Minimizing Movement to Establish the Connectivity of Randomly Deployed Robots

K. Selim Engin, Volkan Isler
Computer Science and Engineering, University of Minnesota
{engin003, isler}@umn.edu

Abstract

We study the following connectivity formation problem: Robots equipped with radio transmitters with a bounded communication range are scattered over a large area. They would like to relocate so as to form a connected network as soon as possible. Where should each robot move?

We present an $O(\sqrt{n})$-factor approximation algorithm for this problem when n robots are initially distributed uniformly at random in a bounded area. In addition to analytical proofs, we verify the performance of our algorithm through simulations.

Introduction

Many multi-robot tasks require the robots to communicate. In the absence of a communication infrastructure, the robots must form a communication network themselves. Imagine a set of robots with communication devices that have a maximum communication range. The robots are scattered over an environment in such a way that their communication network is not connected. How should the robots move so as to form a connected network in the shortest time? In this paper, we study this problem using a centralized approach for the case when the initial locations of the robots are chosen uniformly at random.

Demaine et al. introduced a set of movement minimization problems where the input is a graph along with a set of pebbles placed on the vertices (Demaine et al. 2009). The goal is to move the pebbles along the edges, so that the graph induced by their final locations satisfies a desired property such as connectivity or independence. Among others, they studied the ConMax problem where the objective is to minimize the maximum movement so as to achieve connectivity. The problem is shown to be NP-complete even in the presence of an oracle that knows the pebble positions and guides them to their final configuration. When there exists an oracle, a pebble does not need to search for others, and it can directly go to its final position provided by the oracle. The authors presented an $O(\sqrt{n})$-approximation factor algorithm for this setting with a centralized approach. In their formulation, communication and motion are coupled (edges represent both motion steps and communication radius) and the results do not directly apply to the Euclidean case. This case was studied by (Anari et al. 2016) who presented an $O(n)$-factor approximation algorithm. Anari et al. also proved the problem to be $(2 - \frac{\sqrt{2}}{2})$-inapproximable.

Solutions using a centralized approach, however, may not be applicable in all scenarios, such as in the case of failures or disconnections in the communication network. Poduri and Sukhatme analyzed the time to achieve connectivity using a decentralized approach (Poduri and Sukhatme 2007). With the robot movement modeled as random direction mobility, they show the time to reach connectivity decreases by $O(1/\sqrt{n})$ as the number of robots increases.

The connectivity problem was also studied from a connectivity control perspective in a number of papers. Zavlanos and Pappas developed a gradient-based method for maintaining the connectivity of a mobile network (Zavlanos and Pappas 2007). In their centralized control framework, the mobile robot network is represented as a dynamic graph, and the potential field is defined over the Laplacian matrix of this graph. Their framework allows tasks including connectivity maintenance and tracking of a leader robot while avoiding collisions. In a follow-up study, they proposed a decentralized controller capable of maintaining the connectivity of the network while the agents perform secondary objectives, assuming the initial network is connected. However, due to this assumption, the framework they present is not applicable to the connectivity maintenance problem described in (Demaine et al. 2009) where the robot configurations are initially disconnected. Michael et al. (Michael et al. 2009) later conducted an experimental study to demonstrate the distributed connectivity control algorithms proposed in (Zavlanos and Pappas 2008).

In this paper, we revisit the problem studied in (Anari et al. 2016). We improve the result for random deployments and present an $O(\sqrt{n})$-factor approximation algorithm. Our result has many practical applications including settings where robots have two modes of communication: a long range, low bandwidth communication mode such as XBee and a shorter range but high capacity mode such as WiFi. The robots can exchange their locations by XBee but might want to form a connected WiFi network, for example to exchange videos. Our algorithm provides an efficient method for doing so.

We start by formalizing the problem.

Notations and Environment Model

In this section we present the notation and model we use throughout the paper.

Environment Model: Consider n robots each with a prescribed connectivity radius r. In this model, the robots are distributed uniformly at random over a bounded circular area of radius L.

The initial positions of the robots are represented as $\mathcal{X} = \{x_1, x_2, ..., x_n\}$. The Euclidean distance between two robots x_i and x_j is denoted as $d(x_i, x_j)$. We represent the final positions of the robots in a connected configuration by $\mathcal{X}'$.

In our analysis we will use the smallest circle C enclosing the initial positions of the robots. The boundary of this circle C is denoted as ∂C, and we let R be the radius of C.

We will use some properties from graph theory, and therefore we examine the relationship between the representation of the robots in the Euclidean domain and graphical settings. By $K(\mathcal{X})$, we denote the complete graph induced on $\mathcal{X}$ where the edge weights are Euclidean pairwise distances. The subgraph $G(\mathcal{X}; r)$ of $K(\mathcal{X})$ includes every edge in $K(\mathcal{X})$ with length less than or equal to r.

