Computer Networks 92 (2015) 116–133

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Optimization algorithms for transmission range and actor movement in wireless sensor and actor networks☆

Hyunbum Kim [a], Jorge A. Cobb [b,*]

[a] Department of Computer Science, University of North Carolina at Wilmington, Wilmington, NC 28403, United States
[b] Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, United States

ARTICLE INFO

*Article history:*
Received 6 January 2015
Revised 11 September 2015
Accepted 15 September 2015
Available online 9 October 2015

*Keywords:*
Transmission radius
Actor movement
Coverage
Wireless sensor and actor networks

ABSTRACT

Wireless sensor and actor networks are composed of static sensors and mobile actors. We assume actors have a random initial location in the two-dimensional sensing area. The objective is to move each actor to a location such that every sensor node is within a bounded number of hops from some actor. Because sensors have limited energy, the new actor locations are chosen to minimize the transmission range of the sensors. However, actors also have a limited (although larger) power supply, and their movement depletes their resources. It follows that by carefully choosing the new actor locations, the total actor movement can be minimized. In this paper, we introduce the problem of simultaneously minimizing the required transmission range and amount of actor movement. To find a solution, we formulate the problem using an ILP framework. For the ILP solution to be feasible, we introduce a finite set of possible actor locations such that an optimal solution is guaranteed to be found within this set. We also present a heuristic for this problem. As a preliminary step, we study minimizing the transmission range necessary for multi-hop communication. Various heuristics for this smaller problem are proposed and their results are compared by simulation. The best of these heuristics is then enhanced to incorporate minimizing the movement of actors, and its performance is compared to the optimal ILP solution.



## 1. Introduction

Wireless sensor and actor networks (WSANs) consist of two groups of devices. The first is a collection of sensor nodes. Sensors typically have a fixed location, and their resources are limited, in particular, their battery life. The second is a collection of actor nodes. These have significantly more resources than the sensor nodes, and are often mobile. When a sensor node gathers data about some event, it sends the sensing information to an actor. Actors make decisions for various issues and execute necessary actions based on the received information from sensor nodes and from other actors [4]. WSANs can be used in numerous applications, such as battlefield surveillance, urban search and rescue, environmental monitoring, etc.

The effectiveness and performance of the network depend heavily on the location of the actor nodes. For example, a proper placement of actors can improve network lifetime by reducing the transmission range required for sensor nodes, which in turn increases the lifetime of their batteries. Similarly, proper placement of actors can reduce the number of hops traversed between a sensor and its closest actor, which in turn affects network delay and also affects network lifetime. Thus, given the fixed locations of a set of

☆ A part of this paper has appeared in *Proceedings of IEEE Conference on Local Computer Networks (LCN 2011, 2012)* [1,2] and *Proceedings of IEEE International Conference on Computing, Networking and Communications (ICNC 2012)* [3].

* Corresponding author. Tel.: +14695634331.
*E-mail addresses:* kimh@uncw.edu, hyunbumkim@gmail.com (H. Kim), cobb@utdallas.edu (J.A. Cobb).

http://dx.doi.org/10.1016/j.comnet.2015.09.019

sensor nodes, and given the initial locations of a set of mobile actor nodes, we focus on choosing new actor locations that satisfy certain performance requirements, as discussed further below.

We study the problem of finding the smallest transmission radius, $r_{min}$, such that each sensor node can communicate with at least one actor within a maximum of $d$ hops, where $d$ is a parameter to our problem. Therefore, if there are $k$ actors, $k$ locations must be determined such that all sensor nodes can reach one of these locations within $d$ sensor hops.

Another factor to consider is that WSANs powerful advantage is the mobility of its actors. This mobility can be exploited to perform various goals such as maximizing network lifetime and minimizing data latency. Although the actors are equipped with more powerful resources when compared to sensor nodes, we assume their energy is nonetheless limited, such as from a battery or an on-board fuel supply. We assume the movement of an actor consumes significantly more energy than computation and the collection of data from sensors [11]. Because actors cannot be easily recharged after their deployment, minimizing their movement is a critical issue in WSANs [12,23–25].

The above motivates us to enhance the problem further by simultaneously minimizing the required transmission range and the amount of actor movement. Given are a set of $n$ fixed sensor nodes and a set of $k$ mobile actor nodes, both of which are randomly located in an open field. We seek the joint optimization of two values. The first one is minimizing the transmission radius such that each sensor can communicate with at least one actor within $d$ hops. The second one is minimizing the total actor movement from the initial locations of the actors to their newly chosen locations.

Note that there is a trade-off relationship between the energy consumption from the relocation of actors and the energy consumption from the long-distance communication of sensors. That is, minimizing sensor transmission range and minimizing actor movement are conflicting goals. E.g., actor movement drops to zero by simply extending the transmission range of sensors to reach the current location of the actors. However, such a large extension of transmission range will hasten the depletion of energy at the sensors, causing them to fail. On the other hand, the required sensor transmission range can be minimized if actors can be moved to any point on the plane without restrictions. This however imposes a large energy drain on the actors.

Given this conflict, it must be resolved in favor of one metric or the other. We choose to give higher priority to minimizing the sensor transmission range. From those solutions with the smallest transmission range, we choose the one that minimizes actor movement.

Due to the high complexity of the problem, we propose various heuristics and compare them via simulations. For small instances of the problem, we present an optimal ILP (Integer Linear Programming) solution. For the ILP solution to be feasible, there has to be a finite set of possible actor locations from which the optimal solution is guaranteed to be found. In the literature [9,13], sets of possible actor locations have been presented that guarantee that an ILP solution will find an optimal transmission radius. However, they do not guarantee an optimal movement of actors. Here, we present a novel alternative set of possible locations that guarantee an optimal solution is found for both transmission radius and actor movement.

There are two alternative approaches that we could apply to solve the joint minimization of transmission range and actor movement. The first is the *double-step approach*, in which we first minimize the transmission range, and then, after choosing the actor locations, we minimize the actor movement by pairing actors with their new locations. The second is the *single-step approach*, in which each actor location and its corresponding actor are chosen together. We show that the single-step approach outperforms the double-step approach in ILP solutions and in heuristic solutions.

This paper is organized as follows. Related work is presented in Section 2. A formal definition of the problem addressed is given in Section 3. Then, Section 4 presents how a finite set of possible actor locations can be derived. A finite set of possible radii is derived in Section 5. The ILP formulation for our problem is presented in Section 6. Section 7 presents heuristics for sensor coverage, while Section 8 presents our joint optimization heuristics. Section 9 provides the simulation results. Finally, we present concluding remarks and suggest future research directions in Section 10.

## 2. Related work

In WSANs, many important studies have been done. In [4], the authors introduced the concept of WSANs applications and also described various open research topics which should be considered in WSANs.

In [5,6], the authors formally defined a unifying framework for coordination and communication problems in WSANs. A sensor-actor coordination model is proposed using event-driven partitioning paradigm. In the model, a new notion, reliability, is introduced to consider the timely delivery of data packets and to minimize energy consumption. Also, a new model of actor-actor coordination is defined and formulated as a task assignment optimization problem. Because the timely execution of correct actions in WSANs is a critical issue, authors also proposed distributed algorithms for both sensor-actor coordination and actor-actor coordination as well as centralized optimal solutions.

Furthermore, coordination and communication problems in WSANs movable actors have been studied in [7,8]. Authors proposed a proactive location management scheme in order to handle the mobility of actors with minimal energy consumption of the sensor nodes. To do so, actors broadcast updates which limit their scope based on Voronoi diagrams. The location management scheme enables geographical routing for sensor-actor communication. Also, the authors proposed a new model for actor-actor coordination, which assigns works to mobile actors and coordinates their mobilities based on the characteristics of the events.

On the other hand, there have been several works in the area of minimizing actor movements in WSANs. Most works address only the selection of locations where actors (or other significant nodes) are to be placed. Some of these works are the following.

In [9], authors defined $k$-sink placement problem whose objective is to minimize the maximum hop-distance between sensor node and its nearest sink. Also, the placement of actors for load balancing is addressed. Their objective is to find clusters such that the size of each cluster is bounded and the number of hops from each sensor to each cluster is also bounded [18]. In [10], in order to reduce data latency, authors proposed two approaches based on genetic algorithms, which show how to choose locations for multiple sinks such that the average hop and Euclidean distance between all sensor nodes and their nearest sink are minimized. In [19], they presented an actor placement scheme that provides both maximizing coverage of area and minimizing data gathering latency. In [20,21], authors defined the problem of computing the optimal trajectories of multiple mobile mules to minimize data collection latency and then proposed constant factor approximation algorithms. Also, in [22], authors investigated how to collect data from fixed sensors using multiple robotic vehicles under different circumstance such as different mobilization conditions.

Once locations are chosen for actors, pairing actors with their new location poses an optimization problem if the distance traveled by actors is to be minimized. In [23], the authors proposed a pairing of actors and cluster heads using a heuristic based on matching theory. Its goal is to minimize total actor movement or total matching distance between actors and cluster head. On the other hand, [24] studied the different goal of restoring connectivity among actors by relocating actors with minimal movement in case of actor failure. However, [25] pointed out the algorithms in [24] do not work in all scenarios and showed counterexamples. In addition, they described a general actor relocation problem and proposed algorithm which result in an optimal movement of actors.

Our earlier works [2,3] and our results below distinguish themselves from the above works in two ways. First, the transmission radius, instead of a constant, is a parameter to be optimized. Second, we considered a joint optimization of the transmission radius and the actor movement.

## 3. Problem statement

In this section, we define our problem formally as follows.

We are given a set $S$ with the locations of $n$ sensors that are randomly deployed on a two-dimensional plane. Also given is an actor sequence $A$, where $A = a_1, \ldots, a_k$, containing the location of $k$ actors randomly deployed on the plane. The sensor and actor transmission radius are assumed to be homogeneous.

An *actor relocation* is a pair of actor location sequences, $(A, B)$, where $|A| = |B| = k$. The actor in location $a_i$, $1 \le i \le k$, is expected to move to location $b_i$. Let $\textit{total-move}(A, B)$ be the sum of the distances that actors must traverse to move to their new locations, i.e.,

$$\textit{total-move}(A, B) = \sum_{1 \le i \le k} \textit{distance}(a_i, b_i)$$

In later sections, with some mild abuse of notation and depending on context, actor sequences $A$ and $B$ will be sometimes viewed as a set rather than a sequence.

An upper bound $d$ is given on the allowed hop count from any sensor node to its closest actor once actors have been relocated. For a specific transmission radius $r$, predicate $\textit{cover}(S, B, r, d)$ is true iff every sensor in $S$ is able to reach an actor in $B$ in at most $d$ transmission hops.

We say that an actor relocation $(A, B)$ *minimizes* actor *movement* (MM) iff $\textit{cover}(S, B, r, d)$ is true and, for any other actor location sequence $B'$,

$$\textit{total-move}(A, B) \le \textit{total-move}(A, B')$$

Let $\textit{min-radius}(S, d)$ be the smallest value of $r$ such that $\textit{cover}(S, B, r, d)$ is true for some $B$. We often refer to this value as $r_{min}$ when $S$ and $d$ are understood from context.

