimum weight MST version of the problem has been studied previously, and is known as the minimum spanning tree with neighborhoods problem (MSTN). This paper introduces the maximum weight MST version of the problem, which we call the MAX-MSTN problem. Assume we are given a set $D = \{D_1, \ldots, D_n\}$ of disjoint disks in the plane, i.e., $D_i \cap D_j = \emptyset$ if $i \neq j$. The MSTN problem on D asks for the selection of a point $p_i \in D_i$ for each $D_i \in D$ such that the weight of the MST of the selected points is minimized. Similarly, MAX-MSTN asks for a selection of p_i such that the weight of the MST of the selected points is maximized.

1.1 Related Work

The first known MST algorithm was published over 80 years ago [11], and a number of successful variants have followed (see [8] for the history of the problem). A review of models of uncertainty and data imprecision for computational geometry problems is provided in [10]. Here, we discuss a few results that are directly related to the MST problem and our model of imprecision.

The MSTN problem on unit disks has been shown to admit a PTAS [13]. A hardness proof for a generalization of MSTN where the neighborhoods are either disks or rectangles appeared in [13], but the proof was faulty. One of the authors later conjectured that a reduction from planar 3-SAT might be used to show the hardness of the MSTN problem [12, p.106]. In Section 3.2, we prove this conjecture.

Löffler and van Kreveld [10] demonstrated that it is algebraically difficult to compute the MST when the regions of uncertainty are continuous regions of the plane, even for very simple inputs such as disks or squares, as the solution may involve the roots of high degree polynomials. It is of independent interest to see if the problem is combinatorially difficult. In the same paper, authors proved that the MSTN problem is (combinatorially) NP-hard if the regions of uncertainty are not pairwise disjoint, through a reduction from the minimum Steiner tree problem. In this paper we prove the hardness of the special case in which the regions are pairwise disjoint.

Erlebach et al. [6] used a model of uncertainty where information regarding the weight of an edge between a pair of points or the position of a point may be obtained by pinging the edge or vertex, and they sought to minimize the number of pings required while obtaining the optimal solution. The distinction is that in their work, they were interested in reducing the amount of communication that is required to locate points within a region of uncertainty, while in our model, the objective is to optimize the MST given regions of uncertainty.

Researchers have considered other related problems that deal with imprecise data. The travelling salesman with neighborhoods (TSPN) problem has been studied extensively. The problem was introduced by Arkin and Hassin [1], in a paper that has been applied, improved, built-upon or otherwise referenced over 150 times. There exists a PTAS for TSPN when the neighborhoods are disjoint unit disks [5]. The most general version of the problem, where regions may overlap and may have varying sizes, is known to be APX-hard [2]. The problem of maximizing the smallest pairwise distance in a set of n points with neighborhoods has also been studied and proved to be NP-hard [7].

1.2 Our Results

We present a variety of results related to the MSTN and MAX-MSTN problems. For both problems we assume the regions of uncertainty (disks) are disjoint.

– MAX-MSTN: deterministic 1/2-approximation;
– MAX-MSTN: parameterized $1 - \frac{2}{k+4}$-approximation (where k represents the separability of the instance, which is to be defined later);
– MAX-MSTN: proof of hardness of approximation;
– MSTN: parameterized $1 + 2/k$-approximation (k is the separability of the instance);
– MSTN: proof of hardness of approximation.

The deterministic approximation algorithm for MAX-MSTN (Section 2.1) is based on choosing the center points of the disks; the interesting aspect in this section lies in the analysis. The parameterized algorithms (Sections 2.2 and 3.1) for both settings were inspired by the observation that the approximation factor improves rapidly as the distance between disks increases. To address this, we introduce a measure of how much separation exists between the disks, which we call *separability*, and we analyze the approximation factor of the MST on disk centers with respect to separability.

