Distributed Deployment Algorithms for Coverage Improvement in a Network of Wireless Mobile Sensors: Relocation by Virtual Force

Hamid Mahboubi and Amir G. Aghdam

Abstract

Efficient deployment algorithms are developed in this paper to increase the coverage area in a network of wireless mobile sensors. The proposed strategies iteratively compute the new candidate position of each sensor based on the existing coverage holes. These holes are obtained using a Voronoi diagram for the case of sensors with the same sensing ranges, and a multiplicatively weighted Voronoi (MW-Voronoi) diagram for the case of sensors with different sensing ranges. Each sensor is driven by some virtual forces which are applied to it from the vertices and boundaries of its Voronoi cell. These forces are obtained in such a way that when the sensor is relocated, the covered area of the corresponding cell increases. Simulation results demonstrate the efficacy of the proposed strategies, and their superiority to existing algorithms.

I. Introduction

Wireless sensor networks have attracted considerable attention in different areas of science and engineering recently due to their broad range of applications such as environmental monitoring, biomedical systems and traffic control, to name only a few [1], [2], [3], [4], [5], [6]. A mobile sensor network (MSN) is comprised of a number of wireless nodes capable of moving in different directions and communicating with their neighbors. It is desired in this type of system to cooperatively achieve a global objective such as tracking a moving target [7] or improving network coverage [8]. Furthermore, it is often desirable

H. Mahboubi and A. G. Aghdam are with the Department of Electrical & Computer Engineering, Concordia University, 1455 de Maisonneuve Blvd. W., EV012.179, Montréal, Québec H3G 1M8 Canada, {h_mahbo, aghdam}@ece.concordia.ca

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant RGPIN-262127-12.

to use a distributed decision-making algorithm, where each sensor is assumed to communicate with its neighbors only [9]. In addition, the deployment strategy should ideally be independent of the initial location of sensors because in practice these locations are not known *a priori* [10]. Moreover, the cost-effective resource management techniques are required to prolong the network lifetime [11], [12].

The Voronoi diagram is often used for coverage analysis in sensor networks. A Voronoi-based technique is proposed in [13] to improve coverage in a sensor network with no requirement of global location assurance condition for the sensors. In [14], distributed gradient-descent algorithms are given to increase sensing coverage using the Delaunay graph. A class of aggregate objective functions is studied in [15] using the geometry of the Voronoi cells and proximity graphs. An algorithm is developed in [16] for efficient sensor deployment and power assignment in a sensor network. To this end, a multi-objective optimization problem is introduced which is reformulated as a group of single-objective scalar problems.

Non-smooth gradient flows are used in [17] to develop distributed control strategies for the problems of disk covering and sphere packing. In [18], the problem of positioning a group of sensors in a region for detection purposes is investigated by minimizing the maximum probability of non-detection. A decentralized adaptive control law is developed in [19] to properly place a group of sensors in an environment for optimal sensing coverage. Distributed control strategies are proposed in [20] to obtain a convex equi-partition configuration in an MSN. Effective deployment techniques are then developed to increase the sensing coverage. The problem of covering an environment using a group of mobile robots with different sensor footprints is considered in [21]. An efficient deployment algorithm is proposed in [22] which finds the appropriate locations for the mobile sensors by minimizing the maximum error variance and extended prediction variance. In [23]-[27], efficient coverage strategies are developed which do not use simple sensing models or Voronoi partitions. Distributed gradient-based techniques are presented in [23], [24] for optimal coverage in an MSN. To this end, the sensors cooperatively optimize a probabilistic detection metric, as opposed to a simple geometric area metric. Distributed control strategies are introduced in [25] for optimal coverage in an environment with obstacles, where the sensors' field-of-view is limited. In [26], distributed convergence to a Nash equilibrium in an MSN is investigated. A coverage algorithm is provided in [27] for maximizing the probability of detection, where the communications cost is minimized in order to increase the network lifetime. Three distributed deployment algorithms, namely, VEC, VOR and Minimax are introduced in [8] which find proper

locations for sensors to improve coverage. These techniques, however, may not perform efficiently for certain network configurations [28].

In this paper, novel deployment techniques are proposed to increase coverage in a network of mobile sensors. Three algorithms are developed: vertex virtual forces (VVF) algorithm, edge virtual forces (EVF) algorithm, and vertex-edge virtual forces (VEVF) algorithm. The above-mentioned algorithms are then extended to the case of a network with nonidentical sensors using multiplicatively-weighted Voronoi (MW-Voronoi) diagrams [29]. Virtual forces are applied to each sensor from every vertex and boundary of the Voronoi cell containing the sensor, which tend to move the sensor. Each sensor will move only if the coverage area increases from its new location.