Finally, we say that an actor relocation $(A, B)$ *minimizes* the *range* and actor *movement simultaneously* (MRaMS) iff,

$$\textit{cover}(S, B, r_{min}, d)$$

and furthermore, for any other actor location sequence $B'$,

$$\neg\textit{cover}(S, B', r_{min}, d) \vee \textit{total-move}(A, B) \le \textit{total-move}(A, B')$$

Note that for the specific case of $d = 1$, and considering only finding the minimum radius (disregarding actor movement), this is equivalent to the Euclidean $p$-center problem[1] in Operations Research [14–16].

Also note that there always exists a solution to the MRaMS problem provided each of $k$, $n$, and $d$ are greater than zero. This is because any point on the plane can be reached from any sensor node using at most $d$ hops provided the transmission range $r$ of the sensors is chosen large enough. In later sections, we show that $r_{min}$ must be contained within a finite set of values. We also show that, for a given transmission range $r$, the positions that minimize actor movement are also contained within a finite set of values. Thus, $r_{min}$ plus the desired actor relocation can be obtained through an exhaustive search.

Due to the complexity of MRaMS, in later sections we encode the problem using an ILP that obtains an optimal solution, and we also develop a heuristic that is evaluated by comparing it against the optimal solution of the ILP. This is a centralized method, i.e., given the location of sensors and initial actor locations as input, the heuristic obtains the transmission radius and the new actor locations. In practice, however, this information has to be gathered at a central location, followed by running the heuristic, and then redistributing the results to all nodes. How this can be done depends on the specific network being deployed, and is beyond the scope of this paper. Nonetheless, one example could be the following.

Since the location of each sensor is static, a list of sensor and their locations could be preloaded in each actor's memory. Periodically, communication between actors can be established using sensors as intermediate nodes, by using a method such as the one described in [24]. Actors can share their locations with each other, and this information can be gathered at an actor elected as leader (or a base station if available) who in turn will run the heuristic. The new transmission range and location can then be flooded throughout the network.

[1] Given a set of $n$ demand points $D = \{d_1, d_2, \ldots, d_n\}$ on the plane, the Euclidean $p$-center problem is to find a set of $P$ supply points $S = \{s_1, s_2, s_2, \ldots, s_P\}$ of new facilities such that the furthest distance between each demand point and its closest new facility is minimized. That is, $max$ [ $min$ $\{EucDist(d_i, s_j)\}$ ] is minimized (where $1 \le i \le n$, $1 \le j \le P$ and $EucDist(d_i, s_j)$ is the Euclidean distance between $d_i$ and $s_j$). The Euclidean $p$-center problem is a well-known NP-hard problem in Operations Research.

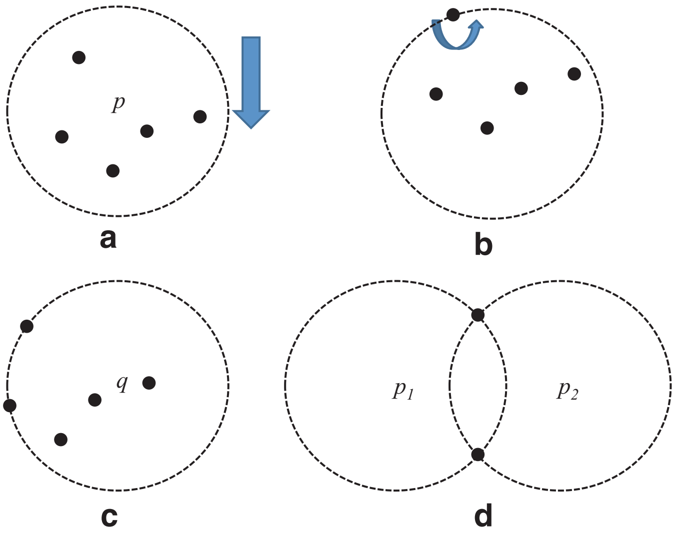



**Fig. 1.** Canonical possible actor locations.

## 4. Finite set of possible actor locations

Solving MRaMS involves finding an actor relocation that satisfies several criteria, and thus, yields a significantly complex problem. Let us address this problem incrementally, and begin with a smaller problem; we temporarily put aside the mobility of actors and the choice of multiple transmission ranges, and focus on the cover problem. That is, given the set $S$ of sensor nodes, the number $k$ of actors, a transmission radius $r$, and a hop-count bound $d$, we must find a set[2] of actor locations $B$ such that $cover(S, B, r, d)$ is true. This problem is an NP-complete problem in and of itself [13,15], so the best known method to solve it requires exponential time.

One method to solve this is to examine all sets consisting of $k$ points on the plane, and test if placing actors on these points allows all sensors to be reached in $d$ hops. The problem with this method is that the number of points on the plane is infinite. However, it was shown in [9,14,17] that there does exist a finite set of points $P$ such that, if there exists a set $\mathcal{B}$ satisfying $cover(S, \mathcal{B}, r, d)$, then there must also exist a subet $B$ of $P$ satisfying $cover(S, B, r, d)$. We refer to $P$ as a set of *possible actor locations*, and is obtained as follows.

Assume a point $p$ is contained in the solution $\mathcal{B}$. Using a transmission range of $r$, point $p$ will be able to reach some sensors, as shown in Fig. 1(a), where point $p$ is at the center of a dashed circle of radius $r$, and the sensors it can reach in one hop are denoted by filled circles. Assume that no sensor is at the border of the circle (if there is, we skip the next step). Next, move the circle down until a sensor reaches its border. The end result is shown in Fig. 1(b). Assume that now we have only one sensor at the border (if two or more we skip the next step). Finally, rotate the circle using the border sensor as an axis, until another sensor reaches the border. The end result is shown in Figure 1(c). Let $q$ be the center of this new found circle.

We thus have that $q$ can reach the same (or more) sensors as $p$ in one hop. Certainly, the sensors that can reach $p$ in $d$ hops can also reach $q$ in $d$ hops. Thus, replacing $p$ by $q$ in $\mathcal{B}$ maintains $cover(S, \mathcal{B}, r, d)$ true. What makes $q$ special is that it has two or more sensors at its border. Thus, our set $P(S, r)$ of possible actor locations are the centers of all circles of radius $r$ with two or more sensors at their borders.

We next examine the size of $P(S, r)$. Consider Fig. 1(d), where an arbitrary pair of sensors are shown (with a distance from each other of at most $2r$). From geometry, there can by only two circles of radius $r$ (whose centers are denoted by $p_1$ and $p_2$ in the figure) that have these two sensors on their border. Thus, $|P(S, r)| = 2 \cdot \binom{n}{2} \in O(n^2)$.

Based on the above observations, the canonical set $P(S, r)$ of possible actor locations is created formally in Algorithm 1.

**Algorithm 1** computing the set of possible actor locations $P(S, r)$.

Inputs: $S$, $r$, Output: $P(S, r)$

1: $P(S, r) \leftarrow \emptyset$
2: For each pair of sensor node $u$ and $v$ such that $EucDist(u, v) < 2 \cdot r$, let $p_1$ and $p_2$ be the centers of two circles whose borders pass both $u$ and $v$. Set $P(S, r) \leftarrow P(S, r) \cup \{p_1, p_2\}$
3: Return $P(S, r)$

Thus, finding a set $B$ that satisfies $cover(S, B, r, d)$ can be found by examining all subsets of $P(S, r)$ of cardinality $k$, and check if all sensors can reach at least one of these points within $d$ hops[3].

[2] $B$ is viewed here as a set rather than a sequence since for the moment it is not being paired with elements from the sequence $A$ of initial actor locations.

[3] For completeness, $P(S, r)$ must also include the location of all sensor nodes that are located more than $2r$ distance units away from any other sensor node.

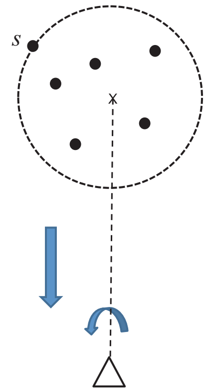



**Fig. 2.** Improved possible actor locations.

### 4.1. *Optimal actor locations to minimize movement*

Although an exhaustive search of the possible actor locations $P(S, r)$ ensures that a solution to $cover(S, B, r, d)$ is found, these locations were obtained with no regard for the original locations $A$ of the actors. Thus, they do not guarantee a minimum amount of actor movement. Below, we propose an alternative new set of possible actor locations, $P_A(S, A, r)$, that, through exhaustive search, can be used to find an actor sequence $B$ such that $cover(S, B, r, d)$ is true, and furthermore, total actor movement is minimized.

Consider Fig. 2. It consists of an actor (the small triangle) in its original location, and the dashed circle corresponds to the area of radius $r$ that the actor will cover after it has moved to a new location. The filled circles represent sensor nodes that will be associated with this actor once it reaches the new location. We will consider various relationships between an actor's previous and new location, and in doing so, we will eliminate many locations that result in non-optimal movement. The locations that remain will be finite, and thus can be searched exhaustively to obtain the optimal actor sequence $B$.

In Fig. 2, it is obvious that the new location is not optimal, for the following reason. If the actor turns slightly counterclockwise before beginning its movement forward, it is able to cover the same set of sensor nodes by traveling a slightly shorter distance. To visualize this, assume the path of the actor (the dashed line) and its coverage area (dashed circle) is a pendulum, whose fulcrum is at the current location of the actor. If the pendulum turns clockwise, then, sensor $s$, who is on the left border, will be outside of the actor range. However, if the pendulum turns slightly counterclockwise, then all sensors remain in the coverage area. In particular, $s$ is now farther inside the coverage area, rather than at its border. The length of the arm of the pendulum can then be made shorter until some node touches the border of the coverage area (most likely $s$). Since the arm of the pendulum is shorter, the actor has to travel a shorter distance than before.

Any optimal actor location cannot be improved by performing the above steps (rotation followed by shortening the distance). We perform a case analysis to eliminate locations that cannot yield an optimal movement. The case analysis is based on the number of sensors that are on the border of the possible new location.

#### 4.1.1. *One sensor at the border*

Consider a possible actor location, shown in Fig. 3(a) as a small filled square, where there is a single sensor, denoted by a small filled circle, in the periphery of the unit disk covered by the location. We would like to determine if this location should be added to our set $P_A(S, A, r)$ of possible actor locations.