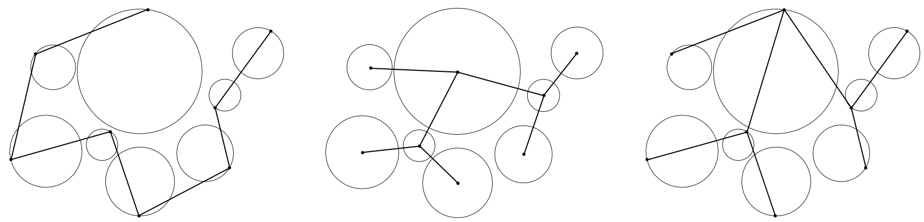
For both hardness of approximation results, we establish that there is no FPTAS for the problems unless P=NP. Although the hardness proofs both consist of reductions from planar 3-SAT, the gadgets used are quite distinct and either reduction is interesting even given the existence of the other. In both cases, we construct an instance of our problem from the planar 3-SAT instance, and show that it is possible to compute the weight of the optimal solution to our problem assuming that the 3-SAT instance is satisfiable. If the instance is not satisfiable, we prove that the weight is changed by at least a constant amount (reduced by at least 0.33 units for MAX-MSTN, and increased by at least 0.84 units for MSTN).

2 MAX-MSTN

In this section we study a couple of approximation algorithms for the MAX-MSTN problem, and then we present the proof of hardness of approximation. We begin with a deterministic algorithm below, followed by a parameterized algorithm in Section 2.2.

2.1 Deterministic 1/2-Approximation Algorithm

To approximate the solution to MAX-MSTN, we first consider the algorithm that builds an MST on the centers of the disks. We show this algorithm approximates the optimal solution within a factor of 1/2, i.e., the weight of the MST built on the centers is not smaller than half of that of the optimal tree.



(a) (b) (c)

Fig. 1. To compare $w(T_c)$ with $w(T_{\text{opt}})$, we use an intermediate tree T'_c. (a) The optimal result for MAX-MSTN (T_{opt}). (b) The MST T_c on centers. (c) The spanning tree T'_c with the same topology as T_c, using the points of T_{opt}.

Theorem 1. *Consider the* MAX-MSTN *problem for a set D of disjoint disks. Let T_c denote the MST on the centers of the disks, and let T_{opt} be the maximum MST (i.e., the optimal solution to the problem). Then $w(T_c) \geq 1/2 \cdot w(T_{opt})$.*

Proof. Let T'_c be the spanning tree (not necessarily an MST) with the same topology (i.e., combinatorial structure of the tree) as T_c but on the points of T_{opt} (see Figure 1). Since T'_c and T_{opt} span the same set of points, and T_{opt} is an MST, we have $w(T_{\text{opt}}) \leq w(T'_c)$. On the other hand, since T'_c and T_c have the same topology, we have $w(T'_c) \leq 2w(T_c)$; this is because when we move the points from the center to somewhere else in the disks, the weight of each edge increases by at most the sum of the radii of the two involved disks and, since the disks are disjoint, the increase is at most equal to the original weight. To summarize, we have $w(T_{\text{opt}}) \leq w(T'_c)$ and $w(T'_c) \leq 2w(T_c)$, which completes the proof. □

2.2 Parameterized $1 - \frac{2}{k+4}$-Approximation Algorithm

Observe that in order to get the approximation algorithm for MAX-MSTN in Section 2.1, we require disks to be disjoint. Intuitively, if we know that disks are further apart, we can get better approximation ratios. We formalize this intuition by providing a parameterized analysis, i.e., we express the performance of the algorithm in terms of a *separability parameter*[5]. Let $r_{\max}$ be the maximum radius of our disks. We say that a given input for our problem satisfies *k-separability* if the minimum distance between any two disks is at least $k \cdot r_{\max}$. The separability of an input instance I is defined as the maximum k such that I satisfies k-separability. With this definition, we have the following result:

Theorem 2. *For* MAX-MSTN *when the regions of uncertainty are disjoint disks with separability parameter $k > 0$, the algorithm that builds an MST on the centers of the disks achieves a constant approximation ratio of $\frac{k+2}{k+4} = 1 - \frac{2}{k+4}$.*

[5] Separability is similar in spirit to the notion of a well-separated pair; see [3].

Proof. Let T_c be the MST on the centers of the disks. We can extend the analysis in the proof of Theorem 1 to show that the approximation factor is $\frac{k+2}{k+4} = 1 - \frac{2}{k+4}$ for any input that satisfies k-separability. Define T_{opt} and T'_c as before. Consider an arbitrary edge e in T'_c and let D_i and D_j be the two disks connected by e. Let r_i and r_j be the radii of D_i and D_j, respectively, and let d be the distance between D_i and D_j. In T_c the disks D_i and D_j are connected by an edge e' whose weight is $d + r_i + r_j$. The weight of e, on the other hand, can be at most $d + 2r_i + 2r_j$. Therefore, the ratio between the weight of an edge in T_c and its corresponding edge in T'_c is at least

$$\frac{d + r_i + r_j}{d + 2r_i + 2r_j} \geq \frac{kr_{\max} + r_i + r_j}{kr_{\max} + 2r_i + 2r_j} \geq \frac{kr_{\max} + r_{\max} + r_{\max}}{kr_{\max} + 2r_{\max} + 2r_{\max}} = \frac{k+2}{k+4}.$$

