Strongly Connected Spanning Digraphs with Bounded Edge Length and Out-Degree

Stefan Dobrev[1], Evangelos Kranakis[2], Oscar Morales Ponce[2], and Milan Plžík[3]

[1] Institute of Mathematics, Slovak Academy of Sciences, Bratislava, Slovak Republic
Stefan.Dobrev@savba.sk

[2] School of Computer Science, Carleton University, Ottawa, Canada
kranakis@scs.carleton.ca, omponce@connect.carleton.ca

[3] Comenius University, Bratislava, Slovak Republic
milan.plzik@gmail.com

Abstract. We study the following problem: Given a set of points in the plane and a positive integer $k > 0$, construct a geometric strongly connected spanning digraph of out-degree at most k and whose longest edge length is the shortest possible. The motivation comes from antennae assignment in sensor networks where each sensor has k directional antennae; the problem is to construct a viable communication network while minimizing energy costs.
The contribution of this is paper is twofold:

- We introduce a notion of *robustness* in geometric graphs. This allows us to provide stronger lower bounds for the edge length needed to solve our problem, while nicely connecting two previously unrelated research directions (graph toughness and multiple directional antennae).
- We present novel upper bound techniques which, in combination with stronger lower bounds, allow us to improve the previous approximation results for the edge length needed to achieve strong connectivity for $k = 4$ (from $2\sin(\pi/5)$ to optimal) and $k = 3$ (from $2\sin(\frac{\pi}{4})$ to $2\sin(\frac{2\pi}{9})$).

1 Introduction

Consider a set of sensors in the plane such that each sensor has k directional antennae and the sum of angles covered by these antennae is at most ϕ. What is the smallest communication range r such that there exists an orientation of the antennae such the resulting communication graph is strongly connected? When $\phi = 0$ the problem is equivalent to the problem of finding spanning digraph of maximum out-degree k that is strongly connected and minimizes the maximal edge length used.

Before giving the formal definition of the problem we will introduce the following notation. Let $\overrightarrow{G}$ denote a weighted directed geometric graph and let $\Delta^+(\overrightarrow{G})$ denote the maximum out-degree of $\overrightarrow{G}$. For any edge (u, v) let $w(u, v)$ denote its weight. We refer to our problem as the *Bottleneck Strongly Connected Spanning Digraph with Bounded Out-degree*(BSCBOD) and define it formally as follows:

Problem 1. Given a weighted complete digraph $\overrightarrow{G}$ and an integer $k \geq 1$, determine a strongly connected spanning subgraph $\overrightarrow{H}$ such that $\Delta^+(\overrightarrow{H}) \leq k$ and the maximum weight is minimum, i.e.,

$$\min\{\max_{w(u,v)\in\overrightarrow{H}} : \overrightarrow{H} \text{ is a strongly connected and } \Delta^+(\overrightarrow{H}) \leq k\}$$

Recall that the UDG (Unit Disk Graph) of a set of n points P with parameter r is the geometric graph, denoted by $U(P; r)$, where the set of vertices is P and vertices u, v are adjacent (with a straight line segment) if and only if $d(u, v) \leq r$, where $d(\cdot, \cdot)$ denotes the Euclidean distance between u and v.

Now we can rephrase our problem in geometric setting as follows:

Problem 2. Given a set of points P and an integer $k \geq 1$. Find the smallest edge length r such that UDG $U(P, r)$ has a strongly connected spanning directed subgraph $\overrightarrow{H}$ with $\Delta^+(\overrightarrow{H}) \leq k$ and construct $\overrightarrow{H}$.

1.1 Related work

When $k = 1$ the problem is equivalent to the well-studied problem of bottleneck traveling salesman problem. Parker and Radin [8] give a 2-approximation when the weights satisfy the triangle inequality and show that it is not possible to approximate to $2 - \epsilon$ unless $P = NP$.

In the Euclidean version of the problem, weights are defined by the Euclidean distance between the two points in $\mathbf{R}^2$. Papadimitriou [7] shows that in this setting the problem remains NP-Complete.

In the context of sensor networks Caragiannis et al. [2] proposed the problem of replacing omnidirectional antennae with directional antennae. They study the setting when each sensor has one directional antenna of a given angle. They showed that the problem is NP-hard when the angle is less than $2\pi/3$ and the communication range is less than $\sqrt{3}$ times the maximum weight of the MST (denoted by r_{MST}).

In [4], the authors study the problem in the context of multiple directional antennae in a wireless sensor network. In this setting, sensors are equipped with directional antennae of given beam-width (angle) and range (radius); their goal is to give algorithms for orienting the antennae and study angle/range trade-offs for achieving strong connectivity. On one hand, when $k \geq 5$, the problem is trivially solved by using each edge of the MST in both directions, since there always exists a MST (Minimum Spanning Tree) on a set of points with maximum degree five. On the other hand, they show that the problem is NP-complete when $k = 2$ even for a scaling factor of 1.3 and/or sum of angles less than $9\pi/20$. They also give an algorithm to compute upper bounds for the cases when $k = 2, 3, 4$. Table 1 summarizes the results given in [4] relating to our problem. A comprehensive survey is presented in [6].