We consider connected components of $G(\mathcal{X}; r)$ as clusters, and denote the set of disjoint clusters by $\Pi = \{P_1, ..., P_m\}$. The number of robots in a cluster P_i is shown as $|P_i|$, hence $\sum_{i=1}^{m} |P_i| = n$. The distance between two clusters P_i and P_j is denoted with $d(P_i, P_j)$, and it evaluates to $\min_{u \in P_i, v \in P_j} d(u, v)$.

Borrowing from the notation in (Steele and Tierney 1986), we denote the length of the longest nearest-neighbor link in $K(\mathcal{X})$ by Z_n. Finally, the connectivity distance M_n denotes the smallest edge length such that $G(\mathcal{X}; M_n)$ is a connected graph.

For an instance σ of the problem, let $A(\sigma)$ be the solution of our approximation algorithm, and $OPT(\sigma)$ be the solution of the optimal strategy. We compute the approximation ratio of an algorithm A as $\mathbb{E}[A(\sigma)/OPT(\sigma)]$, where the expectation is computed over all possible instances of the problem, and each instance is equally likely.

An approximation algorithm has α-approximation if $\mathbb{E}[A(\sigma)/OPT(\sigma)] \leq \alpha$. We define α to be 1 if $OPT(\sigma)$ is zero, that is when the initial configuration is connected. We assume that A can check whether this condition is true, and produces a solution $A(\sigma) = 0$, if it is the case.

We are now ready to formally state the problem.

Problem Statement: Given a fixed connectivity radius r, and positions of n uniformly distributed robots in a bounded circular area of radius L, compute the final locations of the robots such that their configuration forms a connected communication network and the expected maximum movement of a robot is minimized.

In the next section, we continue with results from the existing literature that are relevant for our analysis.

Preliminaries

In this section we present the existing results we use in our analyses.

The first result, due to (Penrose 1997), establishes the relationship between the connectivity distance and the number of robots. Penrose proves that the value of $(n\pi M_n^2 - \log n)$ converges to the double exponential distribution for points independently and uniformly distributed on a unit ball in the two dimensional Euclidean space.

Result 1. *(Penrose 1997) If M_n is the smallest radius such that $G(\mathcal{X}; M_n)$ is connected, then the following result serves as an upper bound for M_n.*

$$\lim_{n \to \infty} Pr[n\pi M_n^2 - \log n \leq s] = \exp(-e^{-s}) \qquad (1)$$

for any $s \in \mathbb{R}$.

Recall that Z_n is the length of the longest nearest neighbor link (in other words, for each node, we compute the distance of its nearest neighbor and take the maximum value in this set). The following result for lower bounding Z_n is by (Steele and Tierney 1986).

Result 2. *(Steele and Tierney 1986) For independently and uniformly distributed n points on a unit ball in $\mathbb{R}^2$, the following relationship between Z_n and n holds:*

$$\lim_{n \to \infty} Pr[Z_n^2 \geq (t + \log n)/\pi n] = 1 - \exp(-e^{-t}) \qquad (2)$$

for any $t \in \mathbb{R}$.

Here, we remark that the function $\log(n)/n$ acts as a threshold both for the connectivity and the largest nearest-neighbor distances. Penrose shows as n increases M_n/Z_n approaches to 1 almost always. We also note $M_n \geq Z_n$ since having no isolated points does not imply connectivity.

Since these are bounds in the limit, we investigated their tightness through simulations. Figure 1 shows the relation between M_n, Z_n and their bounds for varying number of robots in a fixed area. We compare the theoretical bounds above with values computed from simulations. The s and t values for M_n and Z_n bounds were 23 and -1.93, respectively. The chosen values for s and t make the bounds hold with a probability approaching to 1.

In Figure 1 we see that with high probability the upper bound for M_n holds, and it is tight. We also observe that M_n is always larger than Z_n.

The network formation algorithm

In this section we introduce an approximation algorithm for the connectivity maintenance problem. We first upper bound the performance of the algorithm and then derive the approximation ratio in the analysis section.

The algorithm we propose consists of three parts. In all cases, we start by checking if the network is already connected. If the value of M_n is smaller than or equal to the connectivity radius r, then we know that the initial configuration is already connected. Thus, the algorithm will return the initial positions.