Since this holds for any edge of T'_c, we get $w(T_c) \geq \frac{k+2}{k+4} w(T'_c) \geq \frac{k+2}{k+4} w(T_{\text{opt}})$, and we get an approximation factor of $\frac{k+2}{k+4}$. □

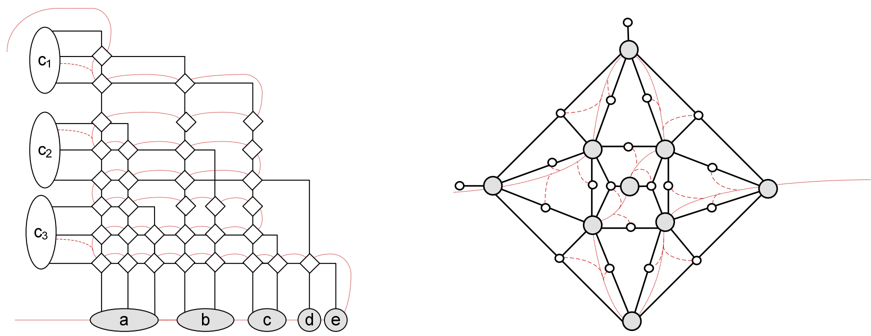
The approximation ratio gets arbitrarily close to 1 as k increases. This confirms our intuition that if the disks are further apart (more separate), we get a better approximation factor.

2.3 Hardness of Approximation

We present a hardness proof for the MAX-MSTN problem by a reduction from the planar 3-SAT problem [9]. Planar 3-SAT is a variant of 3-SAT in which the graph $G = (V, E)$ associated with the formula is planar.

Theorem 3. MAX-MSTN *does not admit an FPTAS unless* P=NP.

We show a reduction from any instance of the planar 3-SAT problem to the MAX-MSTN problem. In planar 3-SAT, we have a planar bipartite graph $G = (V, E)$, where $V = V_v \cup V_c$, so that there is a vertex in V_v for each variable and a vertex in V_c for each clause; there is an edge (v_i, v_j) in E between a variable vertex $v_i \in V_v$ and a clause vertex $v_j \in V_c$ if and only if the clause contains a literal of that variable in the 3-SAT instance. In [9] it was shown that the planar 3-SAT problem is NP-hard via a reduction from the standard 3-SAT problem. Further, it was observed that the resulting instance of planar 3-SAT permits the construction of a path $P = (V_v, E_P)$ using a set of edges E_P such that $E \cap E_P = \emptyset$, where P is connected and passes through all vertices in V_v without crossing any edge in E. We call this path P the *spinal path*. We further observe that additional edges can be added to P to get a *spinal tree* T which also covers clause vertices V_c. In this sense T will be a tree that covers all vertices without crossing an edge of G such that all vertices corresponding to clauses are leaves. These observations are illustrated in Figure 2. To prove the hardness of MAX-MSTN, we make use of the spinal tree. Due to lack of space, the complete reduction is omitted from this presentation (the details may be found in the complementary technical report [4]).



(a) The reduction from 3-SAT to planar 3-SAT.

(b) The planar gadget located in the intersection points. Variable and clause vertices are represented by large and small circles, respectively.

Fig. 2. The reduction from 3-SAT to planar 3-SAT as presented in [9]. The variable and clause vertices of 3-SAT are located respectively in x and y axis, and the edges are drawn as orthogonal paths (a). A planar gadget is placed on each intersection point. Each gadget includes some new variable and clause vertices (b). In [9], it is observed that there is a path (we call it the spinal path) that covers all variable vertices of planar instance without crossing any edge (solid lines). We observe that additional edges can be added to the spinal path to obtain a tree (spinal tree) which spans clause variables as leaves (dashed lines).

3 MSTN

In this section we present a parameterized algorithm for the MSTN problem, followed by the proof of hardness of approximation.

3.1 Parameterized $1+2/k$-Approximation Algorithm

Recall that to have *k-separability* means that the minimum distance between any two disks is at least $kr_{\max}$, and the separability of an input instance I is defined as the maximum k such that I satisfies k-separability.