Out-degree	Lower Bound	Upper Bound	Approx. Ratio	Complexity
4	r_{MST}	$2\sin(\pi/5)r_{MST}$	$2\sin(\pi/5)$	Polynomial
3	r_{MST}	$2\sin(\pi/4)r_{MST}$	$\sqrt{2}$	Polynomial
2	r_{MST}	$2\sin(\pi/3)r_{MST}$	$\sqrt{3}$	Polynomial
2	-	-	≤ 1.3	NP-Complete

Table 1: Angle/Range tradeoffs given in [4].

Also related problem is the minimum spanning tree with degree k. In [5] Francke and Hoffmann show that it is NP-Hard to decide whether a given set S of n points in the plane admits a spanning tree of maximum vertex four whose sum of edge lengths does not exceed a given threshold k.

1.2 Results and outline of the paper

This paper contains three major contributions, presented in sections 2, 3, and 4, respectively. In Section 2 we introduce the new concepts of t-strong robustness (σ_t) and k-weak robustness (α_t) of a UDG radius, which are closely related to the well studied concept of graph toughness. The primary motivation comes from the observation that the $1/k$-strong and $1/k$-weak robustness provide a more refined and higher lower bound than the weight of the longest edge of the MST for the problem we study[4].

In Section 3 we present the result that achieves an optimal algorithm for $k = 4$. In Section 4 we combine the technique used in 3 and a modified version of the technique from [4]. This allows us to save one more out-going arc at the cost of having approximation ratio of $2\sin(2\pi/9)$, which is still an improvement over the best previous results from [4]. Due to space constraints technical proofs are left in the Appendix as well as the proofs of Section 4.

Table 2 summarizes the main results of Section 4. We conclude

Out-degree	Lower Bound	Upper Bound	Approx. Ratio	Complexity
4	$\sigma_{1/4}$	$\alpha_{1/4}$	1	$O(n \log n)$
3	$\sigma_{1/3}$	$2\sin(2\pi/9)\alpha_{1/3}$	$\leq 2\sin(2\pi/9)$	$O(n \log n)$

Table 2: Summary of results obtained in this paper

in Section 5 with a discussion of various related issues and open problems.

2 Robustness of Unit Disk Graphs

The concept of toughness of a graph as a measure of graph connectivity has been extensively studied in the literature (see the survey [1]). Intuitively, graph toughness measures the resilience of the graph to fragmentation after subgraph removal.

As defined in [1], a graph G is t-tough if $|S| \geq t\omega(G \setminus S)$, for every subset S of the vertex set of G with $\omega(G \setminus S) > 1$. The toughness of G, denoted $\tau(G)$, is the maximum value of t for which G is t-tough (taking $\tau(K_n) = \infty$, for all $n \geq 1$).

[4] While the strong robustness provides stronger lower bound, the weak robustness is easier to compute and use.

What we are interested in is the toughness of UDGs over a given point set P, and in particular how does the toughness of $U(P, r)$ depend on the radius r. This is expressed in the following definitions:

Definition 1. *[Strong and Weak t-robustness for UDG radius] Let P be a set of points in the plane.*

1. *A subset $S \subseteq P$ is called t-tough if $\omega(U(P \setminus S; r)) \leq |S|/t$. Similarly, a point u is called t-tough if the singleton $\{u\}$ is t-tough.*
2. *The strong t-robustness of the set of points P, denoted by $\sigma_t(P)$, is the infimum taken over all radii $r > 0$ such that for all $S \subseteq P$, the set S is t-tough for the radius r.*
3. *The weak t-robustness of the set of points P, denoted by $\alpha_t(P)$, is the infimum taken over all radii $r > 0$ such that for all $u \in P$, the point u is t-tough for the radius r.*

Note that when P is finite (as in the rest of this paper), it is sufficient to consider only the radii corresponding to pairwise distances between the points of P; in such case it is sufficient to consider minimum instead of infimum.

As we are interested in solving BSCBOD , we are interested in the optimal radius that allows achieving strong connectivity for a given maximum out-degree k.

Definition 2 (Optimal Radius). *For $k \geq 1$, define $r_k(P)$ to be the minimum radius necessary to construct a strongly connected spanning digraph $\overrightarrow{H}$ such that $\Delta^+(\overrightarrow{H}) \leq k$.*

The following theorem motivates our study of robustness of UDG radius:

Theorem 1. $\sigma_{1/k}(P) \leq r_k(P)$

Proof. By contradiction. Assume there exists P and k such that $r_k(P) < \sigma_{1/k}(P)$. Therefore, there must exists $S \subseteq P$ such that S is not $1/k$ tough for radius $r_k(P)$. From the Definition 1 we have $\omega(U(P \setminus S, r_k(P))) > k|S|$, i.e. removing S creates more than $k|S|$ connected components. This is in contradiction with the fact that there exists a solution for BSCBOD with radius $r_k(P)$ and maximal outdegree k, as there are not enough antennae in vertices of S to reach all components of $U(P \setminus S, r_k(P))$. □

Let $r_{MST}(P)$ denote the length of the longest edge of the MST of P. Observe that for $k < 5$, Theorem 1 yields stronger lower bound for $r_k(P)$ than $r_{MST}(P)$.