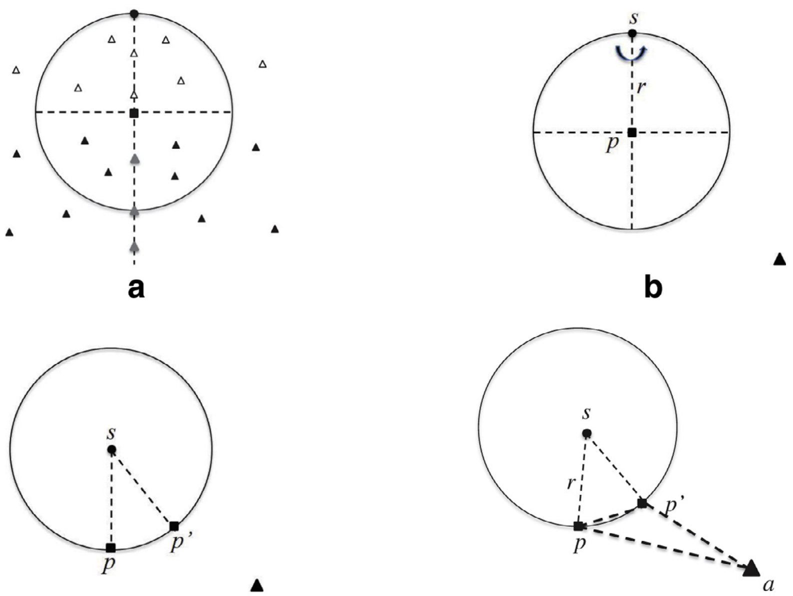



**Fig. 3.** Single sensor at the border.

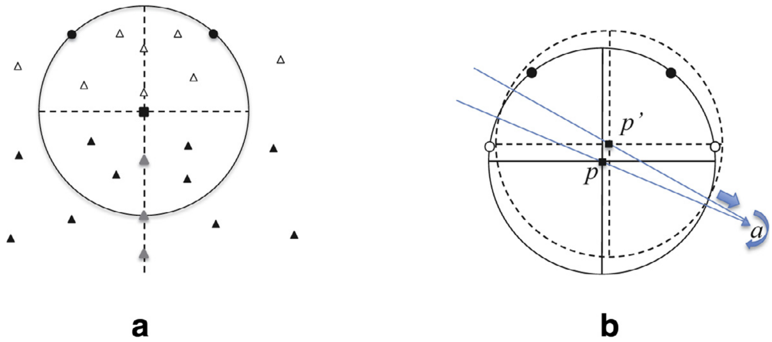



**Fig. 4.** Two sensors at the border.

Let us do a case analysis of where actors could be located with respect to this possible new location. The initial actor locations are denoted by small triangles.

- Actors above the horizontal line of the unit disk are drawn unfilled.
- Actors below the horizontal line of the unit disk are drawn filled.
- Actors below the horizontal line and along the vertical line of the unit disk are shown in gray.

For the actors above the horizontal line, if we move the possible location (the center of the circle) slightly upwards (towards the sensor), the distance from the actor to the possible location reduces, and hence, this is not an optimal location. This is true for actors located inside or outside the circle.

For the actors along the vertical line, there is no manner in which we can move the unit circle closer to them without losing coverage of the top sensor; thus, the center of the circle has to be considered as a possible optimal location. This adds a maximum of $n \cdot k$ points to $P_A(S, A, r)$.

For actors below the horizontal line, the location is not optimal, but showing this is a little more complicated. The steps below work for both actors inside the circle and actors outside the circle.

1. Consider rotating the unit disk around the sensor node as in Fig. 3(b), even slightly, making sure that other sensors inside the disk are not caused to be outside the disk. The rotation is in the direction of the actor, in this case, counter-clockwise.
2. Let the radius be $r$. Consider the circle of radius $r$ formed by the line $\overline{sp}$ (sensor $s$, possible position $p$) as the original disk rotates around $s$, as shown in Fig. 3(c). This new circle has $s$ as its center. The old and new centers (of the original unit disk) are $p$ and $p'$. The rotation causes actor $a$ to become closer to the center, i.e., line segment $\overline{ap}$ is longer than segment $\overline{ap'}$. This indicates that position $p$ is not optimal.

*4.1.2. Two sensors at the border*

Assume now that we have two sensors in the periphery of our unit disk centered at a possible actor location. Note that the two sensors have to be at most $2 \cdot r$ apart from each other. If the two sensors are exactly $2 \cdot r$ apart, then from the discussion below, it easily follows that we can not improve upon the distance to the actor, so this is considered a possible optimal location.

Let's consider when the sensors are less than $2 \cdot r$ apart as Fig. 4(a). In Fig. 4(a), set the axis so the two sensors are horizontal. In Fig. 4(a), actors above the horizontal (drawn unfilled) will improve their distance to the possible location (center of disk) if the disk moves up slightly; hence, the location is not optimal. The actors along the central line (drawn in gray) can not improve their distance if the disk is moved, so for them the center of this circle is a possible optimal location. For the actors below the horizontal (drawn filled), the location will be optimal for some and not optimal for others, as we will show below using Fig. 4(b).

- Consider the original disk centered at the possible position $p$ in Fig. 4(b). Think of the current actor location $a$ as the fulcrum, and consider the line connecting $a$ through $p$. We will rotate around $a$, which results in the new disk drawn with dashed lines. After rotation, the disk will be moved in the direction of $a$ (indicated by the arrow).
- Note that if the pair of sensors that defined the disk are above the line $\overline{ap}$ (drawn filled), then they can still be covered after the rotation and translation, and hence, $p$ does not define a possible optimal location.
- However, if one of the pair of sensors that defined the disk falls below the line $\overline{ap}$ (these sensors are drawn unfilled), then they cannot be covered after the rotation, in particular, the left point will be left uncovered, and hence, for this pair of sensor nodes and this initial actor location $a$, $p$ is a possible optimal location. This adds a maximum of $n^2 \cdot k$ points to $P_A(S, A, r)$.

*4.1.3. Three or More Sensors at the Border*

Finally, assume there are three or more sensors at the border of the coverage area. Three or more points on the plane define a single circle. The circle must be of radius $r$, i.e., the coverage area of the actor. It is unlikely that many triples of sensor nodes will precisely define a circle of radius $r$. Hence, rather than trying to eliminate them from our set $P_A(S, A, r)$ of possible optimal locations by using a complex case analysis, we simply add them to our set. This is because they are so few in practice that they will not significantly influence the time required to find an optimal solution.

From the above sections, $|P_A(S, A, r)| \in O(k \cdot n^2)$, which is greater than $|P(S, r)| \in O(n^2)$. Nonetheless, this increase in

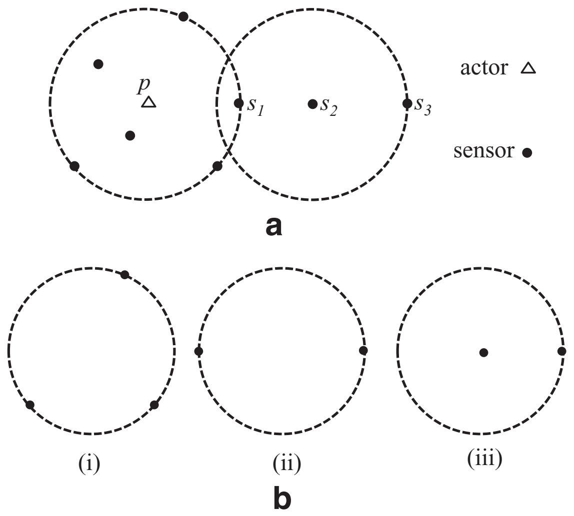



**Fig. 5.** Finite radii locations.

size comes with a significant reduction in actor movement, as shown in Section 6.

## 5. Finite set of radii

In MRaMS, we are interesting in finding an actor sequence $B$ such that $cover(S, B, r_{min}, d)$ is true, and $total\text{-}move(A, B)$ is minimized. If we knew $r_{min}$ in advance, we can do an exhaustive search of $P_A(S, r)$ for the $k$ points of $B$ that provide coverage and minimum total movement.

For the moment, assume that there is a finite set $R(S, d)$ from which $r_{min}$ is guaranteed to be an element. We could then solve MRaMS by applying the above method to all values $r \in R(S, d)$, and keep track of the best values of $B$ and $r$.

For efficiency, note that not all values in $R(S, d)$ need to be examined. This is because if the sensors can be covered with a radius of $r$, then they can also be covered with a radius $r'$, where $r' > r$. Thus, we can perform binary search over the radii in $R(S, d)$ to obtain $r_{min}$[4].

To argue that the set of possible radii $R(S, d)$ is finite, consider an actor placement $\mathcal{B}$ such that $cover(S, \mathcal{B}, r_{min}, d)$ is true. Because there is no smaller radii that can cover the sensors in $d$ hops (by definition of $r_{min}$), then reducing the radius below $r_{min}$ must prevent some sensors to reach their actor in $d$ hops. Consider Fig. 5, where the actor that cannot be reached is $p$ and several sensors with $d = 3$ are drawn. The reason the transmission radius cannot be reduced must be one of the following:

(i) Reducing the transmission radius would prevent two sensors from being one hop away from each other. For example, in Fig. 5(a), sensor $s_3$ is unable to reach sensor $s_2$ if the transmission radius is reduced.

(ii) Reducing the transmission radius would remove one or more sensors from the one-hop neighborhood of $p$. For example, in Fig. 5(a), sensor $s_1$ would not be able to reach $p$ in one hop if the transmission radius is reduced. This in turn would prevent $s_3$ from reaching $p$ in three hops.

Consider case (i) above. For every pair of sensor nodes $s_i$ and $s_j$, the distance between them is a possible value for $r_{min}$, and thus this distance is added to $R(S, d)$. This case is not considered in [13,15] because the Euclidean $p$-center problem is not multi-hop.

Consider case (ii) above. Let $S_p$ be the one-hop sensors of point $p$. Consider all points $p_{i,1}, \ldots, p_{i,j}$ in $\mathcal{B}$ such that these actor locations have one or more sensors at the border of their transmission radius $r_{min}$. This set cannot be empty since otherwise we can reduce the transmission range of each actor without changing its one-hop neighborhood, which violates the definition of $r_{min}$. Similarly, at least one node $p_{i,k}$, $1 \le k \le j$, will have a one hop neighborhood $S_{p_{i,k}}$ such that the smallest possible circle that can encompass all the nodes in $S_{p_{i,k}}$ has radius $r_{min}$. Again, if this is not the case then the definition of $r_{min}$ is violated.

Because $S_{p_{i,k}}$ is unknown to us, we must consider all possible subsets of $S$. For each of these subsets, the smallest circle that encompasses all the sensors in the subset defines a possible radius, and thus is added to $R(S, d)$.

Thus, consider any arbitrary subset $S'$ of $S$. The smallest circle that encompasses $S'$ must contain either three (or more) sensors in its periphery (e.g. see Fig. 5(b)(i)), or two sensors at exactly opposite points in the circle (e.g. see Fig. 5(b)(ii))[13,15]. There could also be isolated sensors that require an actor next to them. There are $O(n^3)$ sensor triples, $O(n^2)$ sensor pairs, and $O(n)$ single nodes, and hence, there are $O(n^3)$ possible radii from which $r_{min}$ can be chosen.

## 6. Solving MRaMS via ILP

In this section, we present an ILP formulation of the MRaMS problem to quantify the effect of choosing the new actor locations from the above set $P_A(S, A, r)$ of possible optimal positions (with respect to movement) vs. choosing from the canonical set $P(S, r)$ given in [9,14,17]. It will also serve as a reference for the MRaMS heuristics that we present in Section 8.

The ILP below is given a set of sensors $S$, a set of actors $A$, a radius $r$, a hop count $d$, and a set of potential locations $P$. The ILP will search for an actor sequence $B$ such that $cover(S, B, r, d)$ is true (provided such a sequence exists for $r$), and also such that $total\text{-}move(A, B)$ is minimized.