Theorem 4. *For* MSTN *when the regions of uncertainty are disjoint disks with separability parameter $k > 0$, the algorithm that builds an MST on the centers of the disks achieves a constant approximation ratio of $\frac{k+2}{k} = 1 + 2/k$.*

Proof. Assume that we have a set D of n disks that satisfies k-separability. Let T_c be the MST on the centers and T_{opt} be an optimal MST, i.e., an MST that contains one point from each disk and its weight is the minimum possible. Define *Temp* as the spanning tree (not necessarily an MST) with the same topology as T_{opt} but on the points of T_c, i.e., on the centers. Since T_c is an MST on centers, we have $w(T_c) \leq w(\mathit{Temp})$. Consider an arbitrary edge e in *Temp* and let D_i

and D_j be the two disks that are connected by e. Let r_i and r_j be the radii of D_i and D_j, respectively, and let d be the distance between D_i and D_j. In T_{opt} the disks D_i and D_j are connected by an edge e' whose weight is at least d. The weight of e on the other hand is $d + r_i + r_j$. Therefore the ratio between the weight of an edge in T_{opt} and its corresponding edge in *Temp* is at least

$$\frac{d}{d + r_i + r_j} \geq \frac{kr_{\max}}{kr_{\max} + r_i + r_j} \geq \frac{kr_{\max}}{kr_{\max} + r_{\max} + r_{\max}} = \frac{k}{k+2}.$$

Since this holds for any edge of *Temp*, we get $w(T_c) \leq w(\mathit{Temp}) \leq \frac{k+2}{k} w(T_{\text{opt}})$. Therefore we get an approximation factor of $\frac{k+2}{k} = 1 + 2/k$ for the algorithm. □

As with the parameterized algorithm for MAX-MSTN, as the disks become further apart (as k grows), the approximation factor approaches 1.

3.2 Hardness of Approximation

To prove the hardness of the MSTN problem, we present a reduction from the planar 3-SAT problem. Recall that planar 3-SAT is a variant of 3-SAT in which the graph $G = (V, E)$ associated with the formula is planar.

Theorem 5. MSTN *does not admit an FPTAS unless* P=NP.

In the hardness proof of MAX-MSTN, we used a spinal tree in the reduction. In this section, we use the *spinal path* as a path $P = (V_v, E_P)$ with a set of edges E_P such that $E \cap E_P = \emptyset$, where P passes through all variable vertices in G without crossing any edge in E. As mentioned earlier, the restricted version of planar 3-SAT remains NP-hard [9]. To reduce planar 3-SAT to MSTN, we begin by finding a planar embedding of the graph associated with the SAT formula. We force the inclusion of the spinal path as a part of the MST using wires. We define a wire as a set of disks of radius 0 placed in close succession, so that we may interpret a wire as a fixed line in the MSTN solution. We replace each variable vertex of V by a *variable gadget* in our construction. These gadgets are composed of a set of disks and some wires, and are defined in such a way that we may choose the points so that the size of the MST is equal to a certain value, if and only if the SAT formula is satisfiable.

Variable Gadgets A variable gadget is formed by a *k-flower*, where $k = 4c + 6$ and c is the number of clauses in the planar 3-SAT instance that include the variable (each clause requires 4 disks, and each of the edges of the spinal path requires 3 disks). As illustrated in Figure 3, a k-flower is composed of k disks of unit radius, centered on the vertices of a regular k-gon. Also, each disk is tangent to its two neighboring disks, and each pair of consecutive disks D_i, D_{i+1} intersects at a single point $q_{i,i+1} = D_i \cap D_{i+1}$, which we call a *tangent point*[6].

[6] Using this construction, pairs of disks of the k-flower trivially intersect at a single point, which simplifies our analysis. To achieve strict disjointedness, the disks of the