2.1 Efficiently Computing Weak t-robustness

The weak t-robustness $\alpha_t(P)$ of a set P of points refers to single points and as such it can be computed in polynomial time for a set of n points. A naive algorithm takes $O(n^4)$ time: test in $O(n^2)$ time for a given radius, checking each of the $O(n^2)$ possible radii given by distances between pairs of points. However, this is not the case for the strong t-robustness $\sigma_t(P)$ which depends on all subsets $S \subseteq P$.

As each singleton vertex is also a subset of P, definition 1 directly yields

Lemma 1. *$\alpha_t(P) \leq \sigma_t(P)$, for all t.*

It is not difficult to see that any connected UDG is at least 1/5-robust since the removal of any vertex leaves a maximum of five connected components.

The first important observation is that the 1/4-weak-robustness and 1/4-strong-robustness of a set of points coincide and as a consequence the 1/4-strong-robustness can be computed efficiently.

Theorem 2. *For any set P of points, $\alpha_{1/4}(P) = \sigma_{1/4}(P)$.*

Proof. In view of the observation in Lemma 1 above we only need to show that $\alpha_{1/4}(P) \geq \sigma_{1/4}(P)$. We will show that if for some r the graph $G = U(P; r)$ is not 1/4-robust then there exists a vertex v such that $U(P\backslash\{v\}; r)$ has 5 connected components. Let S be the set such that $U(P\backslash S; r)$ has at least $4|S|+1$ connected components. Consider the bipartite graph $H_S(G) = (S \cup P_S, E_S)$ defined as follows: P_S is the set of connected components of $G \setminus S$, $E_S = \{\{u, v\} : u \in S, v \in P_S$ such that there is an edge in G between u and a vertex from $P_S\}$.

Note that the maximal degree of vertices from S is 5: Assume the converse, i.e. there is a vertex $u \in S$ of degree at least 6. This means there exist edges $\{u, v_i\}$ for $i = 1, 2, \ldots 6$ such that $\{u, v_i\} \in G$ and each v_i is from a different connected component of $U(P \setminus S; r)$. However, at least one of the angles between these six edges must be at most $\pi/3$ and therefore they cannot all lead to different connected components.

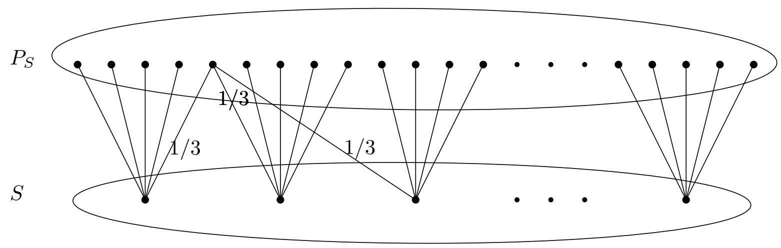

Fig. 1: bipartite graph $H_S(G) = (S \cup P_S, E_S)$.

Let us assign a weight $w(e)$ to each edge as follows: Each vertex v from P_S equally distributes weight 1 among its incident edges, i.e., $1/deg(v)$. Since P_S is an independent set of $H_S(G)$, each edge is given a unique weight $1/i$ for some i. Note that

$$\sum_{u \in S} \sum_{e=\{u,\cdot\}} w(e) = \sum_{v \in P_S} \sum_{e=\{\cdot,v\}} w(e) = \sum_{v \in P_S} 1 = |P_S| > 4|S|.$$

Therefore, there must exist a vertex u from S such that $\sum_{e=u,\cdot} w(e) > 4$ However, since the weight of each edge is $1/i$ for some i and the maximal degree of vertices in S is at most 5, this is only possible if at least four edges incident to u have weight 1 and one has weight greater than 0, i.e., $U(P \setminus \{v\}; r)$ has 5 connected components. $\square$

In fact, 1/4-weak-robustness (and more generally, $1/i$-weak-robustness for $i < 5$) can be computed much more efficiently than a trivial $O(n^4)$ algorithm. The basic idea is to maintain a tree H of red and black vertices, where the black vertices represent bi-connected components (blocks) and the red vertices represent the separator vertices. Each vertex v of P points (in variable $h(v)$) to its representative in H. Furthermore, each red vertex maintains in variable $c(v)$ (separator degree) the number of components it connects. Initially, H starts as the MST of P; subsequently the edges are added according to their increasing length. The $h(\cdot)$ pointers allow to efficiently determine whether an edge connects two different vertices of H. If adding an edge e closes a cycle in H, the separator degrees of the red vertices (except those incident to e) in this cycle are reduced by one. The process is repeated by taking progressively longer edges until no red vertices remain. As the $h(\cdot)$ pointers can be maintained using standard techniques with the overall cost of $O(n \log n)$ and the overall cost of processing and collapsing created cycles is $O(n)$, the overall cost if $O(|E| + n \log n)$, where E is the set of processed edges.

The second idea of the algorithm comes from the observation that only a very limited set of edges of size $O(n)$ will ever be processed.