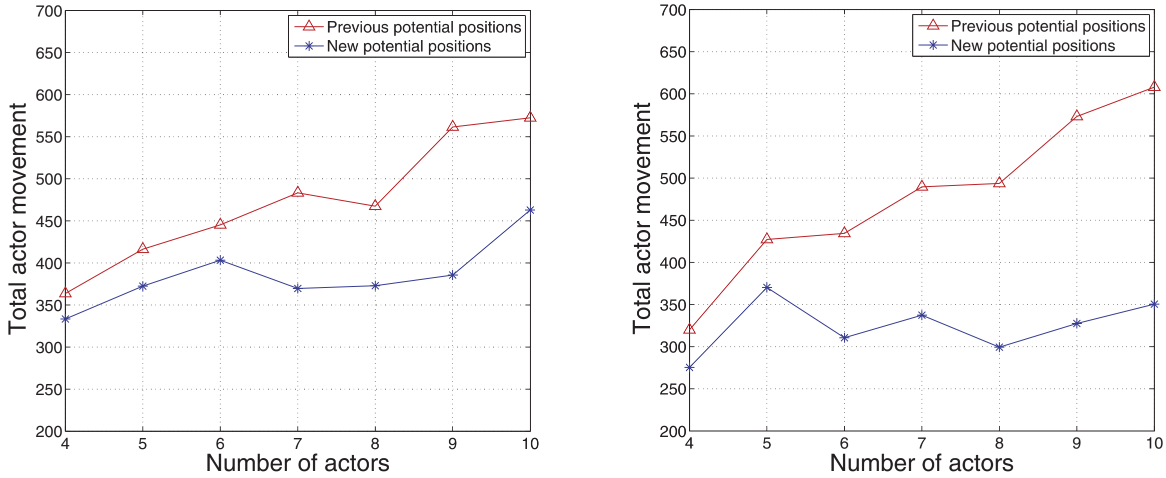
Note that a binary search over the set of possible radii $R(S, d)$ is still necessary, since the ILP will only determine if a specific radius $r$ is sufficient to cover all sensors, but does not directly obtain $r_{min}$. Note also that both choices for $P$, i.e., $P_A(S, A, r)$ vs. $P(S, r)$, will yield the same minimum radius $r_{min}$. However, $P_A(S, A, r)$ guarantees the optimal actor movement.

### 6.1. ILP formulation

The notations used in the ILP formulation as follows.

$P$: set of possible new locations for the actors.
$n$: total number of sensor nodes.
$m$: total number of possible new locations for actors ($m = |P|$).

[4] A similar strategy of binary search on the list of possible radii is also used in solving the $p$-center problem [14–16].




(a) hop bound $d = 1$ (b) hop bound $d = 2$

**Fig. 6.** Comparison of total actor movement from ILP using 50 sensor nodes.

$k$: total number of deployed actors.
$i$: index for a sensor node ($1 \le i \le n$).
$j$: index for a deployed actor ($1 \le j \le k$).
$p$: index for a location in set $P$ ($1 \le p \le m$).
$\lambda_{j,p}$: distance from location $p$ to initial location of actor $j$.
*sensor-hops*$(i, d)$: subset of $P$ reachable from sensor $i$ in $d$ hops.

Also, we define the following integer variables.

$$Y_{j,p} = \begin{cases} 1, & \text{if an actor } j \text{ moves to location } p \\ 0, & \text{otherwise.} \end{cases}$$

$$Z_p = \begin{cases} 1, & \text{if location } p \text{ is chosen as} \\ & \text{one of the } k \text{ actor locations} \\ 0, & \text{otherwise.} \end{cases}$$

Our objective function is to minimize the sum of the distances between the initial and the final locations for the actors. Thus, the objective function is as follows.

$$\textbf{Minimize} \sum_{j=1}^{k} \sum_{p=1}^{m} \lambda_{j,p} \cdot Y_{j,p} \tag{1}$$

Subject to:

$$\sum_{p=1}^{m} Y_{j,p} \le 1, (\forall j) \tag{2}$$

$$Y_{j,p} \le Z_p, (\forall j, \forall p) \tag{3}$$

$$\sum_{p=1}^{m} Z_p \le k \tag{4}$$

$$\sum_{p \in sensor-hops(i,d)} Z_p \ge 1, (\forall i) \tag{5}$$

From constraint (2), each actor $j$ is allowed to move to at most one location $p$. Constraint (3) requires that if an actor selects some location, that location must be among the $k$ selected locations. Constraint (4) forces the number of selected actor locations to be $k$. Finally, constraint (5) requires that each sensor be within at most $d$ hops from the new location of some actor.

### 6.2. Impact of optimal potential locations

Fig. 6 compares the results obtained from the ILP when $P = P(S, r)$ and $P = P_A(S, A, r)$. I.e., it compares the set of possible actor locations proposed in earlier works against the set we presented above. The same minimum radius is obtained in both cases, so we present only the total actor movement. Due to the large resources required for the ILP, we limit the number of sensor nodes to 50, and each point is the average of ten different scenarios.

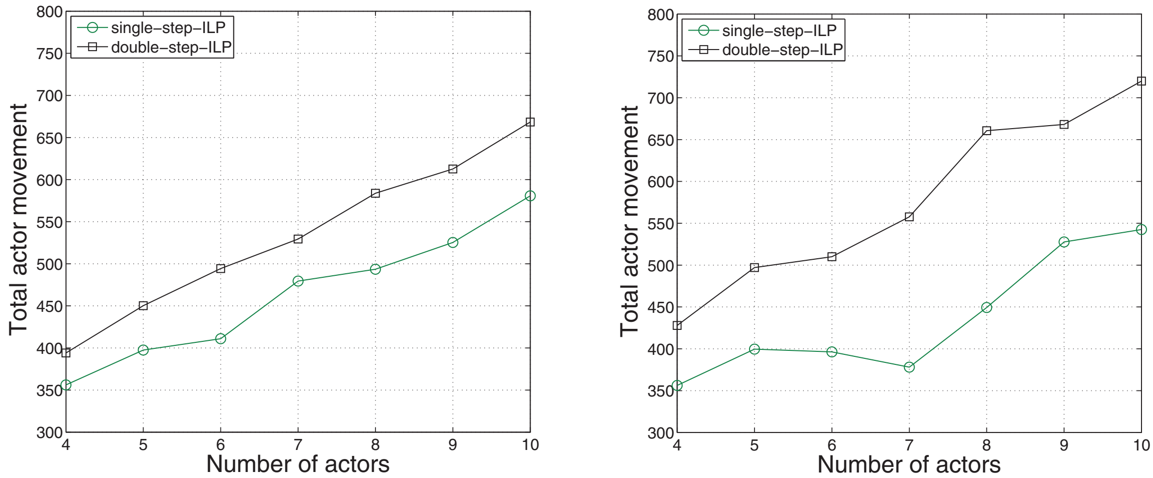
As can be seen from Fig. 6, the new set of possible actor locations $P_A(S, A, r)$ outperforms the set presented in earlier works. The difference becomes more significant as the number of hops increases, and also as the number of actors increases.

### 6.3. Double-step vs. single-step approach

Solving MRaMS requires us to find a set $B$ of $k$ points that cover all sensors within $d$ hops with a transmission radius of $r_{min}$, and also to find a mapping between the points in $B$ and the current actor locations $A$ so that the total movement from $A$ to $B$ is minimized.

With respect to mobility of the actors, we can thus take two different approaches to the problem. One is to obtain the $k$ points of $B$ first, and then based on the chosen points find a mapping to the current locations $A$. We refer to this as the *double-step approach*. The other approach is that, when selecting the $k$ points, we simultaneously take into consideration how these points are mapped to the current actor locations. We refer to this as the *single-step approach*.

The ILP formulation given in Section 6.1 uses the single-step approach. To find a solution using a double-step approach we solve two ILP systems, which we do not present for terseness. The first ILP system chooses the $k$ actor positions where the actors will be relocated to. The second ILP system uses as input the $k$ positions found by the first system, and assigns each actor to one of these $k$ positions such




(a) Total actor movement with $d = 1$

(b) Total actor movement with $d = 2$

**Fig. 7.** Comparison of the total actor movement with 50 sensor nodes by single and double step approaches.

that the total distance that the actors have to travel to their new position is minimized.

Consider Fig. 7, where we compare the single and double step ILP. The transmission range obtained is the same for both, so we only present the total actor movement. Due to the high computing cost of the ILP, we present only for 50 sensor nodes with $d = 1$ and $d = 2$. The number of actors is varied from four to ten.

As can be seen from the figure, the single-step approach obtains a significantly smaller total actor movement than the double-step approach. We will take advantage of this when we present our heuristics for MRaMS.

## 7. Heuristics for sensor coverage

The high complexity of finding a solution for MRaMS originates not in finding the minimum radius $r_{min}$, since this can be done through a binary search of the set of possible radii $R(S, d)$. The complexity originates from finding a set $B$ such that $cover(S, B, r, d)$ is true. As mentioned earlier, this problem in and of itself is an NP-complete problem even when $d = 1$ [13,15].

In this section, we set aside optimizing the movement of actors, and we focus on several heuristics for the $cover(S, B, r, d)$ problem. A binary search over all the radii is performed to determine which heuristic is able to find actor locations that cover all the sensors with the smallest transmission radius. In the next section, the winning heuristic will be used as a foundation for solving the MRaMS problem by incorporating the optimization of the movement of the actors to its new locations.

We present three different heuristics in this section. The first one is based on a heuristic for the $p$-center problem in operations research. The second heuristic is a greedy method that we originally proposed in [17]. The final heuristic is included for completeness, and is based on ignoring the specific locations of the sensor nodes, and simply attempting to cover the entire area with the given $k$ actors.

### 7.1. Sparse heuristic

We next describe a heuristic that was introduced in previous works [9]. The heuristic is given $S$, $r$, and $d$ as input, and returns the set $B$ of $k$ actor locations that cover all sensors with at most $d$ hops. We refer to this heuristic as the *sparse-heuristic*. The heuristic operates as follows.

- First, an actor location is chosen randomly from $P(S, r)$, and all sensor nodes within $d$ hops from the chosen location are removed from the graph.
- Then, a new actor location is chosen from $P(S, r)$ that is the farthest away from all previously chosen locations.
- These steps repeat until all $k$ actor locations have been chosen.

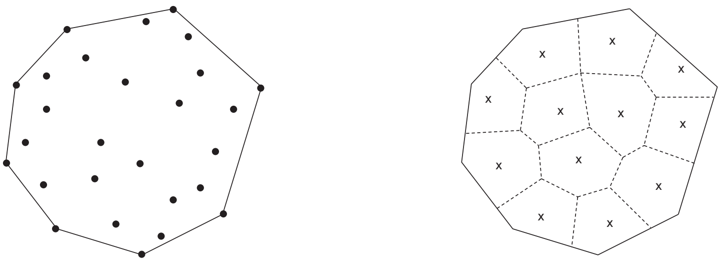
The heuristic declares success if all sensor nodes are eliminated from the graph after $k$ iterations. The pseudocode of the heuristic is illustrated in Algorithm 2 in more detail.

### 7.2. Dense heuristic

We next describe a greedy heuristic that we introduced in [1]. It has the same inputs and outputs as above. We refer to this heuristic as the *dense-heuristic*. Its operation is in rounds, as follows.