If the configuration is not connected, we use a parameter characterized by L/r to decide which algorithm to use as a

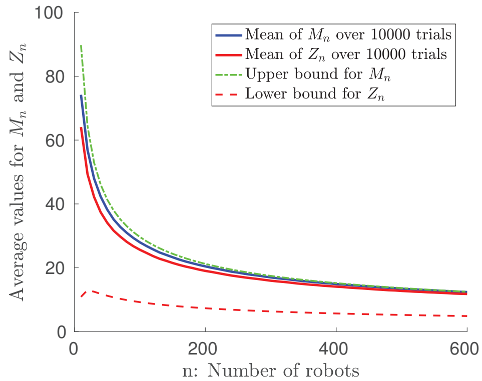


Figure 1: Average M_n and Z_n values for fixed $L = 100$ and varying $n = 10$ to 600.

subroutine to solve the problem. The value of L/r and its importance will be explained in the analysis section.

The main algorithm NETWORK-FORMATION is presented in Algorithm 1. It relies on the subroutine CLUSTER-CONNECT when the L/r ratio is smaller than $\sqrt{n}$. For any other scenario STAR-CONNECT is called. Intuitively, the algorithm chooses CLUSTER-CONNECT for densely distributed configurations since the robots are likely to coexist as clustered connected networks. For sparsely distributed positions STAR-CONNECT is preferred since gathering the robots in a central area is in fact a near-optimal strategy if the configuration is sparse enough. We next explain these subroutines.

Algorithm 1 NETWORK-FORMATION

Input: $\mathcal{X}$: robot positions, r: connectivity radius, L: environment radius
Output: $\mathcal{X}'$: final robot positions in a connected configuration

Compute smallest M_n s.t. $G(\mathcal{X}; M_n)$ is connected
if $r \geq M_n$ **then**
    return $\mathcal{X}' \leftarrow \mathcal{X}$
else if $L/r \leq \sqrt{n}$ **then**
    return $\mathcal{X}' \leftarrow$ CLUSTER-CONNECT$(\mathcal{X}, r)$
else
    return $\mathcal{X}' \leftarrow$ STAR-CONNECT$(\mathcal{X}, r)$

Star Connectivity Algorithm

The star connectivity algorithm rearranges the robots to form a connected communication network in a star topology. The algorithm starts by computing the Smallest Enclosing Circle (SEC) of a set of n robots in $\mathcal{X}$. Then it finds the closest robot u to the center c of that circle, and moves u to c.

After u is relocated at c, the algorithm moves each robot in $\mathcal{X} - \{u\}$ towards u, until its distance to u is at most r. The details are presented in Algorithm 2.

Let R be the radius of the SEC of the robots in the initial configuration. The robot with the shortest distance to c is moved by the algorithm to c. Here, the distance traveled by a robot can be at most R. Then, connecting all other robots to the center costs no more than $R - r$. Therefore, the solution produced by Algorithm 2 is always less than or equal to R.

Algorithm 2 STAR-CONNECT

Input: $\mathcal{X}$: robot positions, r: connectivity radius
Output: $\mathcal{X}'$: final robot positions in a connected configuration

$c \leftarrow$ The center of the SEC of $\mathcal{X}$
$x_i \leftarrow \arg\min_{x_j \in \mathcal{X}} d(x_j, c)$
$x'_i \leftarrow c$
for all $x_j \in \mathcal{X} - \{x_i\}$ **do**
    $\vec{d} \leftarrow \frac{\overrightarrow{c - x_j}}{||c - x_j||} \cdot \max\{0, ||c - x_j|| - r\}$
    $x'_j \leftarrow x_j + \vec{d}$
return $\mathcal{X}' \leftarrow \{x'_1, ..., x'_n\}$

Cluster Connectivity Algorithm

The cluster connectivity algorithm considers the connected components of $G(\mathcal{X}; r)$ as clusters, chooses an attracting cluster, and moves all other clusters to the attracting cluster in a sorted order with respect to their distances.

The algorithm initially partitions the robot positions in $G(\mathcal{X}; r)$ into a set of disjoint connected clusters $\Pi = \{P_1, ..., P_m\}$.

After computing the SEC of the starting locations, it compares the clusters according to what we call their attraction factors. We define the attraction factor of a cluster P_i to be $\rho_i := |P_i|/d(P_i, c)$, where $|P_i|$ denotes the number of robots in cluster P_i, and $d(P_i, c)$ is the distance between the closest robot in P_i to the center c of the SEC. The intuition behind this heuristic is to choose a cluster that is large and close to the center as possible to attract other clusters.

The cluster with the largest attraction factor is said to be the attracting cluster P^*. The robots in P^* stay put until all the robots are connected.

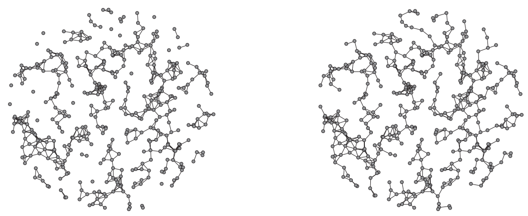
The algorithm sorts the non-attracting clusters with respect to their distances to P^* in a non-decreasing order. Then starting with the closest cluster, the algorithm connects the clusters to the attracting cluster in a sorted order. The procedure of connecting a cluster to another is summarized in Algorithm 3.