Theorem 3. *For every point of P, weak $1/i$-robustness, for $1 \leq i < 5$, can be computed in time $O(|P| \log |P|)$.*

Algorithm 1 Algorithm $\mathcal{A}\alpha_{1/i}$

Compute the MST T of the point set P.
Set the colour of leaves of T to *black*, colour *red* the remaining vertices
Set $H \leftarrow T$.
for each vertex $v \in T$ **do**
    Set $c(v)$ to the degree of v in T.
    Set $h(v) \leftarrow v$.
end for
for each edge $e = \{u, v\} \notin T$, processed in the order of increasing length **do**
    if $h(u) \neq h(v)$ **then**
        Let C_e be the cycle closed by $(h(u), h(v))$ in H.
        for each red vertex $w \in C_e \setminus \{h(u), h(v)\}$ **do**
            $c(w) \leftarrow c(w) - 1$
            Output $\alpha_{1/c(w)}(w)) = \max(r_{MST}, d(u, v))$
            if $c(w) = 1$ **then**
                Set the colour of w to *black*.
            end if
        end for
        Remove all edges of C_e from H.
        Add a black vertex x_e to H and connect it to all vertices of C_e.
        Collapse the all-black component of x_e into a single black vertex x'_e.
        Unify $h(\cdot)$ for the vertices of the bi-connected component represented by x'_e.
    end if
end for

The proof of the theorem will be provided after proving the following lemmas. Let us denote by $TUDG(P, r)$ the graph $T \cup UDG(P, r)$, where T is the MST of P.

Lemma 2. *Algorithm $\mathcal{A}\alpha_{1/i}$ can be implemented with time complexity $O(n \log n)$.*

Proof. It is known that the Delaunay triangulation and the MST can be computed in $O(n \log n)$ time [3]. The set of edges in $\bigcup_{w \in P} RNG_w(P)$

to be processed on line 8 is of size $O(n)$ and can be computed from $Del(P)$. Therefore, it can be computed in $O(n \log n)$. We will show that the cumulative cost of processing on lines 10-21 is $O(n \log n)$ as well.

Everything done on lines 10-20 is linear in the size of the cycle C_e. We charge the cost of this to the red vertices of C_e. Since there are no neighbouring black vertices in H, at least half of the vertices in C_e are red and each one of them will be charged a constant. We show that each red vertex w is charged constant number of times: a) w will be charged at most 5 times in cycles where w is not incident to $u(u)$ or $h(v)$, because in each of those cases, $c(w)$ is decremented. b) the maximal degree of w in $T \cup \bigcup_{w \in P} RNG_w(P)$ is constant and therefore it will be charged constant number of times whenever its $c(w)$ is not decremented: The sum of two consecutive angles incident to a vertex must be at least $\pi/3$, otherwise there would be two vertices in $S(u, |e|) \cap S(v, |e|)$, which we know by Lemma 6 is not possible.

Let us now evaluate the total cost of line 21. The components corresponding to the black vertices might have a lot of vertices, leading to a lot of work changing those $h(.)$ values. What we do is we identify the largest component and change the $h(.)$ values for all vertices of the other components. In this way, whenever a vertex changes its $h(.)$ value, the size of its component at least doubles. The overall cost of line 21 is therefore $O(n \log n)$. □

Theorem 3 follows directly from Lemma 2 and Lemma 5.

3 Digraph with Max Out-Degree Four

In this section, we prove that given a set of n points in the plane, there is always a strongly connected spanning digraph with max out-degree four and optimal length. In fact, it can be constructed in $O(n \log n)$.

Theorem 4. $r_4 = \alpha_{1/4}$.

Proof. Let P be a set of points and T be the MST on P. We may assume that T has maximum degree five. Consider the set S of vertices of degree five in T. Let r_4 be the radius of $\alpha_{1/4}$ obtained from Algorithm $A\alpha_{1/i}$. For each vertex $v \in S$ we compute the set E_v of shortest edges of length at most r_4 that join two distinct components of $T \setminus \{v\}$. Let $G = T$ and $\overrightarrow{G}$ be the strongly connected graph obtained from orienting in both direction each edge of G. We will add new edges to G in order to form cycles that include every vertex in S. Thus, the max out-degree of $\overrightarrow{G}$ is decreased to four by orienting the cycles in one direction. We will process edges in $\bigcup_{v\in S} E_v$ in descending order according to the hop-length of the cycle that they form with the edges of T.

Let $\{u, w\} \in \bigcup_{v\in S} E_v$ be the edge that forms the longest cycle C. We consider two cases:

- $|C| > 3$. Let $S(x, r)$ denote the open disk centered at x of radius r. Since $\{u, w\}$ is the shortest edge, the lune formed by $S(u, d(u, w)) \cap S(w, d(u, w))$ is empty. Therefore, the angle that $\{u, w\}$ forms with the edges incident to u and w is at least $\pi/3$. Hence, u and w have degree at most four in T. Add $\{u, v\}$ to G and orient C in clockwise order in $\overrightarrow{G}$. Since $\{u, v\}$ is the shortest edge that forms the longest cycle for each vertex in $S \cap C$, we can remove from S all the vertices in C, i.e., $S = S \setminus C$. Observe that vertices in C have out-degree at most four in $\overrightarrow{G}$ and the strong connectivity is not broken.
- $|C| = 3$. Observe that $\{u, v\}, \{w, v\} \in T$. Let $v \in C \setminus \{u, w\}$. Consider the two components G_u, G_w obtained from $G \setminus \{v\}$. Assume that $u \in G_u$ and $w \in G_w$. Since $\{u, w\}$ creates the longest cycle, there does not exist an edge distinct to $\{u, w\}$ in $\bigcup_{x\in S\setminus\{v\}} E_x$ that joins G_u and G_w. However, since $\{u, v\}, \{v, w\}$ are in T, the lune formed by $S(u, d(u, w)) \cap S(w, d(u, w))$ contains v. Therefore, u and w may have degree five in T, since the angle that $\{u, w\}$ forms with $\{u, v\}$ and/or $\{v, w\}$ is less that $\pi/3$.