- Find an actor location $p$ from $P(S, r)$ such that maximum number of sensor nodes can be reached within $d$ hops.
- Remove from the graph those sensor nodes within $d$ hops from $p$.
- These steps repeat until all $k$ actor locations have been chosen.

The heuristic also declares success if all sensor nodes are eliminated from the graph after $k$ iterations. The pseudocode of the heuristic is illustrated in Algorithm 3 in more detail.



(a) Convex hull of sensors (b) Coverage by $k$ voronoi cells

**Fig. 8.** Coverage of convex hull.

**Algorithm 2** sparse-heuristic.

Inputs: $S, r, d$ Output: set $B$ of $k$ actor locations, empty if failure

$S' \leftarrow S$
$P' \leftarrow P(S, r)$
$B \leftarrow \emptyset$
**while** $|B| < k \wedge S' \neq \emptyset$ **do**
  **if** $B = \emptyset$ **then**
    Randomly select location $p$ from $P'$.
  **else**
    Find the location $p$ from $P'$ such that $p$ is farthest away from elements of $B$
  **end if**
  $P' \leftarrow P' - \{p\}$
  $B \leftarrow B + \{p\}$
  $S'_{d,p} \leftarrow$ subset of $S'$ within $d$ hops of $p$
  $S' \leftarrow S' - S'_{d,p}$
**end while**
Return $B$ if $S' = \emptyset$, otherwise return $\emptyset$.

**Algorithm 3** dense-heuristic.

Inputs: $S, r, d$ Output: set $B$ of $k$ actor locations, empty if failure

$S' \leftarrow S$
$P' \leftarrow P(S, r)$
$B \leftarrow \emptyset$
**while** $|B| < k \wedge S' \neq \emptyset$ **do**
  Find the potential location $p$ from $P'$ that can reach the maximum number of sensor nodes from $S'$ within $d$ hops
  $P' \leftarrow P' - \{p\}$
  $B \leftarrow B + \{p\}$
  $S'_{d,p} \leftarrow$ subset of $S'$ within $d$ hops of $p$
  $S' \leftarrow S' - S'_{d,p}$
**end while**
Return $B$ if $S' = \emptyset$, otherwise return $\emptyset$.

### 7.3. Voronoi heuristic

We next consider a heuristic which is oblivious to the positions of the sensors, and simply chooses $k$ locations for the actors such that they actors are evenly spaced out throughout the sensing area. The hope is that sensors are also evenly spaced out throughout the area, and thus each sensor will have an actor close by.

Because the sensor positions are ignored, one can come up with scenarios where the heuristic performs poorly by having the sensors positioned in a non-uniform manner throughout the area. If the sensors are placed randomly though, we expect reasonable performance.

In order to minimize the area to be covered, a convex hull can be drawn around the sensors (as shown in Fig. 8(a)) using any standard algorithm, such as *Andrew's Monotone Chain Algorithm* [28]. This results in a polygon containing all the sensors.

**Algorithm 4** Voronoi-heuristic.

Inputs: $S, d$ Output: set $B$ of $k$ actor locations and $r_{min}$

$Poly \leftarrow$ convex polygon surrounding $S$
$B \leftarrow k$ random locations inside inside initial $Poly$
$r \leftarrow \infty$
**while true do**
  Compute $VOR(Poly, B)$, and its corresponding cells $\{vor(b_1), \ldots, vor(b_k)\}$
  Compute the circumscribing circle $c_i$ of each cell $vor(b_i)$
  $B \leftarrow \{\text{center}(c_i) \mid 1 \leq i \leq k\}$
  $r' \leftarrow \textbf{max}\{\text{radius}(c_i), 1 \leq i \leq k\}$
  **if** $r = r'$ **then**
    **break**
  **else**
    $r \leftarrow r'$
  **end if**
**end while**
$LH \leftarrow \{\text{distance}(b_i, s_j) \mid 1 \leq i \leq k, 1 \leq j \leq n\}$
$SH \leftarrow \{\text{distance}(s_i, s_j) \mid 1 \leq i \leq n, 1 \leq j \leq n\}$
Let $r_{min}$ be the smallest element in $LH \bigcup SH$ such that all sensors can reach an actor in $d$ hops using transmission radius $r_{min}$
Return $B$ and $r_{min}$

An iterative solution to this problem has been proposed by [27], as follows. Given an initial random set $B$ of $k$ points, $\{b_1, b_2, \ldots, b_k\}$, located inside a convex polygon, a *voronoi diagram*, denoted by $VOR(B)$, is constructed. $VOR(B)$ consists of dividing the polygon into $k$ disjoint convex polygons, also known as *voronoi cells*. This is shown in Fig. 8(b).

Each voronoi cell, $vor(b_i)$, has the following interesting property. For any arbitrary point $p \in vor(b_i)$ and any other voronoi cell $vor(b_j)$, $EucDist(b_i, p) \leq EucDist(b_j, p)$, where *EucDist* denotes the Euclidean Distance between two points.

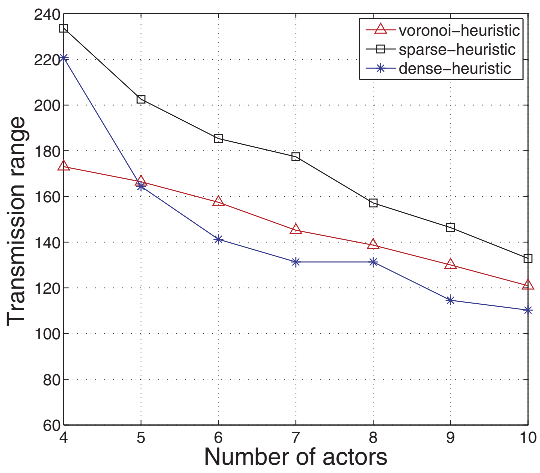
The original random points are unlikely to be evenly spread, so an iterative approach to refine their locations is given in [27] as follows. Each voronoi cell $vor(b_i)$ is circumscribed with the smallest possible circle $c_i$. Then, each point $b_i$ is relocated to the center of the circle $c_i$. This process is repeated until no more benefit is reaped from it. The heuristic is presented in more detail in Algorithm 4.

Once the set $B$ of points is obtained by the above method, what remains is to compute the minimum transmission radius $r_{min}$ such that all sensors can reach one of these points within $d$ hops. This can be done as follows.

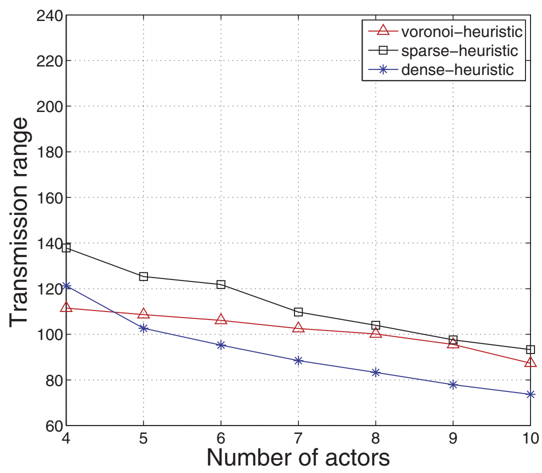
For a specific transmission radius $r$, we can find the number of hops from each sensor to an actor using standard minimum hop routing. The limiting factor in reducing the transmission radius must be either when a sensor cannot reach another neighboring sensor, or when a sensor cannot reach a neighbor actor. Thus, the set of possible transmission radii in which $r_{min}$ must be found is finite, because $r_{min}$ must be either the distance between some actor and some sensor, or the distance between some pair of sensors. This set has at most $n^2 + k \cdot n$ values, and a binary search can be performed over these values to obtain the smallest satisfactory radius.

### 7.4. Heuristics performance comparison

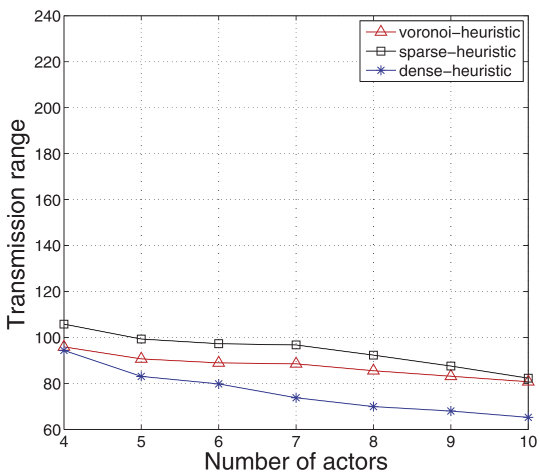
We implemented a simulator in C++ to compare the performance of the above heuristics. Fig. 9 compares the



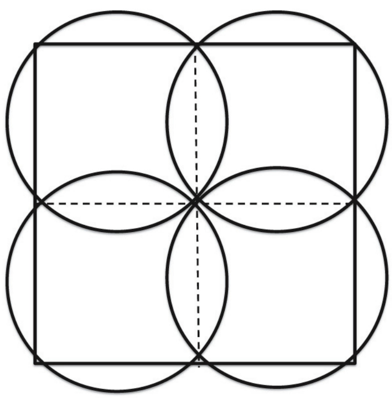

(a) Transmission range with $d = 1$




(b) Transmission range with $d = 2$




(c) Transmission range with $d = 3$



(d) Coverage of square with four actors and $d = 1$

**Fig. 9.** Comparison of heuristics for sensor coverage.

transmission range obtained from *voronoi-heuristic*, *sparse-heuristic* and *dense-heuristic*. Each point is the average of 30 simulation runs. Each run consists of 100 sensor nodes randomly deployed in an area of 500 × 500 m. The number of actors is varied from 4 to 10. The hop count is also varied from $d = 1$ up to $d = 3$.

As can be seen from Fig. 9, *dense-heuristic* performs best overall. It is only surpassed by the *voronoi-heuristic* when the number of actors is 4 and the hop count is low, in particular, when the hop count is one. A possible explanation for this is the following. Consider Fig. 9(d), where the 500 × 500 m square is shown divided into four quadrants. Dividing this square into four equal sized voronoi cells results in the square broken into four quadrants, denoted by the dashed lines, with each quadrant being a voronoi cell. The center of each of these quadrants becomes an actor location. The radius needed for the four circles to totally cover the square is about 170 m. This is the radius reported by the voronoi-heuristic in Fig. 9(a).

The dense-heuristic, on the other hand, is not an area-coverage but a point-based heuristic. Thus, assume that there are four concentrations of sensors close to the four corners of the square. In this case, dense-heuristic will choose four locations significantly away from the center of the square. If there are a few sensors in the center, then the transmission radius needed will be larger than that required by the voronoi heuristic.

On the other hand, consider the case where all sensors are located in one quadrant. In this case, voronoi-heuristic would return the same result, since it is oblivious to sensor positions. The dense-heuristic would return a much smaller radius by placing all actors in the same quadrant as the sensors.

Overall, as the number of actors increases, and as the hop count increases, the clear winner is the *dense-heuristic*. Based on this knowledge, we will use the *dense-heuristic* as the foundation for our heuristic for the MRaMS problem in Section 8.

## 8. Heuristics for MRaMS

In Section 7.4, we evaluated multiple heuristics that obtain a set of new actor locations $B$ and a smallest radius $r_{min}$ such that $cover(S, B, r_{min}, d)$ is true. We discovered that *dense-heuristic* works best among these heuristics, so we incorporate this heuristic into our MRaMS heuristics.