The paper is organized as follows. In Section II, some preliminaries concerning the Voronoi diagram are presented, and important definitions and assumptions are also given. Section III provides new methods for efficient coverage in a network of identical sensors. These algorithms are extended to the case of nonidentical sensors in Section IV. Simulations are given in Section V, which demonstrate the efficacy of the proposed deployment strategies. The paper concludes with a summary in Section VI.

II. Preliminaries

Consider a set of n nodes in a 2D plane, denoted by $\bar{\mathbf{S}} := \{S_1, S_2, \ldots, S_n\}$. The Voronoi diagram partitions the plane into n convex polygons, each containing only one node. Each polygon is called a Voronoi polygon, and the node inside it is referred to as the *generating node* of that polygon. The Voronoi polygons are constructed in such a way that the generating node of each polygon is the nearest node to any point inside that polygon [30]. To construct the Voronoi diagram, the perpendicular bisector of any segment connecting a node to each one of its neighboring nodes is drawn first. The smallest polygon created by these perpendicular bisectors which contains the node is, in fact, the Voronoi polygon associated with that node. The Voronoi diagram is used for the analysis and synthesis of sensor deployment strategies.

Consider now n mobile sensors in a 2D field of polygon shape. Let each sensor be represented by a node; construct the Voronoi polygon for every node as explained above, to cover the whole field.

Definition 1. Consider a sensor S_i, $i \in \mathbf{n} := \{1, \ldots, n\}$ with the sensing radius r. Let Π_i denote the Voronoi polygon of S_i, and Q be an arbitrary point inside Π_i. The intersection of the polygon Π_i and

a circle of radius r centered at Q is called the *i-th coverage area with respect to Q*, and is represented by $\beta^{Q}_{\Pi_i}$. The i-th coverage area with respect to the position of the sensor S_i is referred to as the *local coverage area* of that sensor.

Definition 2. Consider an arbitrary point Q inside the Voronoi polygon Π_i, $i \in \mathbf{n}$. The intersection of the Voronoi polygon Π_i and the region outside the i-th coverage area with respect to Q is called the *i-th coverage hole with respect to Q*, and is denoted by $\theta^{Q}_{\Pi_i}$. The i-th coverage hole with respect to the position of the sensor S_i is referred to as the *local coverage hole* of that sensor.

As noted earlier, the nearest node to any point inside a Voronoi polygon is the generating node of that polygon. Thus, assuming that all sensors have the same sensing range, if a sensor cannot sense a certain point inside its polygon, no other sensor in the network can sense it either. This implies that in order to discover the coverage hole of a network, it suffices to find all local coverage holes first. The total coverage hole of the network is, in fact, the union of all local coverage holes.

III. Deployment Protocols for Identical Sensors

Three sensor deployment protocols are introduced in this section for efficient coverage in a network of identical sensors. The algorithms are iterative, and every iteration includes four steps. In the first step, every sensor S_i, $i \in \mathbf{n}$, transmits its position information P_i to other sensors, receives similar information from other sensors, and subsequently constructs its Voronoi polygon. In the second step, every sensor finds the local coverage hole by checking its Voronoi polygon. If a coverage hole exists in a polygon, say the i-th polygon, then a proper scheme is used to find a point $\acute{P}_i$ in it such that by placing the sensor in that point, the coverage hole would be closed or at least would become smaller by a certain threshold. Once the new destination is found, the coverage area of the sensor with respect to this location (i.e. $\beta^{\acute{P}_i}_{\Pi_i}$) is obtained in the third step. If the coverage area with respect to the new destination is greater than the local coverage area, i.e. $\beta^{\acute{P}_i}_{\Pi_i} > \beta^{P_i}_{\Pi_i}$, the sensor moves there; otherwise, it stays at its present location. In order for the algorithm to terminate in finite time, a proper threshold ϵ is considered such that if the size of the local coverage area of every sensor does not increase more than ϵ, the iterations stop. Note that the algorithms developed in this section differ only in the second step. In each round, the new destination for each sensor is computed according to an appropriate deployment technique.

As noted above, under the techniques introduced in this work a sensor moves to a new location only if the coverage area of that sensor with respect to the new position in the old Voronoi polygon increases. Similar to Theorem 1 of [28], it can be shown that under this type of strategy the total coverage is guaranteed to increase.