Algorithm 3 Connect

Input: $P_i, P_j \in \Pi$, r: connectivity radius
Output: P'_i: final positions of the robots in P_i

$u^*, v^* \leftarrow \arg\min_{u \in P_i, v \in P_j} d(u, v)$
$d = d(u^*, v^*)$, $\vec{d} = (d - r)(v^* - u^*)/||v^* - u^*||$
for all $u \in P_i$ **do**
    $u' \leftarrow u + \vec{d}$, $P'_i \leftarrow P'_i \cup \{u'\}$
return P'_i

Once a cluster connects to the attracting cluster, it does not move anymore and becomes a part of a growing con-



(a) Disconnected initial configuration (b) Connected final configuration

Figure 2: The solution of the CLUSTER-CONNECT subroutine for a densely distributed initially disconnected network of robots

nected component. When the last cluster is attracted to the large connected component, the algorithm terminates since the connectivity is maintained. The details of this subroutine are presented in Algorithm 4.

Algorithm 4 CLUSTER-CONNECT

Input: $\mathcal{X}$: robot positions, r: connectivity radius
Output: $\mathcal{X}'$: final robot positions in a connected configuration

```
c ← The center of the SEC of 𝒳
Partition 𝒳 into clusters Π = {P_1, ..., P_m}
for all P_i ∈ Π do
    Compute ρ_i = |P_i|/d(P_i, c)
P* ← arg max_{P_i∈Π} ρ_i
Sort P_i ∈ Π − {P*} w.r.t. d(P_i, P*)
𝒳' ← P*
for all P_i ∈ Π − 𝒳' in sorted order do
    P'_i ← Connect(P_i, 𝒳')
    𝒳' ← 𝒳' ∪ P'_i
return 𝒳'
```

The solution of CLUSTER-CONNECT to an instance of densely distributed network of 400 robots with $L/r = 10$ is presented in Figure 2.

Analysis of the algorithm

In the connectivity maintenance problem, we are given n robots deployed uniformly at random in a circular area of size πL^2, each of them with a connectivity radius r. The goal is to have a final configuration that is a connected communication network where no edge is larger than r.

We analyze the performance of Algorithm 1 in three different cases. The purpose is to understand the setup of the robots so as to design a strategy that works well for the initial configuration. If the configuration is almost connected for instance, we do not want our algorithm to treat the robots as far-flung from each other.

In Theorem 1, we present the main result of this paper.

Theorem 1. *There exists an $O(\sqrt{n})$-factor approximation algorithm for the connectivity maintenance problem with the maximum movement minimization objective function, where n is the number of robots.*

To characterize the initial configuration, we first compute the L/r ratio and compare it to two different functions of n. We use Result 1 as a threshold value for the L/r ratio to determine the probability of the connectivity of the nodes. We say if L/r is smaller than the threshold, the robots are distributed densely and the network is either connected or can be connected with a little perturbation. This constitutes the *Case 1* where the configuration is initially almost or already connected. We further divide the *Case 1* into two subcases *Case 1a* and *Case 1b*. *Case 1a* corresponds to configurations that are initially connected almost surely, and *Case 1b* characterizes dense configurations not initially connected.

We consider the robots are sparsely distributed if L/r is larger than a constant times n. This will be what we call the *Case 3*, and any value of L/r between the two thresholds will form the *Case 2*.

To develop an intuition for the separation of these cases we provide Figure 3. Note that the robots are distributed in a square-shaped area instead of a circle for illustrative purposes. The environment length L is the grid side length, and the grid cells have side length r in Figure 3.

Case 1: Dense configuration

In Case 1, we want to find a threshold for the value of L/r, such that when L/r is smaller than the threshold the initial configuration is either already connected or close to being connected.

Already connected configurations We use Result 1 as an upper bound for M_n. The initial configuration is guaranteed to be connected when the connectivity radius r is larger than M_n. Hence, when r is greater than the threshold the configuration is a connected network. Furthermore, if p is the probability that the inequality in Result 1 holds, then we can rearrange the relation for a connected configuration as follows.