The movements of actors still needs to be addressed. That is, set $B$ must be viewed as a sequence rather than a set. This allows us to map each element in the original actor sequence $A$ to an element in $B$, resulting in an actor relocation $(A, B)$.

To address actor movement, we have two options. The first option is the double-step approach, in which $cover(S, B, r, d)$ returns a set $B$ of new actor locations, followed by a step where the elements of $B$ are ordered into a sequence and mapped to the elements of the original actor sequence $A$, resulting in the actor relocation $(A, B)$. For the double-step approach, we choose a greedy heuristic for the mapping from $A$ to $B$, such as the one used in [23]. Actors are assigned to new locations in order of increasing distance, i.e., the actor that is closest to a new location is the first to be paired, followed by the next actor that is closest to a remaining location, etc.. We refer to this heuristic as the *double-step-dense heuristic*.

The second option is the single-step approach, in which as soon as an element is added to $B$, the element is also paired with an element of $A$. Thus, the actor relocation $(A, B)$ is obtained at the same time that $cover(S, B, r, d)$ is solved. For the single-step approach, we will use an enhanced version of the single-step heuristic that we presented in [3], which we overview below. Compared to the double-step approach, it delivers significant gains in actor mobility, at the expense of only a modest increase in transmission radius. We will investigate the performance of this single-step heuristic with two new additions over our earlier work [3]: the network will be multi-hop rather than single-hop, and we will use the new locations $P_A(S, A, r)$ introduced in this paper.

To incorporate actor movement into the *dense-heuristic*, we need more flexibility in the choice of the next actor location. At each iteration, we select a subset of actor locations that provide sensible, but not necessarily the largest, coverage of sensors. From this set, we choose the location with the smallest distance to any unassigned actor. This adds an (actor, new location) pair to $(A, B)$. The flexibility in choosing the subset of actor locations is governed by a user-defined parameter $\alpha$, $0 \le \alpha \le 1$.

We thus perform the following steps for $k$ iterations.

- Determine the maximum number of sensor nodes (*MaxSensors*), which are covered by any one potential location from $P_A(S, A, r)$.
- Find the subset of locations from $P_A(S, A, r)$ such that each location covers at least ($MaxSensors \times (1 - \alpha)$) sensor nodes.
- From these possible new locations and from the initial actor locations, select the pair (initial actor location $a$, new

**Algorithm 5** Single-step-dense-heuristic.

Inputs: $S, A, r, d, \alpha$, Output: actor relocation $(A, B)$ or empty-set if failure

$B \leftarrow < nil >$ ▷ sequence of $k$ nil values
$A' \leftarrow A$ ▷ set unmapped actor locations
$P' \leftarrow P_A(S, A, r)$ ▷ set unused possible actor locations
$S' \leftarrow S$ ▷ set uncovered sensor nodes
**while** $|A'| > 0$ **do**
  Let *MaxSensors* be the maximum number of sensors
    in $S'$ that can be reached from a single location
    in $P'$ using at most $d$ hops
  Let $T$ be the subset of $P'$ such that each location in $T$
    covers at least ($MaxSensors \times (1 - \alpha)$)
    sensor nodes in $S'$ using at most $d$ hops
  Let $(x, t)$, where $x \in A'$ and $t \in T$, be the pair
    with minimum distance from $x$ to $t$
  Let $i$ be the index of $x$ in $A$ ▷ i.e., $a_i = x$
  $B[i] \leftarrow t$ ▷ $b_i$ is set to $t$, thus, $(a_i, b_i)$ are paired
  $A' \leftarrow A' - \{a\}$
  $S'_d \leftarrow$ subset of $S'$ within $d$ hops of $t$
  $S' \leftarrow S' - S'_d$
**end while**
Return $(A, B)$ if $S' = \emptyset$, otherwise the empty set

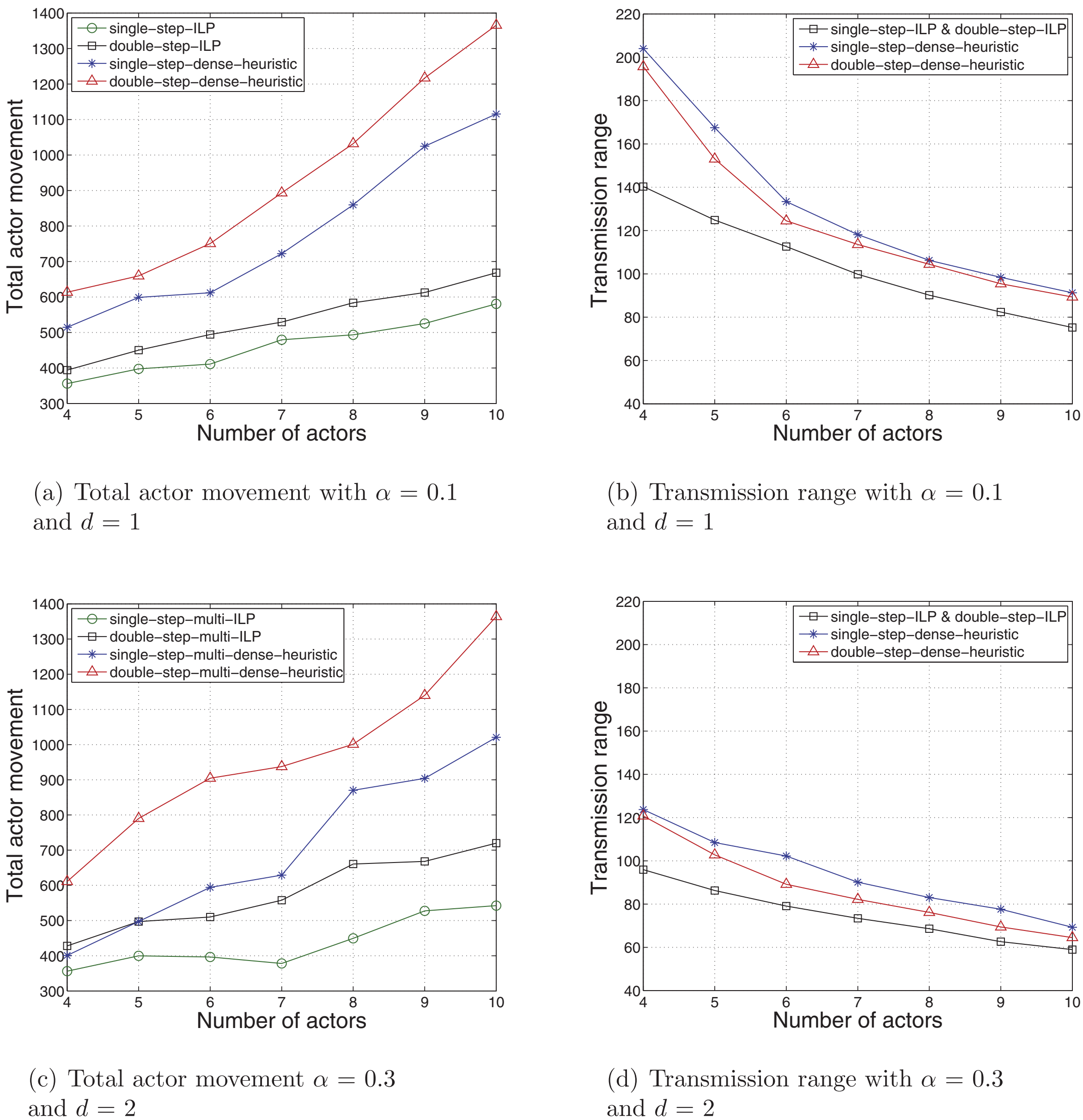
location $b$) with the closest distance from $a$ to $b$. Add this pair to $(A, B)$.

- Search for sensor nodes within $d$ hops from $b$.
- The above sensor nodes are then removed from the graph.

The details are given in Algorithm 5. For a given radius $r$, the heuristic returns an actor relocation $(A, B)$. Thus, a binary search over all possible radii is still required to obtain the minimum radius $r_{min}$. We refer to this heuristic as the *single-step-dense-heuristic*.

A few remarks about our single-step heuristic are worth highlighting. First, as $\alpha$ is increased, we expect to see a reduction in actor movement at the expense of a larger transmission radius. Because in each iteration we do not necessarily choose the location with the best sensor coverage, we might obtain a larger radius than the double-step approach. However, we would like to emphasize that significant gains are made in the reduction of actor movement (as shown in Section 9) without significant sacrifices in radius.

An interesting choice for $\alpha$ is $\alpha = 0$. This places the greatest value in reducing transmission radius, with an unwillingness to sacrifice any transmission radius in favor of actor movement. When $\alpha = 0$, the single-step-dense heuristic appears to be very similar to the double-step-dense heuristic. However, these two schemes are non-comparable: scenarios can be constructed where each of them outperforms the other. To highlight some of their differences, consider how ties are broken when choosing a new location to place an