A. Vertex Virtual Forces (VVF) Strategy

The movement of each sensor under the VVF strategy depends on some virtual forces applied from the vertices of the Voronoi polygon containing the sensor. Denote the vertices of the i-th Voronoi polygon by $\mathbf{V}_i = \{V_{i1}, V_{i2}, \ldots, V_{il}\}$. Also, denote by $d_{V_{ij}}^{S_i}$ the distance between sensor S_i and the j-th vertex of its Voronoi polygon. Denote the sensing radius of every sensor by r. In this strategy, if the distance between sensor S_i and vertex V_{ij} is less than r, then a virtual force form V_{ij} will push S_i, tending to move the sensor away by $r - d_{V_{ij}}^{S_i}$. If on the other hand $d_{V_{ij}}^{S_i} > r$, then a virtual force form V_{ij} will pull the sensor, tending to move it toward V_{ij} by $d_{V_{ij}}^{S_i} - r$. Eventually, each sensor moves in the direction of the vector sum of all virtual forces, $\overrightarrow{V_v^i}$, applied to it from the vertices of the corresponding Voronoi polygon. The new destination $\acute{P}_i$ is equal to $P_i + \alpha\overrightarrow{V_v^i}$, where α is a parameter (not necessarily constant) which is to be chosen properly. For example, a line search procedure can be used to find the optimal value for α in order to maximize the i-th local coverage area. However, in this paper α is chosen as $\frac{1}{4}$ based on simulation to reduce the computational complexity of the strategy [31].

Figure 1 shows an illustrative example of the VVF strategy. In this figure, the virtual forces applied to the sensor are depicted by dashed vectors, and the displacement $\frac{1}{4}\overrightarrow{V_v^i}$ is shown by a red vector.

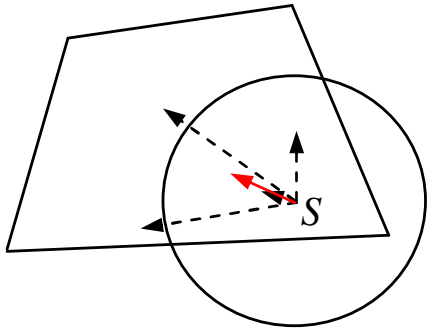

Fig. 1: An illustrative example of the VVF strategy.

B. Edge Virtual Forces (EVF) Strategy

The sensor deployment strategy introduced in the previous subsection is vertex-based, as it operates based on the distance of every sensor from the vertices of its Voronoi polygon. Although this algorithm proves effective in many scenarios, it is sometimes not as effective for specific sensor configurations. Consider for example the polygon in Figure 2, where the sensor is denoted by S. It can be easily verified that the sensor should move in the up-left direction in order for the coverage area to increase. However, under the VVF strategy the sensor is forced (by the corresponding virtual forces) to move to point A (which is almost in the opposite direction in this specific configuration), although the movement adjustment scheme described earlier does not allow the sensor to move. This shortcoming of the VVF algorithm can be addressed by using an edge-based method described in the sequel.

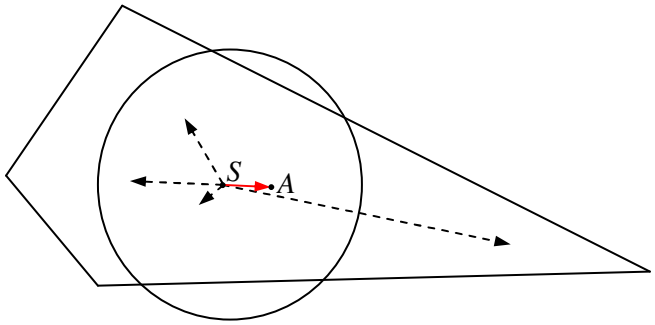

Fig. 2: A configuration for which the VVF technique is not as effective.

Denote the set of edges of the i-th Voronoi polygon by $\mathbf{E}_i = \{E_{i1}, E_{i2}, \ldots, E_{il}\}$, and the distance between sensor S_i and the j-th edge of its Voronoi polygon by $d_{E_{ij}}^{S_i}$. In the EVF method, the movement of each sensor results from the vector sum of all virtual forces applied from the edges of its polygon. If the distance between sensor S_i and edge E_{ij} is greater than the sensing radius r, then a virtual force form E_{ij} will pull S_i, tending to move the sensor toward the edge by $d_{E_{ij}}^{S_i} - r$. If on the other hand $d_{E_{ij}}^{S_i} < r$, then a virtual force form E_{ij} will push S_i, tending to move it away by $r - d_{E_{ij}}^{S_i}$. Similar to the VVF strategy, the displacement of a sensor under the EVF technique is proportional to the vector sum of all virtual forces applied to it from the edges of the corresponding Voronoi polygon, i.e. $\acute{P}_i = P_i + \beta \overrightarrow{V_e^i}$, where β is a given constant. An illustrative example of the EVF strategy is shown in Figure 3 for $\beta = \frac{1}{4}$. In this figure, the virtual forces applied from the edges to the sensor are shown by dotted vectors, and the displacement $\frac{1}{4}\overrightarrow{V_e^i}$ is depicted by a red vector.

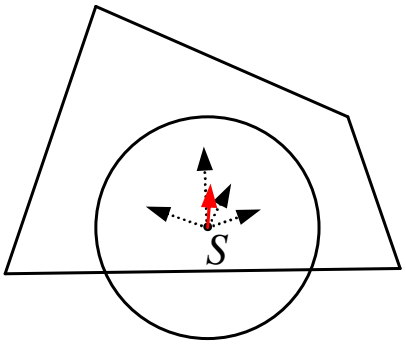

Fig. 3: An illustrative example of the EVF strategy.