  Consider the cycle $vu \dots u'v$ in $G_u \cup \{v\}$ and the cycle $vw \dots w'v$ in $G_w \cup \{v\}$. Let $G = G \cup \{u, w\}$ and remove v from S. Remove $\{v, u\}$ from G if $u \neq u'$. Similarly, remove $\{v, w\}$ from G if $w \neq w'$. Orient the new cycle in $\overrightarrow{G}$ as $vu' \dots uw \dots w'v$. Observe that v has out-degree at most four in $\overrightarrow{G}$ and the out-degree of the vertices in the new cycle does not increase. Furthermore, the strong connectivity is not broken.

When S is empty, $\overrightarrow{G}$ will have max out-degree four. The theorem follows since the strong connectivity of $\overrightarrow{G}$ is never broken. The pseudocode is given in Algorithm 2. □

The following Lemma is a simple observation of the construction given in Theorem 4 and is given without proof.

Lemma 3. *Let $\overrightarrow{G}$ be the digraph obtained from 4. For each vertex v the angle that v forms between out-going edges is at least $\pi/3$ and the angle that v forms between in-going edges is at least $\pi/3$.*

Theorem 5. *Algorithm 2 can be computed in $O(n \log n)$; where n is the number of points.*

Proof. The MST can be computed in $O(n \log n)$ time. From Lemma 7, the number of edges to be processed are $O(n)$. Thus, it takes $O(n \log n)$ time to sort the edges to be processed in line 8. Since every edge is in at most one cycle and there are $O(n)$ edges, the time to complete the construction is $O(n \log n)$.

4 Digraph with Max Out-Degree Three

In this section, we prove that given a set of n points, there exists a strongly connected spanning digraph with max out-degree three and length bounded by $2 \cdot \sin(2\pi/9) \cdot \alpha_{1/3}$ and it can be constructed in $O(n \log n)$.

Theorem 6. *There exist UDGs such that $\alpha_{1/3} \geq \sigma_{1/3}$.*

Theorem 7. $r_3 \leq 2 \cdot \sin(2\pi/9) \cdot \alpha_{1/3}$.

Theorem 8. *Algorithm 3 that constructs a strongly connected spanning digraph with max out-degree three can be computed in $O(n \log n)$; where n is the number of points.*

5 Conclusion

In this paper we studied the problem of how to construct from a set of points in the plane and a positive integer $k > 0$, a geometric strongly connected spanning digraph of out-degree at most k and whose longest edge length is the shortest possible. We proved that the problem can be solved with optimal edge length in polynomial time when the out-degree is at least 4. To quantify the problem we introduced the concept of k-robustness for a UDG radius. We also improved the previous best known upper bound when the out-degree is at most 3. However, it is unknown whether the problem can be solved optimally in polynomial time when the out-degree is at most 3.

References

1. D. Bauer, H. Broersma, and E. Schmeichel. Toughness in graphs–a survey. *Graphs and Combinatorics*, 22(1):1–35, 2006.
2. I. Caragiannis, C. Kaklamanis, E. Kranakis, D. Krizanc, and A. Wiese. Communication in wireless networks with directional antennae. In *20th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'08)*, pages 344–351, Munich, Germany, June 14–16 2008. IEEE/ACM.
3. M. De Berg, O. Cheong, and M. Van Kreveld. *Computational geometry: algorithms and applications*. Springer-Verlag New York Inc, 2008.
4. S. Dobrev, E. Kranakis, D. Krizanc, O. Morales Ponce, J. Opatrny, and L. Stacho. Strong connectivity in sensor networks with given number of directional antennae of bounded angle. In *Proceedings of the 4th Annual International Conference on Combinatorial Optimization and Applications (COCOA 10). Part II, LNCS 6509*, pages 72–86, Big Island, Hawaii, Dec 18-20 2010. Springer-Verlag.
5. A. Francke and M. Hoffmann. The euclidean degree-4 minimum spanning tree problem is np-hard. In *Proceedings of the 25th annual symposium on Computational geometry*, pages 179–188. ACM, 2009.
6. E. Kranakis, D. Krizanc, and O. Morales. Maintaining connectivity in sensor networks using directional antennae. In S. Nikoletseas and J. Rolim, editors, *Theoretical aspects of Distributed Computing in Sensor Networks*, chapter 3, pages 59–84. Springer, 2010. ISBN 978-3-642-14848-4.
7. C.H. Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.
8. R.G. Parker and R.L. Rardin. Guaranteed performance heuristics for the bottleneck traveling salesman problem. *Operations Research Letters*, 2(6):269–272, 1984.

Appendix

The proof of the Theorem 3 will be provided after proving the following lemmas. Let us denote by $TUDG(P, r)$ the graph $T \cup UDG(P, r)$, where T is the MST of P.