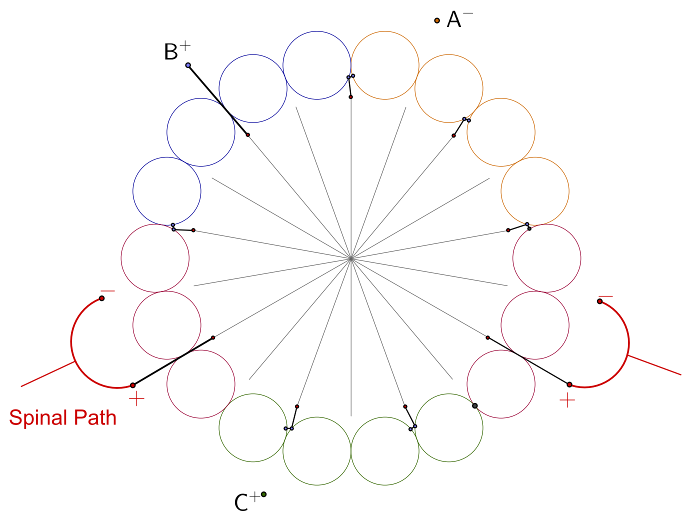


Fig. 3. A variable gadget with eighteen disks (containing an 18-flower and an 18-star) for a variable x. Here B^+ and C^+ are the endpoints of the wires that connect to clauses that include x in the positive form, while A^- represents a clause that includes x in negative form. The picture illustrates the case in which the algorithm takes the positive choice for x, and clause B is satisfied with x. Clause C is satisfied via some other variable, as is clause A, assuming that it is satisfied. Note that every other path on the k-star connects to a pair of disks on the k-flower.

Moreover, there is a k-*star* in the middle of the gadget composed of k fixed wires, where the i^{th} wire connects a point unit distance from the tangent point $q_{i,i+1}$ to the center point of the k-star. The spinal path is placed so that it approaches the variable gadget twice, and each of these approaches requires three disks. We split the wires of the spinal path once near the variable gadget as shown in Figure 3, and wires terminate at a distance ≈ 1.755 from the nearest tangent point, for reasons discussed in the Clause Gadgets section.

Lemma 1. *Suppose we are given two unit disks D_1 and D_2 that intersect exclusively at a single point $q = D_1 \cap D_2$, and a line ℓ such that $q \in \ell$ and ℓ is tangent to both D_1 and D_2 (i.e., ℓ is the perpendicular bisector of the center points of D_1 and D_2). Now, given a point $p \in \ell$ where p is unit distance from q, the shortest path consisting of points $p, q_1 \in D_1$, and $q_2 \in D_2$ has weight $d \approx 0.755$.*

k-flower may be contracted to have radius $1-\gamma$ so that the tangent point is now distance γ from the nearest point in the adjacent disks. Any path which uses the tangent point in our analysis will have less than 2γ units of additional weight on these shrunken disks, and there are fewer than $n(4m+6)$ disks, where n and m are the number of variables and clauses respectively. Choosing an appropriate value of γ so that $2\gamma n(4m+6) \ll 0.845$ achieves the same result as our simplified analysis.

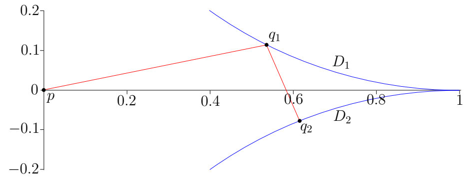


Fig. 4. The shortest possible path is shown from a point at the origin to some point in each of two unit disks; one of the disks is centered at (1,1), the other is at (1,-1).

Proof. If $q_1 = q_2 = q$, then the path has unit length, so a path of length d is shorter. A path with edges $e_1 = (q_1, p)$ and $e_2 = (p, q_2)$ has length at least 0.828, since the nearest point on D_1 or D_2 to p is $\sqrt{2} - 1 > 0.414$ units distant.

Therefore, we may assume without loss of generality that the path consists of the edges $e_1 = (p, q_1)$ and $e_2 = (q_1, q_2)$ and the path has length $d = w(e_1)+w(e_2)$, where $w(e)$ is the length of the edge e. Therefore, we must choose q_1 and q_2 so that d is minimized. Note that candidate positions for each of q_1 and q_2 may be restricted to the boundaries of their respective disks.