C. Vertex-Edge Virtual Forces (VEVF) Strategy

The effectiveness of each of the two deployment strategies described so far depends on the relative position of sensors with respect to each other. One can take advantage of the strengths of both techniques, by developing a new technique as a combination of the VVF and EVF strategies. In this algorithm, which is referred to as the VEVF strategy, each sensor selects two points in every round, as its potential new position: one point based on the VVF method, and the other one based on the EVF technique. Any of the two points from which the sensor coverage improves the most is then selected as the new position for the sensor.

IV. Deployment Protocols for Nonidentical Sensors

When sensors have different sensing radii, a point that is not covered by the generating sensor of the polygon containing the point, may be covered by a neighboring sensor. This means that for a network of mobile sensors with different sensing radii, the conventional Voronoi diagram cannot be as useful for the development and analysis of sensor deployment strategies. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is used in such networks, as described in the next subsection.

A. MW-Voronoi Diagram

Consider a set of n distinct weighted nodes $(S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)$ in a 2D field, where $w_i > 0$ is the weighting factor of the node S_i, $i \in \mathbf{n}$. Let this set be denoted by $\mathbf{S}$. Let also the weighted distance of a point q from the node (S_i, w_i) be defined as:

$$d_w(q, S_i) = \frac{d(q, S_i)}{w_i}$$

where $d(q, S_i)$ is the Euclidean distance between the point q and the node S_i.

Partition the field into n regions, where each region contains only one node, which is the closest node in terms of weighted distance, to any point within that region. The diagram resulted from this partitioning is referred to as the *multiplicatively weighted Voronoi* (MW-Voronoi) diagram [29]. Each region $\hat{\Pi}_i$ in this diagram can be described as:

$$\hat{\Pi}_i = \left\{ q \in \mathbb{R}^2 \mid d_w(q, S_i) \leq d_w(q, S_j), \forall j \in \mathbf{n} - \{i\} \right\} \tag{1}$$

for any $i \in \mathbf{n}$.

Definition 3. Given two points A, B and a constant k, *Apollonian circle* of the segment AB is the locus of all points C satisfying the equality $\frac{AC}{BC} = k$, and is denoted by $\Omega_{AB,k}$ [32].

To obtain the MW-Voronoi region $\hat{\Pi}_i$, the Apollonian circles $\Omega_{S_i S_j, \frac{w_i}{w_j}}$ need to be obtained first for all $S_j \in \mathbf{S} \backslash \{S_i\}$. The smallest region created by these circles which contains node i is, in fact, the MW-Voronoi region $\hat{\Pi}_i$.

Consider now a group of nonidentical sensors in a field, and let each sensor be represented by a node whose weight is equal to the sensor's sensing radius. Draw the MW-Voronoi region for every sensor; the resulting diagram covers the whole field. Using the mathematical formulation of the MW-Voronoi diagram described by (1), it is straightforward to show that every sensor should only check its own MW-Voronoi region in order to identify the coverage holes of the whole network.

One can define the notions of coverage area, local coverage area, coverage hole and local coverage hole for the case of nonidentical sensors on noting that the Voronoi cells in this case are not polygons.

B. Deployment Protocols

Deployment algorithms similar to the ones developed in the previous section for a network of identical mobile sensors can also be developed for the case of a network of nonidentical sensors by using the MW-Voronoi partitions. Note that the boundaries of the regions in this case are parts of some Apollonian circles, and are not straight edges in general. Note also that an MW-Voronoi region will not have any vertex when the region is a circle, in which case the center of this circle is considered as the new location of the corresponding sensor. The corner points and boundary curves of an MW-Voronoi region

can be regarded as the vertices and edges of that region, respectively. Then, analogously to the VVF, EVF and VEVF strategies developed in the previous section, one can introduce the corner point virtual forces (CPVF), boundary curve virtual forces (BCVF), and point-curve virtual forces (PCVF) strategies, respectively.

Remark 1. The complexity of calculating a new sensor destination in all strategies introduced in this work is of order $O(m_i)$ or $O(e_i)$, where m_i and e_i are the number of vertices and edges of the i-th Voronoi polygon (or MW-Voronoi region), respectively. Since typically a Voronoi polygon (or MW-Voronoi region) does not have too many vertices and edges, the complexity of the proposed techniques for computing the new sensor destinations is usually not very high.