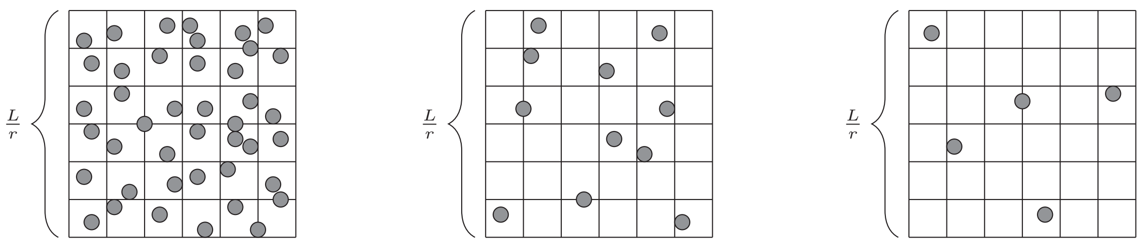
$$L\sqrt{\frac{\log n - \log\log \frac{1}{p}}{n\pi}} \leq r \tag{3}$$

Using (3) we conclude the threshold value for connectivity in already connected configurations: If $L/r \leq \sqrt{n\pi/(\log n - \log\log \frac{1}{p})}$ then we say the initial configuration is already connected almost always, and belongs to *Case 1a*.

When the configuration is initially connected, both the optimal and our algorithm's solutions are 0. In this case, the approximation is 1. We next analyze the *Case 1b*.

Critical configurations When the robots are distributed densely but not as dense to make the initial configuration already connected, there is a range of L/r values corresponding to what we consider as the critical configuration.

This case includes initially disconnected settings with an L/r ratio smaller than $\sqrt{n}$. For this intermediate range of L/r values Ganesan proves that with a high probability there is a giant connected component of $G(\mathcal{X}; r)$ which contains



(a) *Case 1* (b) *Case 2* (c) *Case 3*

Figure 3: Three cases depicting dense, moderate, and sparse configurations in respective order. In these illustrations L denotes the grid side length, and r is the side length of a grid cell. The L/r ratio is the number of grid cells in a column.

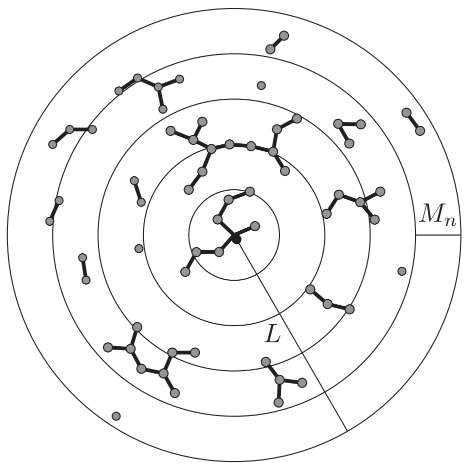


Figure 4: Robots densely distributed in a circular area, initial configuration is not connected

nearly all the robots (Ganesan 2013). We say a configuration is in *Case 1b* if L/r is less than $\sqrt{n}$, but the initial configuration is not connected.

In this scenario the robots are distributed densely, but not as dense to make the initial configuration already connected. An instance of this setting can be seen in Figure 4. We next prove the result for this case.

Lemma 1. *There is an $O(\sqrt{n})$-factor approximation algorithm for the connectivity maintenance problem when $L/r \leq \sqrt{n}$.*

Proof. Notice in Figure 4 that we expect to have a cluster at every ring since the robots are densely and uniformly distributed. We verify this as follows. A ring has at least one robot with a probability proportional to the ring's area. The innermost ring has area πM_n^2, and the i^{th} innermost ring has area $\pi i^2 M_n^2 - \pi(i-1)^2 M_n^2 = \pi(2i-1)M_n^2$. Let $k = L/M_n$ be the number of rings. Clearly, k cannot be greater than $\sqrt{n}$ since $L/r \leq \sqrt{n}$ and $M_n > r$. Then, the total area is $\pi k^2 M_n^2$, and the probability of having a robot in the i^{th} ring is $(2i-1)/k^2$. A ring has at least probability $1/k^2 \geq 1/n$ to contain a robot, and because we have n robots, we expect each ring to have a cluster.

Moreover, between an attracting and a non-attracting cluster there are at most $2L/M_n$ layers of clusters with high probability, and the distances between these cluster layers are not larger than M_n.

The attracting cluster is chosen to be near the center of the SEC of $\mathcal{X}$. Therefore, as clusters connect to the attracting cluster the inner layers move towards the center. A cluster at the i^{th} layer moves no more than $i(M_n - r)$ to connect to the attracting cluster. Since there are at most $2L/M_n$ layers of clusters between an attracting and a non-attracting cluster, the maximum distance traveled by a robot SOL is smaller than $\frac{2L}{M_n}(M_n - r)$.

The solution of the optimal strategy OPT can be lower bounded with $(M_n - r)/2$ since the connectivity cannot be maintained if no robot moves more than that.