Lemma 4. *The following invariant holds for $\mathcal{A}\alpha_{1/i}$ at the beginning of each iteration of the loop. Let $e = \{u, v\}$ be the last processed edge.*

- *H is a tree*
- *A vertex is red if and only if it is a separator vertex of $TUDG(P, d(u, v))$. Furthermore, for every red vertex w : $h(w) = w$ and $c(w)$ is the number of connected components of $TUDG(P, d(u, v)) \setminus \{w\}$.*
- *$\forall x, y \in P : h(x) = h(y)$ if and only if x and y lie in the same block of $TUDG(P, d(u, v))$.*

Proof. By induction over loop iterations. The base step at the beginning of the first iteration of the loop follows from the construction and the definitions.

The induction step:

- H remains a tree, because each created cycle is replaced by a star graph. The subsequent collapse of the all-black component just reduces the size of the resulting tree.
- Let $u = \{u, v\}$ be the last processed edge. If $h(u) = h(v)$, by induction hypothesis the edge lies within the same block and adding it does not influence any separator vertex. Whenever $h(u) \neq h(v)$, the edge $(h(u), h(v))$ creates a cycle C_e in H; this corresponds to a cycle C'_e in $TUDG(P, d(u, v))$ (since the edges are added in order of increasing length, all shorter edges of $UDG(P, d(u, v))$ have already been processed). The cycle C'_e connects two components for each separator (by induction hypothesis red) vertex lying on this cycle, except the vertices incident to e. This is exactly reflected on line 12 – by induction hypothesis the red vertices processed on line 12 are exactly the separators lying on the cycle C'_e. The invariant is maintained by recoloring red vertices to red on line 15 whenever the separation degree reaches 1, i.e. the vertex stops being a separator. Note also that $h(w)$ remains equal to w while w remains red.
- This invariant is maintained on line 21.

This completes the proof of the lemma. $\square$

From the second point of the previous lemma and from line 13 of $\mathcal{A}\alpha_{1/i}$ we get

Lemma 5. *Algorithm $\mathcal{A}\alpha_{1/i}$ correctly computes values $\alpha_{1/i}(v)$ for $1 \leq i < 5$ and for each vertex $v \in P$.*

Proof. The lemma follows easily from the second point of the previous lemma and from line 13 of $\mathcal{A}\alpha_{1/i}$.

Now we will prove that the number of processed edges is of the order of $O(n)$. Let $RNG(P)$ denote the Relative Neighbourhood Graph of the point set P and let $RNG_w(P) = RNG(P \setminus \{w\})$. The following simple lemma is crucial:

Lemma 6. *Let $e = \{u, v\}$ be the shortest edge connecting two connected components of $TUDG(P, r) \setminus \{w\}$ where $r < d(u, v)$. Then either $e \in RNG(P)$ and it forms angle at least $\pi/3$ with all incident edges of $TUDG(P, r) \setminus \{w\}$, or $e \in RNG_w(P)$ and only the angles $\angle wuv$ or $\angle wvu$ might be smaller than $\pi/3$.*

Proof. Let $S(x, r)$ denote an open (i.e. not containing the boundary vertices) sphere/disk centered at vertex x with radius r. If $S(u, |e|) \cap S(v, |e|)$ is empty then $e \in RNG(P)$ and the lemma holds. Consider now a vertex $p \in S(u, |e|) \cap S(v, |e|)$. Let C_u and C_v denote the components of $TUDG(P, r)$ containing u and v, respectively. $p \notin C_u$, otherwise (p, v) would be the shortest edge connecting C_u and C_v. Analogously $p \notin C_v$. Therefore, the only possibility is $p = w$. After removing w, $S(u, d(u, v)) \cap S(v, d(u, v))$ becomes empty, i.e. $e \in RNG_w(P)$. $\square$

Consider now an edge $e = \{u, v\}$ such that $h(u) \neq h(v)$. By Lemma 4 u and v belong to different blocks of $G' = TUDG(P, d(u', v'))$ where $\{u', v'\}$ is the last edge processed before e. Therefore there exists a separator vertex w such that u and v are in different components of $G' \setminus \{w\}$. Since the edges are processed in the order of increasing length, e must be the shortest edge connecting different components of $G' \setminus \{w\}$. By Lemma 6 $e \in RNG_w(P)$. This means that it is enough to consider on line 8 only the edges from $\bigcup_{w \in P} RNG_w(P)$.

Lemma 7. *The number of edges in $\bigcup_{w \in P} RNG_w(P)$ is in $O(n)$.*

Proof. We will actually show that $\bigcup_{w \in P} Del_w(P)$ has $O(n)$ edges, where $Del(P)$ is the Delauney triangulation of P and $Del_w(P) = Del(P \setminus \{w\})$. Since $\forall S : RNG(S) \subseteq Del(S)$, this is sufficient.

The crucial observation is that $Del_w(P) \setminus Del(P)$ has at most $d(w)$ edges, where $d(w)$ is the degree of w in $Del(P)$. Summing up over all $w \in P$ we have

$$\begin{aligned}
|\bigcup_{w \in P} Del_w(P)| &\leq |Del(P) \cup \bigcup_{w \in P} (Del_w(P) \setminus Del(P))| \\
&\leq |Del(P)| + \sum_{w \in P} d(w) \\
&= |Del(P)| + 2|Del(P)| \\
&= 3|Del(P)| \in O(n),
\end{aligned}$$

which completes the proof of the lemma. □

Proof (Theorem 6). We present a counter example in Figure 2 where the set $S = \{u_1, u_2, u_3\}$ such that $G \setminus S$ leaves 10 connected components.