Theorem 1. *The proposed algorithms are convergent.*

Proof: Denote the positions and MW-Voronoi regions of the sensors in the k-th round by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$ and $\hat{\mathbf{\Pi}}(k) = \left\{\hat{\Pi}_1(k), \hat{\Pi}_2(k), \ldots, \hat{\Pi}_n(k)\right\}$, respectively. Denote also the total covered area of the field in the k-th round by $\beta(k)$. Some sensors will move and change their locations in the k-th round if the algorithm does not terminate in this round. Let sensor i be one of such sensors, and denote by $P_i(k+1)$ its location in the next round (note that $P_i(k+1) \neq P_i(k)$). According to Theorem 1 of [28], if the coverage area with respect to this location is larger than the local coverage area in the k-th step, then the overall coverage area increases in this round. In other words:

$$\beta^{P_i(k+1)}_{\hat{\Pi}_i(k)} > \beta^{P_i(k)}_{\hat{\Pi}_i(k)} \Longrightarrow \beta(k+1) > \beta(k) \tag{2}$$

The total coverage area, on the other hand, is upper-bounded by the entire area of the field. This means that the proposed algorithms are convergent. ■

It is to be noted that the proposed algorithms do not necessarily converge in finite time. As noted earlier, a prescribed threshold ϵ is used to terminate the algorithm in finite time. More precisely, the algorithm will continue after the k-th round if there is a sensor whose coverage improves by at least ϵ in this round, i.e.:

$$\exists i \in \mathbf{n} : \beta^{P_i(k+1)}_{\hat{\Pi}_i(k)} \geq \beta^{P_i(k)}_{\hat{\Pi}_i(k)} + \epsilon \tag{3}$$

The choice of ϵ involves a trade-off between the coverage performance and convergence rate of the

algorithm. On the one hand, a smaller ϵ would lead to a better network coverage. On the other hand, a larger ϵ yields a faster convergence. The following theorem provides an upper-bound on the number of rounds required to run the algorithm, as a function of ϵ.

Theorem 2. *Let a set* $\mathbf{S}$ *of* n *mobile sensors be deployed randomly in a flat plane. Using any of the algorithms proposed in this work with a prescribed coverage threshold* ϵ*, the number of rounds the algorithm iterates before it terminates does not exceed* $\frac{A_{total}}{\epsilon}$*, where* A_{total} *represents the entire area of the field.*

Proof: Denote by ζ_f the number of required rounds to satisfy the termination condition in any of the algorithms. Let the whole area of uncovered region in the k-th round be denoted by $\theta(k)$, and notice that $\beta(k) = A_{total} - \theta(k)$. Let also the locations of the sensors in this round be represented by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$, and denote by $\hat{\Pi}_i(k)$ the MW-Voronoi region associated with the i-th sensor, $i \in \mathbf{n}$. It follows from (1) that:

$$\theta(k) = \sum_{i=1}^{n} \theta_{\hat{\Pi}_i(k)}^{P_i(k)}, \ \forall k \in \{1, \ldots, \zeta_f\} \tag{4}$$

Define the *moving set* of the k-th round as the largest subset of sensors that change position in the k-th round, and let the set $\mathbf{Indx}(k)$ contain the indices of these sensors. Note that at least one sensor moves in the k-th round, i.e. $\mathbf{Indx}(k) \neq \emptyset$, $\forall k \in \{1, \ldots, \zeta_f\}$. Note also that the i-th sensor, $i \in \mathbf{Indx}(k)$, changes position in the k-th round if $\beta_{\hat{\Pi}_i(k)}^{P_i(k+1)} \geq \beta_{\hat{\Pi}_i(k)}^{P_i(k)} + \epsilon$. This implies that:

$$\theta_{\hat{\Pi}_i(k)}^{P_i(k+1)} \leq \theta_{\hat{\Pi}_i(k)}^{P_i(k)} - \epsilon, \quad \forall i \in \mathbf{Indx}(k) \tag{5}$$

On the other hand, part of the area $\theta_{\hat{\Pi}_i(k)}^{P_i(k+1)}$ might be covered by some other sensor positioned at $P_j(k+1)$, where $j \in \mathbf{n} \backslash \{i\}$. This means that:

$$\theta(k+1) \leq \sum_{i=1}^{n} \theta_{\hat{\Pi}_i(k)}^{P_i(k+1)} \tag{6}$$

From (5), (6) and on noting that the i-th sensor does not move if $i \in \mathbf{n} \backslash \mathbf{Indx}(k)$ (which means that

$\theta^{P_i(k+1)}_{\hat{\Pi}_i(k)} = \theta^{P_i(k)}_{\hat{\Pi}_i(k)}$), it can be concluded that:

$$\theta(k+1) \leq \sum_{i=1}^{n} \theta^{P_i(k)}_{\hat{\Pi}_i(k)} - |\mathbf{Indx}(k)|\,\epsilon \tag{7}$$