M_n is not smaller than r, and the largest value L/r can get is $\sqrt{n}$. Then we can derive the approximation factor α_{1b} of *Case 1b* as follows.

$$\frac{SOL}{OPT} \leq \frac{2L(M_n - r)/M_n}{(M_n - r)/2} = \frac{4L}{M_n} < \frac{4L}{r} \leq 4\sqrt{n} = O(\sqrt{n}) \tag{4}$$

□

Suppose a dense configuration is initially connected with probability p. Then, with probability $1 - p$ the configuration is in *Case 1b*, and we have $O(\sqrt{n})$-factor approximation. With the remaining probability the solutions of the optimal strategy and our algorithm are 0. Thus, the approximation for *Case 1* is $\mathbb{E}[\alpha_1] = p \times 1 + (1-p) \times \alpha_{1b} = O(\sqrt{n})$.

Case 3: Sparse configuration

In the third case the robots are sparsely distributed over a large area. Intuitively, when the configuration is sparse the optimal strategy has to move the robots towards a center minimizing the maximum traveled distance.

The center of the SEC of a set of points $\mathcal{X}$ is the point $c \in \mathbb{R}^2$ minimizing $\max_{x_i \in \mathcal{X}} ||c - x_i||_2$. We will use the SEC of the initial configuration for proving the Lemma 2 presented below.

Lemma 2. *Suppose the smallest circle C enclosing all the robots has radius R. There is a $(\sqrt{3}+1)$-factor approximation algorithm when $R > nr$, where n is the number of robots, and r is the connectivity radius.*

Proof. Let ∂C denote the boundary of C, and suppose there are k robots on ∂C. By the definition of the SEC, k can vary as $2 \leq k \leq n$. Now consider two subcases: $k = 2$ and $k > 2$.

When $k = 2$, the line segment between the two robots u and v on ∂C defines the diameter of C, and the Euclidean distance between u and v, $d(u, v) = 2R$.

Suppose x' denotes the position of a robot x in the final configuration. In a solution to the connectivity maintenance problem, distance between a pair of robots can be at most $(n-1)r$. Therefore, the distance between u and v in the final configuration, $d(u', v') \leq (n-1)r$.

Since $r > 0$, the total movement traveled by u and v is at least $2R - (n-1)r \geq 2R - nr$. Hence, the maximum movement by a robot is at least $(2R - nr)/2 = R - nr/2$.

Next, consider the other subcase $k > 2$, that is, there are more than two robots on ∂C in the initial configuration.

Drager et al. showed that the furthest pairwise distance of a set of points cannot be smaller than $R\sqrt{3}$, which is the case when three of the robots on ∂C form an equilateral triangle (Drager, Lee, and Martin 2007). Using a similar argument to $k = 2$ case, we can say the maximum movement traveled by a robot is at least $R\sqrt{3}/2 - nr/2$.

Thus, independent of the number of robots on ∂C, k, the optimal solution OPT to the connectivity maintenance problem cannot be smaller than $R\sqrt{3}/2 - nr/2$. Noting the solution produced by our algorithm SOL is at most R, and $R > nr$, we calculate the approximation factor α_3 of the third case as follows.

$$\frac{SOL}{OPT} \leq \frac{R}{R\sqrt{3}/2 - nr/2} \leq \frac{2R}{R\sqrt{3} - R} = \sqrt{3} + 1 \quad (5)$$

Therefore, a solution produced by our algorithm is at most $\alpha_3 = \sqrt{3} + 1$ times worse than that of the optimal strategy. □

By Lemma 2, we have a constant factor approximation when $R/r > n$. Since we are looking for a threshold value for the L/r ratio we need a relation between L and R.

One trivial observation is R cannot be larger than L since the radius of the SEC of a set of points distributed in a circular area of radius L can be at most L. Furthermore, the average Euclidean distance between two uniformly distributed points in a circular area of size πL^2 is given as $128L/45\pi \approx 0.9L$ in (Burgstaller and Pillichshammer 2009). Therefore, the SEC of a configuration with $n \geq 2$ robots has a radius $R > 0.9L/2 = 0.45L$ with higher probability as n increases. Then if $0.45L$ is larger than nr, so is R. Thus, we say the configuration is sparse when $L/r > 2.2n$, and this bound constitutes the threshold value for *Case 3*.

Case 2: Moderate configuration

When the L/r ratio is larger than $\sqrt{n}$ but smaller compared to $2.2n$, we consider the initial configuration to be moderately distributed. Any placement that is neither dense nor sparse is said to be in *Case 2*. Figure 5 shows the separation of the cases with respect to the relation between L/r and n.

Figure 5: Classification of the cases based on the relation between L/r and n

We next present the result for the case when the robots are moderately distributed.

Lemma 3. *There is an $O(\sqrt{n/\log n})$-factor approximation for the connectivity maintenance problem when $\sqrt{n} < L/r \leq 2.2n$.*

Proof. To prove the lemma we need a lower bound for the maximum movement of the optimal strategy. We will use the largest nearest-neighbor distance Z_n in our argument.