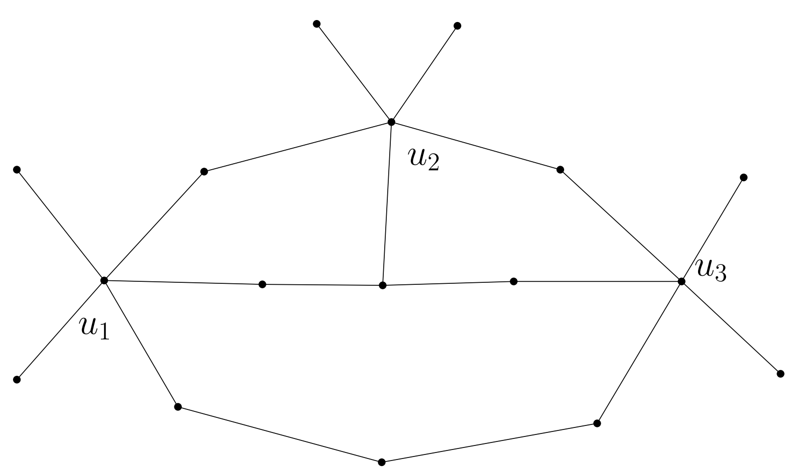

Fig. 2: $\alpha_{1/3} \geq \sigma_{1/3}$.

□

Algorithm 2 Strongly connected spanning digraph with max out-degree four with radius $r_4 = \alpha_4$ on a set of points

Let $G = MST(P)$ and $\overrightarrow{G}$ be the strongly connected digraph obtained from orienting every edge of G in both directions.
Set S be the set of vertices of G of degree five.
Let $r_4 = \mathcal{A}\alpha_4$.
for each v in S **do**
    Set E_v be the shortest edges of length at most r_4 that join two components of $G \setminus \{v\}$.
end for
while S is non-empty **do**
    Let $\{u, w\}$ be edge that forms the longest cycle C in G.
    Let $G = G \cup \{u, w\}$.
    if $|C| > 3$ **then**
        Orient the cycle C in clockwise order.
        Let $S = S \setminus C$.
    else
        Let $v = C \setminus \{u, w\}$.
        Let G_u and G_w be the components of $G \setminus \{v\}$ such that $u \in G_u$ and $v \in G_w$.
        Let $vu \dots u'v$ be a cycle in $G_u \cup \{v\}$ and $vw \dots w'v$ be a cycle in $G_w \cup \{v\}$
        Let $G = G \cup \{u, w\}$ and $S = S \setminus \{v\}$.
        if $u \neq u'$ **then**
            Let $G = G \setminus \{v, u\}$.
        end if
        if $w \neq w'$ **then**
            Let $G = G \setminus \{v, w\}$.
        end if
        Orient the cycle in $\overrightarrow{G}$ as $vu' \dots uw \dots w'v$.
    end if
end while
Return G.

Proof (Theorem 7). Let P be a set of points and $\overrightarrow{G'}$ be the strongly connected digraph of max-out degree four obtained from Theorem 4. Let $G = G'$ and $\overrightarrow{G} = \overrightarrow{G'}$. Consider the set S of vertices of G such that $\omega(G \setminus \{v\}) = 4$. Let r_3 be the radius of $\alpha_{1/3}$ obtained from Algorithm $A\alpha_{1/i}$. For each vertex $v \in S$ we compute the set E_v of shortest edges of length at most r_3 that join two distinct components of $G \setminus \{v\}$.

As in Theorem 4 we will add new edges to G in order to form cycles that include every vertex in S. We will prove that the out-degree of the vertices in S can be always decreased without affecting the max out-degree providing that the cycles have length three. However, when the cycles have length greater than three, the orientation of the cycles does not guarantee that every vertex has max out-degree three. Thus, when we orient a cycle of length greater than three, for each vertex v in the cycle that has out-degree four and $\omega(G\backslash\{v\}) = 3$, we will process the shortest edges that connects two components to form cycles of length three, so that the out-degree can be decreased. However, these new edges may have length greater than r_3. We will prove that the length is bounded by $2\sin(2\pi/9) \cdot r_3$.