For the purposes of simplifying the proof, assume that p is at the origin of the Cartesian plane, and D_1 and D_2 are centered at $(1, 1)$ and $(1, -1)$, respectively. Then a point q_1 on the boundary of D_1 may be expressed as $(sin(\alpha)+1, cos(\alpha)+1)$, for some $\alpha \in [0 \ldots 2\pi]$, and analogously $q_2 = (sin(\beta)+1, cos(\beta)-1)$, for some $\beta \in [0 \ldots 2\pi]$. Therefore, we simply have to find the minimum of the function

$$f(\alpha, \beta) = \sqrt{(\sin\alpha + 1)^2 + (\cos\alpha + 1)^2} + \sqrt{(\sin\beta - \sin\alpha)^2 + (\cos\beta - \cos\alpha - 2)^2},$$

over the variables $\alpha \in [0 \ldots 2\pi], \beta \in [0 \ldots 2\pi]$. Using Maple, we see that this minimum has value $d \approx 0.755$, at $\alpha \approx 3.62, \beta \approx 5.89$. The optimal path in this setting is shown in Figure 4. Since this path is shorter than all other possible path configurations, we conclude that this is the shortest possible path including p and points $q_1 \in D_1$ and $q_2 \in D_2$. ☐

For the remainder of the discussion, we refer to the weight of this shortest path as the constant d. Before going to the details of the reduction, we consider optimal MSTN solutions when the problem instance is a variable gadget, as described above (without the wires approaching from clauses). We claim that such an instance has two possible MSTN solutions, and in each of these solutions consecutive pairs of disks are connected to a single wire of the k-star with a path of length d described in Lemma 1. We associate these two possible MSTN solutions with the two assignments for the variable. To prove the claim, we show that in an optimal MSTN solution for a k-star, there is no path containing points from more than two disks.

Lemma 2. *In an optimal MSTN solution for a k-star, a path containing a single wire of the k-star includes at most two disks from the k-flower, when $k \geq 8$.*

Proof. Recall that by Lemma 1, connecting a pair of disks to a k-wire may be done with weight d, while a wire may be connected to a single disk with weight $\sqrt{2}-1$. Therefore, three consecutive disks in a k-flower may be connected to two wires of the k-star using edges with weight $d+\sqrt{2}-1 \approx 1.169$, while four such disks may be connected with weight $2d \approx 1.51$.

Now consider three consecutive disks that we wish to connect to a single wire of the k-star. Given that $k \geq 8$, the minimum distance between the two non-adjacent disks is $d_{\min} \geq 2\sqrt{2+\sqrt{2}}-2 \approx 1.696$. Therefore, a path simply connecting three disks (and yet still disjoint from the k-star) has greater weight than even the path joining four disks using two wires of the k-star, and thus an optimal path containing one wire of a k-star in the MST contains points from at most two disks of the k-flower. □

Corollary 1. *In the optimal MSTN solution for a k-flower (when k is even), each consecutive pair of disks is connected to a single wire of the k-star via a path of length d.*

This follows immediately from Lemmas 1 and 2. Hence, there are two possible solutions for MSTN on a k-flower where k is an even number (this is the case in our construction). We use this fact to assign a truth value for the variable gadget: one configuration is arbitrarily considered to be `true`, the other `false`. In Figure 3, we show an example where the `true` configuration is used, and every other wire of the k-star has an edge to some point in the k-flower. The `false` configuration would contain edges between the complementary set of wires of the k-star and the disks of the k-flower.

Clause Gadgets The clause gadgets are composed of three wires that meet at a single point. Each wire of the clause gadget is placed so that it terminates at a distance $1+d$ from a tangent point, where the terminal point is collinear with a line of the k-star on the relevant variable gadget. As a result, a line segment of length $2+d$ units can connect the clause gadget to the k-star of a variable gadget, while also intersecting the shared point between two disks of the k-flower. If the truth value of the k-flower gadget matches that of the clause, this means that connecting the clause to the flower requires two units of extra weight, since otherwise the two disks are connected to the k-star with d weight, as outlined in Lemma 1. Therefore, given a clause gadget where at least one literal matches the truth value of the corresponding variable gadget, the clause gadget is connected to the MST with two units of additional weight.

The spinal path wires terminate in positions exactly analogous to those of the clause gadgets so that the analysis is the same. This raises the possibility that the wires of a clause gadget may be connected to two variable gadgets, leaving a gap in the spinal path, but note that such a configuration does not affect the weight of the optimal tree. The spinal path is necessary however, since some variables may not be used by any clauses in an optimal solution.

Lemma 3. *Joining a clause wire to a k-flower that has a truth value differing from that of the clause requires at least ≈ 0.845 units of additional edge weight relative to a configuration with matching truth values.*

Proof. In an optimal MSTN solution on a construction corresponding to a satisfiable 3SAT instance, a pair of disks and a clause wire may be joined to the k-star with weight $2 + d$ units, and an additional adjacent pair of disks may be joined to the k-star with a path of weight d. Therefore, the total weight of the edges incident upon points in four such disks is $2 + 2d$.