It follows from (4) and the above relation that:

$$\theta(k+1) \leq \theta(k) - |\mathbf{Indx}(k)|\,\epsilon \leq \theta(k) - \epsilon \tag{8}$$

or equivalently:

$$\beta(k+1) \geq \beta(k) + |\mathbf{Indx}(k)|\,\epsilon \geq \beta(k) + \epsilon \tag{9}$$

which means that under the proposed sensor deployment scheme the covered area of the sensors improves by at least ϵ in each round. Hence, the increased coverage from the initial round to the termination round is at least $\zeta_f \epsilon$. Now, since the entire covered area is less than or equal to A_{total}, it can be concluded that $A_{total} \geq \zeta_f \epsilon$ or equivalently $\frac{A_{total}}{\epsilon} \geq \zeta_f$. ■

Remark 2. One of the important properties of the MW-Voronoi diagram is that it partitions the field; in addition, there is exactly one sensor in each resultant region. Since under the proposed algorithms the new location of every sensor lies inside the MW-Voronoi region of the sensor and also each sensor moves within its region only, thus the sensors would not collide. However, if a sensor cannot communicate with some of its neighbors, then some boundaries of the resultant MW-Voronoi region (obtained according to the incomplete information) would be different from the exact ones. As a result, the MW-Voronoi regions might not partition the field in the sense that some of them might overlap with each other. This can negatively impact the detection of coverage holes, and may also lead to sensor collisions.

V. Simulation Results

It is well-known that no tractable analytical solution exists for the optimal coverage problem in the sensor networks [8], [33], [34]. The evaluation and comparison of different sensor deployment algorithms are usually performed by simulation. In this section, the performance of the proposed sensor deployment algorithms are evaluated and compared using several simulations with random initial sensor configurations. In the examples given below, the average results are depicted by performing 100 simulations with random initial configurations for Examples 1 and 2, and 20 simulations (also with

random initial configurations) for Example 3.

Example 1: This example compares the performance of the strategies introduced in Section III for different number of sensors: n=20, 30, 40, and 50. The sensors are deployed randomly in a flat square field of size $50\mathrm{m} \times 50\mathrm{m}$ and their sensing and communication ranges are $6\mathrm{m}$ and $20\mathrm{m}$, respectively. The algorithms used in this example are terminated when no sensor's coverage in its corresponding Voronoi polygon increases by more than 1% in the next move. In Figure 4, the final coverage factor (defined as the ratio of the covered area to the whole area of the field) is depicted under the three algorithms introduced in Section III, for different number of sensors. This figure shows that the coverage area under the VEVF strategy is greater than that under the VVF and EVF methods, for different number of sensors.

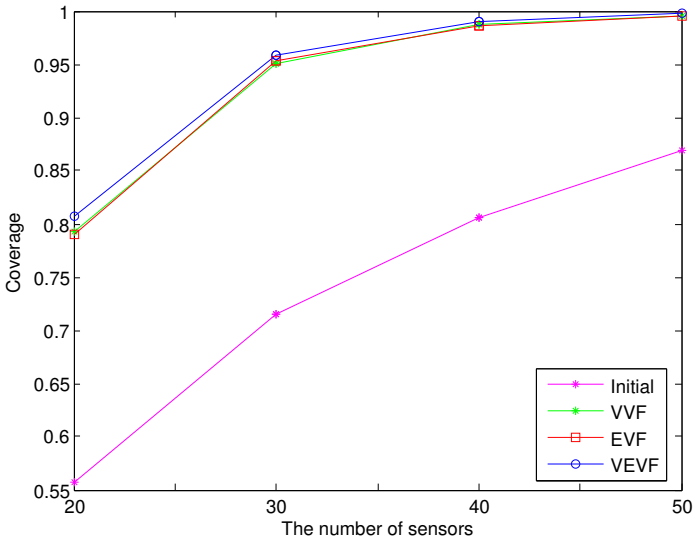

Fig. 4: Coverage factor for different number of sensors in Example 1 using the proposed algorithms.

Convergence rate is also an important issue in the performance assessment of sensor deployment techniques. Since the sensor deployment time in each round of different algorithms is more or less the same, the number of rounds the algorithm runs until the termination condition is met can be used to assess time efficiency. Figure 5 demonstrates that the number of times each algorithm runs in order to meet a prescribed termination condition decreases as the number of sensors increases between 30 and 50. This is because of the fact that for large number of sensors in the target field, the Voronoi polygons are small compared to the corresponding sensing circles. As a result, it is likely that each sensor covers a large portion of its Voronoi polygon; thus, the algorithm reaches the termination condition faster. The

number of rounds in the EVF procedure is relatively small, and hence it is more desirable in terms of convergence rate.