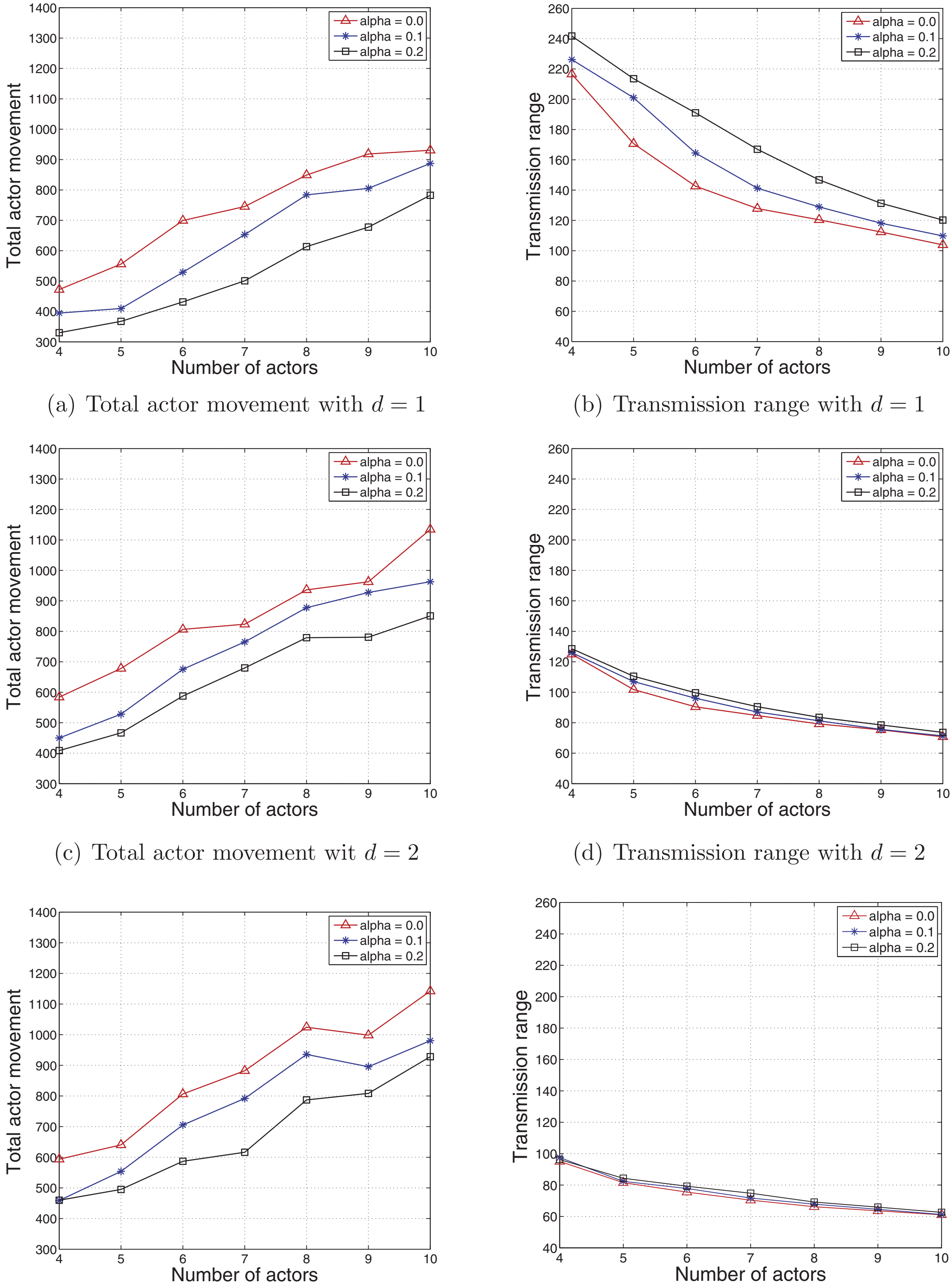

(a) Total actor movement with $\alpha = 0.1$ and $d = 1$

(b) Transmission range with $\alpha = 0.1$ and $d = 1$

(c) Total actor movement $\alpha = 0.3$ and $d = 2$

(d) Transmission range with $\alpha = 0.3$ and $d = 2$

**Fig. 10.** Comparison of the total actor movement and the transmission range with 50 sensor nodes by different approaches.




(a) Total actor movement with $d = 1$

(b) Transmission range with $d = 1$

(c) Total actor movement wit $d = 2$

(d) Transmission range with $d = 2$

(e) Total actor movement with $d = 3$

(f) Transmission range with $d = 3$

**Fig. 11.** Total actor movement and transmission range with 100 sensor nodes by different $\alpha$(alpha) using *single-step-dense-heuristic* with different hop bound.

actor. If multiple locations can reach the same number of sensor nodes, then the double-step heuristic will choose one of these at random, while the single-step heuristic will choose the location that is closest to an unassigned actor. It is possible that both the single and double step will, by chance, choose the same set of actor locations *B*. However, the pairing of the new locations, *B*, with the original locations, *A*, is done at the end of the double-step approach, while the single-step approach performs the pairing as the locations are chosen, which could lead to different pairings and thus different actor movements.

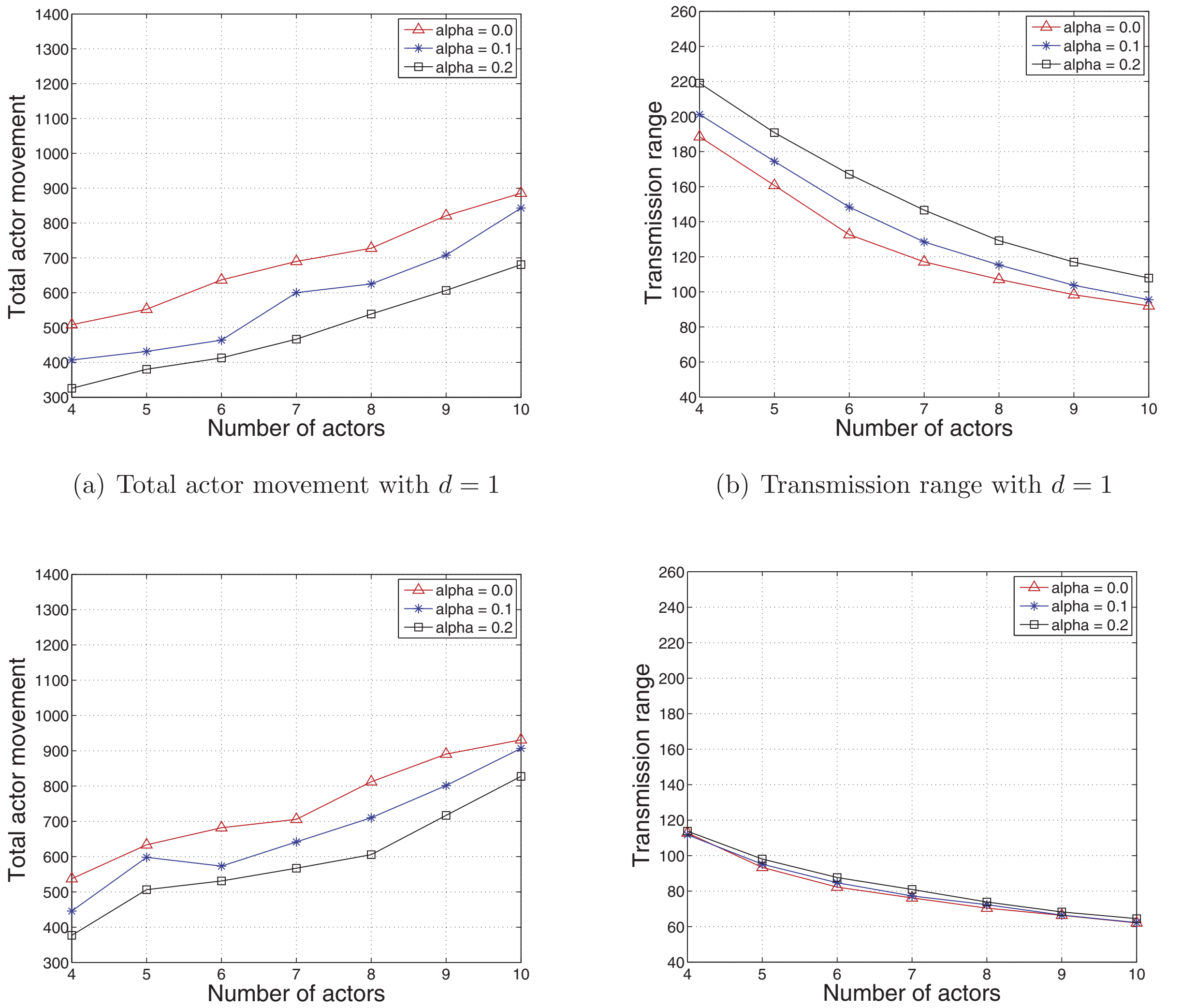
## 9. Experimental evaluation

In this section, we evaluate the performance of our MRaMS heuristics and compare them against the ILP formulation.

We analyze four different approaches: *single-step-dense-heuristic*, *double-step-dense-heuristic*, *single-step-ILP* and *double-step-ILP*. We have simulated our various experiments in a square-shaped and two-dimensional sensing area of size 500 m × 500 m where both *n* sensor nodes and *k* actors are randomly deployed. Each experiment represents the average result of 10 different graphs.

When we execute *single-step-dense-heuristic* and *double-step-dense-heuristic*, we have used a custom-made program based on $C^{++}$, and the results for *single-step-ILP* and *double-step-ILP* are obtained using CPLEX [26].

In our first group of simulations, we compare all four different methods using the new set $P_A(S, A, r)$ of actor locations. We use two evaluation criteria: the minimum radius obtained, and the total actor movement. Due to the overhead of the ILP, we limit the number of sensor nodes to 50. We choose two values for $\alpha$, 0.1 and 0.3, and consider networks




(a) Total actor movement with $d = 1$

(b) Transmission range with $d = 1$

(c) Total actor movement wit $d = 2$

(d) Transmission range with $d = 2$

**Fig. 12.** Total actor movement and transmission range for sensing area size 500 × 400 with 100 sensor nodes by different $\alpha$(alpha) using *single-step-dense-heuristic* with different hop bound.

with hop bound $d = 1$ and $d = 2$. The number of actors ranges from 4 to 10. The results of the first simulations are shown in Fig. 10.

As expected, *single-step-ILP* always outperforms the other schemes because it is guaranteed to be optimal. The next best scheme is *double-step-ILP*, which is optimal for transmission radius, but not optimal for actor movement. Nonetheless, it is able to outperform the heuristics. When we analyze the performance of the heuristics, *single-step-dense-heuristic* is sufficiently better for total actor movement distance than *double-step-dense-heuristic* as the number of actors increases. In particular, it is observed that the total actor movement for the *single-step-dense-heuristic* is at most twice the total actor movement of the *single-step-ILP*. Reaching a value no more than twice the optimal is a significant achievement for a heuristic.

For transmission range, as the number of actors in the network grows, the result of our *single-step-dense-heuristic* becomes closer to the optimum using ILP. For example, when the number of actors is 10, the gap between the *single-step-ILP* and *single-step-dense-heuristic* is very small.

For the second performance analysis, we evaluate the impact of the parameter $\alpha$ on our *single-step-dense-heuristic*. Since we are not constrained by the overhead of the ILP, we increase the number of sensor nodes to 100 and show average results by 30 different graphs. We choose $\alpha = 0.0$, $\alpha = 0.1$ and $\alpha = 0.2$, and consider hop bounds of $d = 1$, $d = 2$ and $d = 3$. The results are shown in Fig. 11.

The figures show that the total actor movement decreases significantly as $\alpha$ increases. This is because of the greater flexibility in choosing actor locations at each step in the heuristic. With respect to transmission range, only a slight