Now consider a configuration where the truth value of the literal for each variable in a clause does not match the truth value of the corresponding variable gadgets. Connecting one of the clause gadget wires to the k-star requires an additional weight of $2 + d$, as discussed previously, which intersects points from two disks; call them D_i and D_{i+1}. The neighboring two disks in the k-flower, D_{i-1} and D_{i+2}, are not attached to the k-star by paths like those found in Lemma 1. Rather, each of these adjacent paths may be shortened to $\sqrt{2}$ to cover the two singleton disks. Note that there may be a non-empty sequence of pairs of disks connected as in Lemma 1 before the singleton is reached, creating a section of the flower with an inverted truth value for the variable[7]. Therefore, the net extra weight of such a transition is $2 + d + 2\sqrt{2} - (2 + 2d) = 2\sqrt{2} - d \approx 2.0735$.

A configuration that may require less additional weight is to connect the clause wire to the k-star using a path with points in disks D_{i-1} and D_i (it is a slightly modified configuration from that of Lemma 1). As k increases, the weight of such a path decreases. To minimize the length of the path, suppose that the centers of D_{i-1}, D_i and D_{i+1} are collinear (which occurs when $k = \infty$). Therefore, we can place the center of D_{i-1} at $(1, 1)$, the end of the k-star wire between D_{i-1} and D_i at $(0, 0)$, and the end of the clause wire at $(2 + d, -2)$ (Figure 5). The weight of the path from the k-star to D_{i-1} to the clause wire may be expressed by the function

$$f(\theta) = \sqrt{(1 + \sin\theta)^2 + (1 + \cos\theta)^2} + \sqrt{(1 + d - \sin\theta)^2 + (-3 - \cos\theta)^2},$$

which has a minimum length slightly greater than 3.60 units at $\theta \approx 3.49$ radians. Since this path intersects D_i, it is also the shortest path that includes a point $p_i \in D_i$. Therefore, w.l.o.g. a path connecting a clause wire to a wire in a variable gadget with a mismatched truth value has weight greater than 3.6. Note that such a path does not affect the truth value of the variable gadget, and so D_{i+1} and D_{i+2} may be joined to the k-star with a path of weight d. Therefore, the extra weight incurred for such a configuration is $> 3.6 + d - (2d + 2) \approx 0.845$. □

As described earlier, the terminal points of the clause wires (and the spinal path) are collinear with wires of the k-star. Since we never place these terminal

[7] D_{i+2} may be more generally indexed as D_{i+2+4c}, where there is a block of $4c$ disks in the k-flower joined to the k-star in a truth configuration opposite of that of the neighboring disks in the k-flower. This does not affect the analysis, it simply relocates the singleton disk. Recall that by Lemma 2, such singletons would exist rather than having three disks connected by a path to a single edge of the k-star.

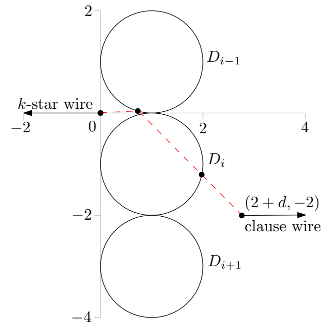


Fig. 5. The shortest possible path is shown (the dashed line) from the end of the clause wire to points in D_i and D_{i-1}, and finally connecting to the k-star wire for D_i and D_{i-1}.

points on adjacent wires of the k-star, the wires need not lie within 4 units of one another, and so there will not be edges directly between different clause wires or between a clause wire and the spinal path.

Reduction We would like to reduce a given instance of planar 3-SAT to the MSTN problem. Note that the given 3-SAT instance is assumed to be embedded on the plane, and there exists a spinal path $P = (V_v, E_P)$ that passes all variable vertices without crossing any edge of G, such that all variable vertices but 2 have degree 2 in P (as mentioned at the beginning of Section 3.2, this restricted version is also NP-hard).

To create the instance of the MSTN problem, we fix the spinal path as a part of the MST, using wires consisting of disks of radius 0. We replace each variable node with a variable gadget as explained. Each clause gadget includes three wires, which we place so that they approach the associated variable gadgets as described.