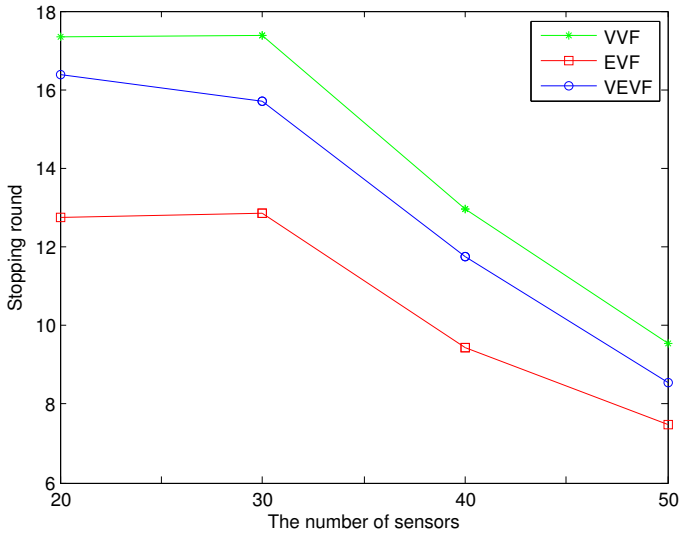

Fig. 5: The number of rounds required to meet the termination condition in Example 1 for different number of sensors using the proposed strategies.

The energy consumed by sensors in order to provide the desired coverage level is another important issue which needs to be taken into consideration when evaluating the efficiency of different algorithms. The energy consumption of the network is closely related to the distance the sensors travel as well as the number of times they stop before reaching their final position (note that once a sensor stops, it has to overcome the static friction in order to move again). Thus, to compare the energy-efficiency of the proposed methods, one should take into account the traveling distance of the sensors as well as the number of times they stop. Figure 6 provides the average traveling distance vs. the number of sensors for all three algorithms. This figure shows that the average traveling distance decreases with increasing the number of sensors in all three algorithms. This results from the fact that when the number of sensors is large, the Voronoi polygons are relatively small. This, in turn, decreases the distance between each sensor and its candidate destination (which is in the corresponding Voronoi polygon). Therefore, the average displacement distance of the sensors under all three strategies decreases. It is also observed from Figure 6 that the average displacement distance using the VVF method is less than that using the other two techniques.

Figure 7 gives the number of sensor movements for various number of sensors. The figure shows that

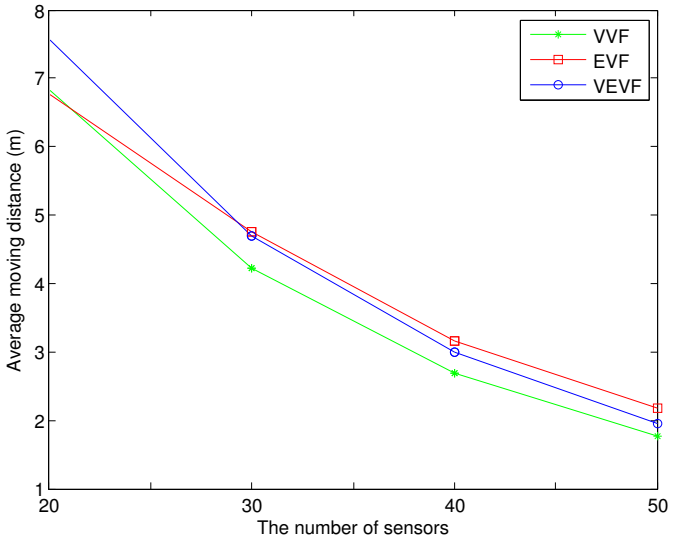

Fig. 6: The average travel distance for different number of sensors in Example 1, using the proposed algorithms.

the number of movements in all scenarios decreases by increasing the number of sensors. This results from the fact that when the number of sensors is large, the Voronoi polygons are small, and hence the sensors are likely to cover a large area of their Voronoi polygons. Thus, the termination condition will be met in a shorter time, which, in turn, decreases the number of sensor displacements. It can also be seen form Figure 7 that the smallest number of movements results from the EVF strategy.

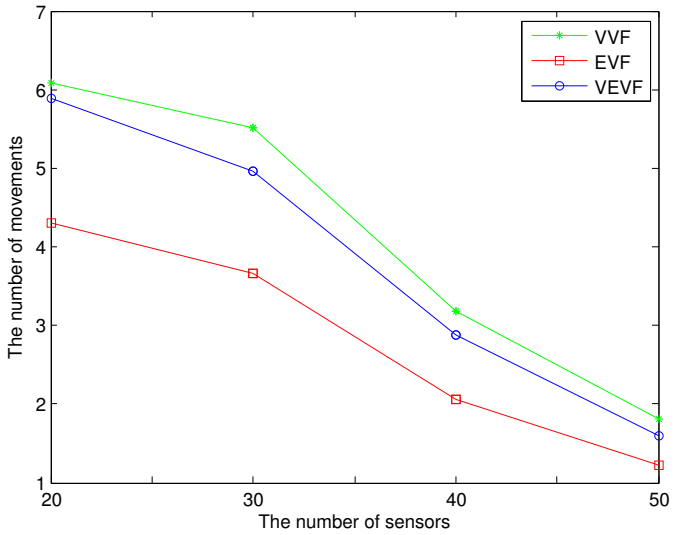

Fig. 7: The number of sensor displacements for different number of sensors in Example 1, using the proposed deployment strategies.