Consider a robot u whose distance to its nearest-neighbor is Z_n initially. Let u' denote the final position of u. Next, suppose v' is a robot connected to u' in the final configuration, and v is its initial position. The distance between u and v in their starting locations is at least Z_n since the nearest-neighbor of u is Z_n units away. Then we can lower bound the maximum movement of the optimal solution OPT as follows.

$$OPT \geq \max\{d(u, u'), d(v, v')\} \geq (Z_n - r)/2 \quad (6)$$

Since we have Z_n for the bound of OPT, a lower bound for the value of Z_n in uniform distributions suffices.

We use Result 2 for lower bounding Z_n. For values of $t \in \mathbb{R}$ smaller than -1.53 the expression $1 - \exp(-e^{-t})$ is greater than 0.99. Then, for a value of $t < -1.53$ the following inequality holds almost always for uniformly distributed points in a circular area of radius L.

$$Z_n \geq L\sqrt{\frac{\log n + t}{\pi n}} \quad (7)$$

We can then rewrite the lower bound for the optimal solution.

$$OPT \geq \frac{1}{2}\Big(L\sqrt{\frac{\log n + t}{\pi n}} - r\Big) = \frac{L}{2}\Big(\sqrt{\frac{\log n + t}{\pi n}} - r/L\Big) \quad (8)$$

Since we are in the *Case 2*, the ratio of L/r cannot be smaller than $\sqrt{n}$. Therefore, the lowest value OPT can get is when r/L is maximized at $1/\sqrt{n}$.

The maximum movement of our algorithm SOL is at most $R \leq L$. Then we have the following approximation factor α_2.

$$\frac{SOL}{OPT} \leq \frac{2}{\sqrt{\frac{\log n + t}{\pi n}} - \frac{1}{\sqrt{n}}} = \frac{2\sqrt{\pi n}}{\sqrt{\log n + t} - \sqrt{\pi}} = O(\sqrt{n/\log n}) \quad (9)$$

□

For all the values of L/r we have an approximation ratio $O(\sqrt{n})$. Thus, Theorem 1 follows.

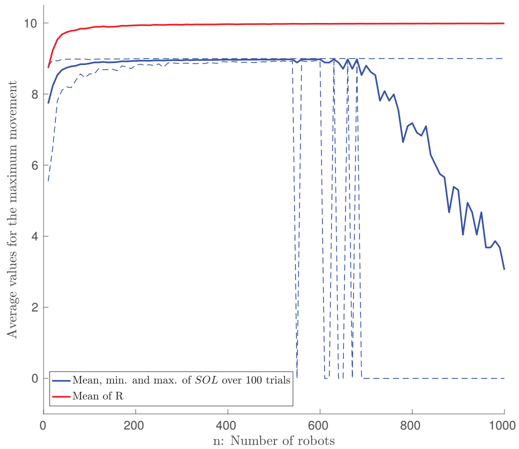


Figure 6: Average, maximum and minimum values for the maximum movement of STAR-CONNECT with $L = 10$, $r = 1$ and varying $n = 10$ to 1000.

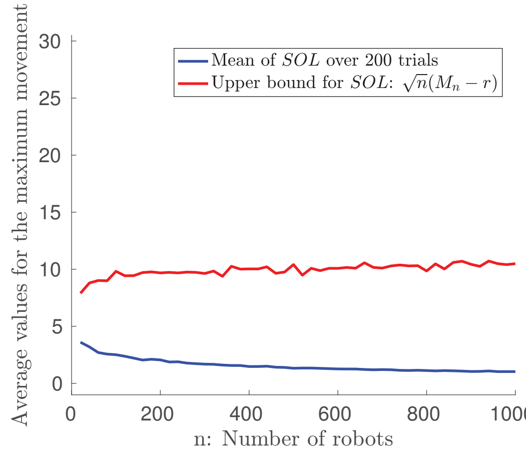


Figure 7: Average values for the maximum movement of CLUSTER-CONNECT with $L = 10$ and varying $n = 10$ to 1000.

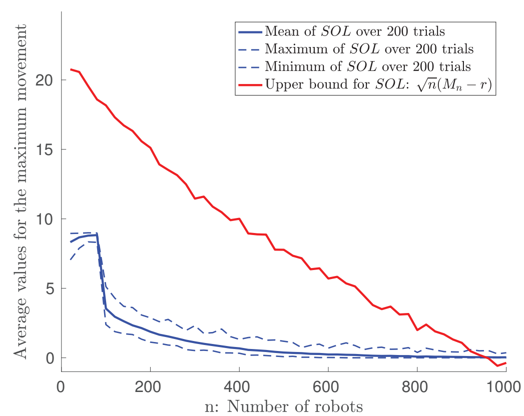


Figure 8: Average values for the maximum movement of NETWORK-FORMATION with $L = 10$, $r = 1$ and varying $n = 10$ to 1000.