The wires forming the spinal path, the m clause gadgets, and each of the n k-stars have a fixed weight, call the total weight of all these wires w_{wires}. The remaining weight of the MST is that of connecting to a point from each disk in the k-flowers, and that of connecting each clause gadget. Suppose there exists a satisfying assignment for the 3-SAT instance. Each pair of disks in the k-flowers can be connected with weight d; this will be the case for all but m pairs. The remaining m pairs will be connected with edges that also join to the clause gadgets in the manner described in Section 3.2 with weight $2 + d$. Therefore, assuming that there is a total of i pairs of disks in the k-flowers of the construction, the remaining weight of the MST is $w_{\text{disks}} = id + 2m$. Thus, if there exists a satisfying assignment to the 3-SAT instance, the total optimal weight of the MST is $w_{\text{tot}} = w_{\text{wires}} + w_{\text{disks}}$.

If there is no satisfying assignment, at least one of the clause gadgets must be connected to the MST in the manner described in Lemma 3, which requires an additional weight of at least 0.845. Now suppose there exists an FPTAS for MSTN. Given an instance of planar 3-SAT, we build the MSTN construction and determine w_{tot}. We choose a value of ε so that $\varepsilon < 0.845/w_{\text{tot}}$, and so a $(1 + \varepsilon)$-approximate solution to the MSTN problem may be used to determine whether there is a satisfying assignment for the planar 3-SAT instance. Since the latter problem is NP-hard, we conclude that MSTN does not admit an FPTAS unless P=NP.

4 Conclusions

We considered geometric MST with neighborhoods problems, and established that computing the MST of minimum or maximum weight is hard to approximate in this setting by proving that there is no FPTAS for either problem, assuming $\mathsf{P} \neq \mathsf{NP}$. We provided a parameterized algorithm for the MSTN problem based upon how well separated the disks are from one another. For MAX-MSTN, we showed that a deterministic algorithm that selects disk centers gives an approximation ratio of 1/2. Furthermore, we showed that when the instance of the problem satisfies k-separability, the same approach achieves a constant approximation ratio of $1 - \frac{2}{k+4}$.

For further research, it will be interesting to study this problem under different models of imprecision. Depending on the application, the regions of uncertainty may consist of other shapes, e.g., line segments, rectangles, etc., or they may be composed of discrete sets of points.

References

1. Arkin, E., Hassin, R.: Approximation algorithms for the geometric covering salesman problem. Discrete Applied Mathematics 55(3), 197–218 (1994)
2. de Berg, M., Gudmundsson, J., Katz, M., Levcopoulos, C., Overmars, M., van der Stappen, A.: TSP with neighborhoods of varying size. Journal of Algorithms 57(1), 22–36 (2005)
3. Callahan, P.B., Kosaraju, S.R.: A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. J. ACM 42(1), 67–90 (1995)
4. Dorrigiv, R., Fraser, R., He, M., Kamali, S., Kawamura, A., López-Ortiz, A., Seco, D.: On minimum- and maximum-weight minimum spanning trees with neighborhoods. Tech. Rep. CS-2012-14, University of Waterloo (2012)
5. Dumitrescu, A., Mitchell, J.S.: Approximation algorithms for TSP with neighborhoods in the plane. Journal of Algorithms 48(1), 135–159 (2003)
6. Erlebach, T., Hoffmann, M., Krizanc, D., Mihalák, M., Raman, R.: Computing minimum spanning trees with uncertainty. In: Symposium on Theoretical Aspects of Computer Science. pp. 277–288 (2008)
7. Fiala, J., Kratochvíl, J., Proskurowski, A.: Systems of distant representatives. Discrete Applied Mathematics 145(2), 306–316 (2005)
8. Graham, R.L., Hell, P.: On the history of the minimum spanning tree problem. IEEE Annals of the History of Computing 7(1), 43–57 (1985)
9. Lichtenstein, D.: Planar formulae and their uses. SIAM J. on Computing 11(2), 329–344 (1982)
10. Löffler, M., van Kreveld, M.: Largest and smallest convex hulls for imprecise points. Algorithmica 56, 235–269 (2010)

11. Nešetřil, J., Milková, E., Nešetřilová, H.: Otakar Borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. Discrete Mathematics 233(13), 3–36 (2001)
12. Yang, Y.: On several geometric network design problems. Ph.D. thesis, State University of New York at Buffalo (2008)
13. Yang, Y., Lin, M., Xu, J., Xie, Y.: Minimum spanning tree with neighborhoods. In: Algorithmic Aspects in Information and Management, pp. 306–316 (2007)