Firstly we process edges in $\bigcup_{v\in S} E_v$ in descending order according to the hop-length of the cycle that they form with the edges of G'. Let $\{u, w\} \in \bigcup_{v\in S} E_v$ be the edge that forms the longest cycle C. We consider two cases:

- $|C| > 3$. Since $\{u, w\}$ is the shortest edge, the lune formed by $S(u, d(u, w)) \cap S(w, d(u, w))$ is empty. Therefore, the angle that $\{u, w\}$ forms with the edges incident to u and w is at least $\pi/3$. Let $G = G \cup \{u, w\}$ and orient C in clockwise order in $\overrightarrow{G}$. Observe that the orientation of C does not break the connectivity. However, it does not guarantee that the out-degree is decreased to 3. Remove from S each vertex $v \in S \cap C$ such that $\omega(G \setminus \{v\}) \leq 3$. Consider the set S' of vertices of out-degree four in $C \backslash S$. For each vertex v' in S', let $E_{v'}$ be the shortest edge $\{u', w'\}$ connecting two distinct components of $G\backslash\{v'\}$ such that $\{v', u'\}$ and $\{v', w'\}$ exist in G'. Add v' to S.
  It remains to prove that $d(u', w') \leq 2\sin(2\pi/9) \cdot r_3$. Let (w_0, v'), $(v', u_0) \in C$ and (v', u_1), (v', u_2) and (v', u_3) be the out-going edges of v' in $\overrightarrow{G}$. From Lemma 3 and the fact that $S(u, d(u, w)) \cap S(w, d(u, w))$ is empty, $\angle(w_0 v' u_0) \geq \pi/3$ and $\angle(u_j v' u_k) \geq \pi/3$. Furthermore, at least two out-going edges $(v', u_a), (v' u_b)$ are in the same component. Therefore, there exist at least two vertices $\{u', w'\}$ in distinct components with angle at most
  $$\frac{2\pi - \angle(w_0 v' u_0) - \angle(u_a v' u_b)}{3} = \frac{4\pi}{9},$$
  i.e., $d(u', w') \leq 2\sin(2\pi/9) \cdot r_3$ since the longest edge of G so far is r_3.
- $|C| = 3$. Observe that $\{u, v\}, \{w, v\} \in T$. Let $v \in C \setminus \{u, w\}$ and G_u, G_w be the two components of $G \setminus v$ such that $u \in$

G_u and $w \in G_w$. Since $\{u, w\}$ creates the longest cycle, there does not exist an edge distinct to $\{u, w\}$ in $\bigcup_{x \in S \setminus \{v\}} E_x$ that joins G_u and G_w. However, since $\{u, v\}, \{v, w\} \in G'$, the lune formed by $S(u, d(u, w)) \cap S(w, d(u, w))$ contains v. Therefore, $\min(\angle(wuv), \angle(vuw)) \leq \pi/3$. Consider the cycle $C' = vu \ldots u'v$ in $G_u \cup \{v\}$ and the cycle $C'' = vw \ldots w'v$ in $G_w \cup \{v\}$. Let $G = G \cup \{u, w\}$ and remove v from S. Remove $\{v, u\}$ from G if $u \neq u'$. Similarly, remove $\{v, w\}$ from G if $w \neq w'$. Orient the new cycle in $\overrightarrow{G}$ as $vu' \ldots uw \ldots w'v$. Observe that v has out-degree at most three in $\overrightarrow{G}$ and the out-degree of the vertices in the cycle does not increase. Furthermore, the strong connectivity is not broken.

This process maintains $\overrightarrow{G}$ strongly connected. When S is empty, the max out-degree of $\overrightarrow{G}$ is three. The theorem follows. $\square$

Proof (Theorem 8). From Theorem G can be computed in $O(n \log n)$ time. From Lemma 7, the number of edges to be processed are of the order of $O(n)$. Thus, it takes $O(n \log n)$ time to sort the edges to be processed in line 8. Since every edge is in a constant number of cycles the time to complete the construction is $O(n \log n)$.

Algorithm 3 Strongly connected spanning digraph with max out-degree three with radius $2\sin(2\pi/9)\cdot\alpha_3$ on a set of points

Let G and $\overrightarrow{G}$ be the strongly connected digraph obtained from Algorithm 2.
Set S be the set of vertices of G such that $\omega(G\setminus\{v\})=4$.
Let $r_3=\mathcal{A}\alpha_3$.
for each v in S **do**
  Set E_v be the shortest edges of length at most r_3 that join two components of $G\setminus\{v\}$.
end for
while S is non-empty **do**
  Let $\{u,w\}$ be edge that forms the longest cycle C in G.
  Let $G=G\cup\{u,w\}$.
  if $|C|>3$ **then**
    Orient the cycle C in clockwise order.
    Remove form S all the vertices v in $S\cap C$ such that $\omega(G\setminus\{v\})\leq 4$.
    Let S' be the set of vertices of out-degree four in $C\setminus S$.
    for each v' in S' **do**
      Set $E_{v'}$ be the shortest edge $\{u',w'\}$ joining two components of $G\setminus\{v'\}$ such that $\{u',v'\},\{w',v'\}\in G$.
      Let $S=S\cup\{v\}$.
    end for
  else
    Let $v=C\setminus\{u,w\}$.
    Let G_u and G_w be the components of $G\setminus\{v\}$ such that $u\in G_u$ and $v\in G_w$.
    Let $vu\ldots u'v$ be a cycle in $G_u\cup\{v\}$ and $vw\ldots w'v$ be a cycle in $G_w\cup\{v\}$
    Let $G=G\cup\{u,w\}$ and $S=S\setminus\{v\}$.
    if $u\neq u'$ **then**
      Let $G=G\setminus\{v,u\}$.
    end if
    if $w\neq w'$ **then**
      Let $G=G\setminus\{v,w\}$.
    end if
    Orient the cycle in $\overrightarrow{G}$ as $vu'\ldots uw\ldots w'v$.
  end if
end while
Return G.