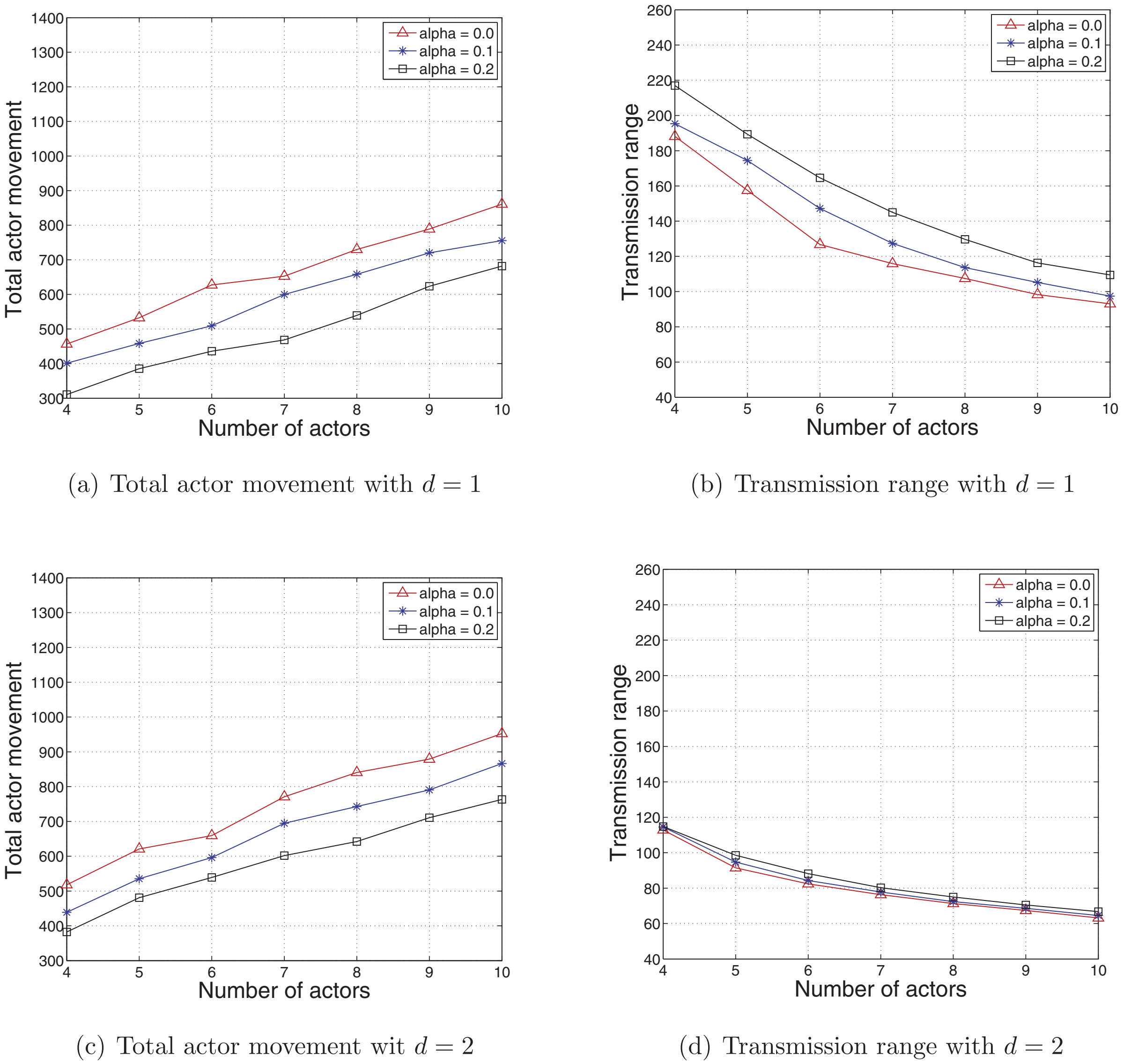



**Fig. 13.** Total actor movement and transmission range for sensing area size 400 × 500 with 100 sensor nodes by different $\alpha$(alpha) using *single-step-dense-heuristic* with different hop bound.

increase in transmission range occurs as we increase $\alpha$. Thus, a significant decrease in actor movement is obtained at the cost of a small increase in transmission range. This result should be a clear advantage of our *single-step-dense-heuristic* for MRaMS.

Note also that, for hop bound $d = 1$, as the number of actors increases, the difference between the transmission range of the three $\alpha$ values decreases. For hop bounds $d = 2$ and $d = 3$, the difference in transmission radius of the three $\alpha$ parameters is negligible.

Finally, for our third set of experiments, we considered different field sizes: two-dimensional sensing areas of size 500 m $\times$ 400 m and 400 m $\times$ 500 m.

We performed *single-step-dense-heuristic* to check the impact by different $\alpha = 0.0, 0.1, 0.2$. The numbers of actors ranges from 4 to 10 and the number of sensor nodes is set to 100. These simulations are implemented with hop bounds of $d = 1$ and $d = 2$, and each graph represents the average value of 30 different graphs. The results are shown in Fig. 12 for 500 m $\times$ 400 m and Fig. 13 for 400 m $\times$ 500 m. Similar to the second performance analysis, both figures shows that our *single-step-dense-heuristic* decreases the total actor movement as $\alpha$ increases at the expense of only a modest increase in transmission range.

## 10. Concluding remarks

In this paper, we have introduced the MRaMS problem consisting of the simultaneous optimization of transmission range and actor movement for multi-hop communication in WSANs. Even though an actor may be placed at any point in the field, we have presented a new and finite set of possible locations for placing actors that guarantee that the optimal solution is found within this set. We thus presented an optimal ILP formulation based on this set.

Also, we developed heuristics for MRaMS. As an intermediate step, we first studied how to minimize the maximum node transmission range for multi-hop communication in WSANs. To solve the problem, a heuristic is proposed and compared against other heuristics. Then, we analyzed the performance of these heuristics and showed our heuristic outperforms the others.

Based on our heuristic for transmission range, we proposed a novel heuristic for the MRaMS problem. From our performance evaluation, we conclude that a significant decrease in total actor movement can be achieved at the cost of a small, or even negligible, increase in transmission range. As a future work, we plan to consider the development of distributed algorithms to solve MRaMS problem and we will also consider simultaneous optimization of transmission radius and actor movement in the three-dimension environment.

## References


[1] H. Kim, J.A. Cobb, Optimal transmission range for multi-hop communication in wireless sensor and actor networks, in: Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN), 2011.
[2] H. Kim, J.A. Cobb, Optimization trade-offs in the design of wireless sensor and actor networks, in: Proceedings of the 37th IEEE Conference on Local Computer Networks (LCN), 2012.
[3] H. Kim, J.A. Cobb, Simultaneous optimization of transmission range and actor movement in WSANs, in: Proceedings of IEEE International Conference on Computing, Networking and Communications (ICNC), 2012.
[4] I.F. Akylidiz, I.H. Kasimoglu, Wireless sensor and actor networks: research challenges, Ad Hoc Netw. J. 2 (2004) 351–367.
[5] T. Melodia, D. Pompili, V.C. Gungor, I.F. Akylidiz, A distributed coordination framework for wireless sensor and actor networks, in: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Urbana-Champaign, IL, 2005.
[6] T. Melodia, D. Pompili, V.C. Gungor, I.F. Akylidiz, Communication and coordination in wireless sensor and actor networks, IEEE Trans. Mobile Comput. 6 (10) (2007) 1116–1129.
[7] T. Melodia, D. Pompili, I.F. Akylidiz, A communication architecture for mobile wireless sensor and actor networks, in: Proceedings of IEEE International Conference on Sensor, Mesh and Ad Hoc COmmunications and Networks (SECON), Urbana-Champaign, IL, 2005.
[8] T. Melodia, D. Pompili, I.F. Akyildiz, Handling mobility in wireless sensor and actor networks, IEEE Trans. Mobile Comput. 9 (2) (2010) 160–173.
[9] D. Kim, W. Wang, N. Sohaee, C. Ma, W. Wu, W. Lee, D.Z. Du, Minimum data latency bound k-sinks placement problem in wireless sensor networks, IEEE/ACM Trans. Netw. 19 (5) (2011) 1344–1353.
[10] W. Youssef, M. Younis, Intelligent gateways placement for reduced data latency in wireless sensor networks, in: Proceedings of International Conference on Communications (ICC), 2007, pp. 3805–3810.
[11] G. Wang, G. Cao, T.L. Porta, W. Zhang, Sensor relocation in mobile sensor networks, in: Proceedings of IEEE International Conference on Computer Communications (INFOCOM), Miami, FL, 2005.
[12] K. Akkaya, F. Senel, Detecting and connecting disjoint sub-networks in wireless sensor and actor networks, Elsevier Ad Hoc Netw. J. 7 (7) (2009) 1330–1346.
[13] J. Tang, B. Hao, A. Sen, Relay node placement in large scale wireless sensor networks, Comput. Commun. 29 (4) (2006) 490–501.
[14] Z. Drezner, The p-centre problem - heuristic and optimal algorithms, J. Oper. Res. Soc. 35 (8) (1984) 741–748.
[15] M.E. Dyer, A.M. Frieze, A simple heuristic for the p-center problem, Oper. Res.Lett. 3 (6) (1985) 285–288.
[16] R.Z. Hwang, R.C.T. Lee, R.C. Chang, The slab dividing approach to solve the Euclidean p-center problem, Algorithmica 9 (1993) 1–22.
[17] H. Kim, J.A. Cobb, Optimal transmission range and actor movement in wireless sensor and actor networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2011.
[18] N. Sohaee, Optimization in Design of Underwater Sensor Networks (Ph.d. thesis), The University of Texas at Dallas, 2009.
[19] K. Akkaya, M. Younis, Cola: a coverage and latency aware actor placement for wireless sensor and actor networks, in: Proceedings of IEEE Vehicular Technology Conference (VTC), Montreal, CA, 2006.
[20] D. Kim, B.H. Abay, R.N. Uma, W. Wu, W. Wang, A.O. Tokuta, Minimizing data collection latency in wireless sensor network with multiple mobile elements, in: Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM 2012), 2012, pp. 504–512.
[21] D. Kim, R.N. Uma, B.H. Abay, W. Wu, W. Wang, A.O. Tokuta, Minimum latency multiple data mule trajectory planning in wireless sensor networks, in: IEEE Transactions on Mobile Computing (TMC), 2013.
[22] L. Xue, D. Kim, Y. Zhu, D. Li, W. Wang, A.O. Tokuta, Multiple heterogeneous data ferry trajectory planning in wireless sensor networks, in: Proceedings of the 33st IEEE International Conference on Computer Communications (INFOCOM 2014), 2014, pp. 2274–2282.
[23] K. Akkaya, I. Guneydas, A. Bicak, Autonomous actor positioning in wireless sensor and actor networks using stable-matching, Int. J. Parallel Emerg. Distrib. Syst. (IJPEDS) 25 (6) (2010) 439–464.
[24] A. Abbasi, M. Younis, K. Akkaya, Movement assisted connectivity restoration in wireless sensor and actor networks, IEEE Trans. Parallel Distrib. Syst. 20 (9) (2009) 1366–1379.
[25] S. Wang, X. Mao, S. Tang, L. X, J. Zhao, G. Dai, On movement-assisted connectivity restoration in wireless sensor and actor networks, IEEE Trans. Parallel Distrib. Syst. 22 (4) (2011) 687–694.
[26] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.
[27] G.K. Das, S. Das, S.C. Nandy, B.P. Sinha, Efficient algorithm for placing a given number of base stations to cover a convex region, J. Parallel Distrib. Comput. 66 (11) (2006) 1353–1358.
[28] A.M. Andrew, Another efficient algorithm for convex hulls in two dimensions, Inf. Process. Lett. 9 (1979) 216–219.

**Hyunbum Kim** is currently an Assistant Professor in the Department of Computer Science at University of North Carolina at Wilmington. He received his Ph.D. in computer science from the University of Texas at Dallas. His research interests include algorithm design/analysis/optimization in various areas including wireless sensor networks, mobile computing, cyber physical systems, distributed computing and cyber security.

**Jorge A. Cobb** is currently an Associate Professor in the Department of Computer Science at The University of Texas at Dallas. He received the B.S. in Computer Science (with highest honors) from The University of Texas at El Paso in 1987. He also received an M.A. in Computer Science and a Ph.D. in Computer Science from The University of Texas at Austin. While in the Ph.D. program at The University of Texas at Austin, he was awarded Ph.D. Scholarship. He has been a member of the technical program committee of many international conferences, including the IEEE International Conference on Network Protocols and the IEEE Local Computer Networks conference. His main research area is computer networking, with an emphasis on scheduling for quality of service guarantees, wireless networks, inter-domain routing, and self-stabilizing systems.