Simulations

We have conducted a set of simulations in MATLAB to verify our analysis and the bounds we use from preliminary works. The robots are deployed according to the model. The program parameters are the number of robots n, environment radius L, and connectivity radius r. We distribute the robots uniformly at random and perform our algorithms to maintain connectivity.

We first show the individual results of the subroutines STAR-CONNECT and CLUSTER-CONNECT separately.

We used the STAR-CONNECT algorithm in a circular area with radius $L = 10$, and the connectivity radius was $r = 1$. Running the algorithm using various number of robots ranging between 10 and 1000, we obtained the result presented in Figure 6. From the figure we see that the maximum movement of a robot is no larger than R. Furthermore, when the L/r ratio is 10 the initial configuration is already connected for values of n starting from 600 as can be inferred from the analysis, and the likeliness of connectivity increases as n grows.

We then tested the CLUSTER-CONNECT algorithm in the critical configuration at different scales. For given values of n and L, we choose the connectivity radius uniformly at random from a range to make the configuration be in *Case 1b*, since this is the case with the bottleneck approximation. We used $L = 10$ and varying n up to 1000 robots. The result is presented in Figure 7.

We plot the average value of the maximum movements for each n, and upper bound it using the approximation ratio we derived. This result validates the analysis of the algorithm CLUSTER-CONNECT.

Finally, we present the results for the main algorithm NETWORK-FORMATION which uses CLUSTER-CONNECT and STAR-CONNECT as subroutines.

Using an environment length of 10 and connectivity radius 1 we obtain the result shown in Figure 8. For values of n smaller than 100 STAR-CONNECT is called, otherwise if the configuration is not already connected the subroutine CLUSTER-CONNECT is executed. In Figure 8 we also show the upper bound for the algorithm's solution using the approximation factor. The bound becomes negative when n is larger than 900 since the average value of M_n is below r, that is the initial configuration is already connected.

Conclusion

In this paper, we studied a novel connectivity formation problem where we are given initial locations of robots and a communication radius. The goal is to move the robots so as to form a connected network while minimizing the maximum movement. We focused on uniform random deployments and presented an $O(\sqrt{n})$-approximation algorithm for connecting n robots. The approximation ratio is significantly better than the $O(n)$ bound available for arbitrary deployments given by (Anari et al. 2016).

In our analysis, we consider three cases based on the density of the deployment. In two of these cases, the algorithm performs strictly better than $O(\sqrt{n})$. Therefore, there might be room for further improvements in the analysis.

Another avenue for research is to consider other initial distributions (e.g. Gaussian). Finally, a challenging but practically important version of the problem is the online version where the robots do not know each others initial locations.

Acknowledgements

This work was supported in part by NSF grant #1617718. We thank the anonymous reviewers for their valuable feedback and suggestions.

References

Anari, N.; Fazli, M.; Ghodsi, M.; and Safari, M. 2016. Euclidean movement minimization. *Journal of Combinatorial Optimization* 32(2):354–367.

Burgstaller, B., and Pillichshammer, F. 2009. The average distance between two points. *Bulletin of the Australian Mathematical Society* 80(3):353.

Demaine, E. D.; Hajiaghayi, M.; Mahini, H.; Sayedi-Roshkhar, A. S.; Oveisgharan, S.; and Zadimoghaddam, M. 2009. Minimizing movement. *ACM Transactions on Algorithms (TALG)* 5(3):1–31.

Drager, L. D.; Lee, J. M.; and Martin, C. F. 2007. On the geometry of the smallest circle enclosing a finite set of points. *Journal of the Franklin Institute* 344(7):929–940.

Ganesan, G. 2013. Size of the giant component in a random geometric graph. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 49, 1130–1140. Institut Henri Poincaré.

Michael, N.; Zavlanos, M. M.; Kumar, V.; and Pappas, G. J. 2009. Maintaining connectivity in mobile robot networks. In *Experimental Robotics*, 117–126. Springer.

Penrose, M. D. 1997. The longest edge of the random minimal spanning tree. *The Annals of Applied Probability* 340–361.

Poduri, S., and Sukhatme, G. S. 2007. Achieving connectivity through coalescence in mobile robot networks. In *Proceedings of the 1st International Conference on Robot Communication and Coordination*, 1–6. IEEE.

Steele, J. M., and Tierney, L. 1986. Boundary domination and the distribution of the largest nearest-neighbor link in higher dimensions. *Journal of Applied Probability* 23(2):524–528.

Zavlanos, M. M., and Pappas, G. J. 2007. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics* 23(4):812–816.

Zavlanos, M. M., and Pappas, G. J. 2008. Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics* 24(6):1416–1428.