Let the energy that a sensor spends to travel 1m (with no stop) be 8.268J [35], [36]. Consider two cases, where the amount of energy spent to stop a sensor and then overcome the static friction following

a complete stop is the same as the energy that the sensor spends to travel 1m (first case) and 4m (second case) [8], [37]. Tables I and II give a summary of the energy consumption results for these two cases, and demonstrate that in both scenarios the EVF strategy performs better than the other two strategies as far as energy-efficiency is concerned.

TABLE I: The energy consumption in Joule for various number of sensors using the proposed strategies in the first case of Example 1.

	$n = 20$	$n = 30$	$n = 40$	$n = 50$
VVF	106.8229 J	80.3771 J	48.5951 J	29.5852 J
EVF	91.4222 J	69.6011 J	43.1367 J	28.1033 J
VEVF	111.2314 J	79.8485 J	48.6748 J	29.3468 J

Example 2: Consider 20 identical sensors with the sensing radius of 6m and the communication radius of 20m, randomly deployed in a field of size 50m by 50m. This example aims to evaluate the performance of the VEVF algorithm and compare it with some existing techniques, namely Minimax, VOR, VEC [8], Minmax-edge, Maxmin-edge, Maxmin-vertex, and VEDGE [28]. The coverage factor of the sensors in each round of different algorithms is provided in Figure 8. This figure indicates that the VEVF algorithm outperforms the other strategies in terms of sensor coverage. In addition, the complexity of finding the new destination of each sensor in the VEVF strategy is of order $O(m)$, while this complexity in the Minimax, Maxmin-edge, Maxmin-vertex, VEDGE and Minmax-edge techniques is of order $O(m^4)$, where m is the number of vertices of the Voronoi polygon.

Example 3: In this example, 27 sensors are deployed randomly in a field of size 50m by 50m: 15 sensors with a sensing range of 6m, 6 with a sensing range of 5m, 3 with a sensing range of 7m, and 3 with a sensing range of 9m. The communication radius of each sensor is assumed to be $10/3$ times larger than its sensing radius. The CPVF algorithm is now compared with six other techniques

TABLE II: The energy consumption in Joule for different number of sensors using the proposed algorithms in the second case of Example 1.

	$n = 20$	$n = 30$	$n = 40$	$n = 50$
VVF	257.8297 J	216.9892 J	127.5152 J	74.5747 J
EVF	198.1787 J	160.3506 J	94.1152 J	58.3443 J
VEVF	257.4014 J	202.7688 J	120.1662 J	68.8397 J

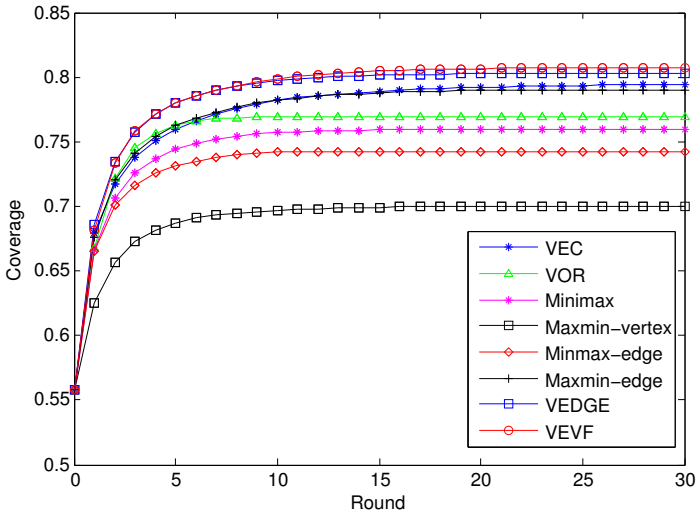

Fig. 8: Coverage factor in each round of the algorithms for 20 sensors in Example 2.

reported in the literature, namely WVB, FPB [38], Minmax-vertex, Maxmin-vertex [39], Minmax-curve and Maxmin-curve [10]. The coverage factor in each round of the algorithm is shown in Figure 9 for different strategies. This figure clearly confirms that the coverage factor obtained by using the CPVF algorithm is better than that obtained by using any other algorithm cited above. As in the previous example, the complexity of finding the new destination of each sensor in the CPVF strategy is of order $O(m)$, while this complexity in the Maxmin-vertex, Minmax-vertex, Maxmin-curve and Minmax-curve algorithms is of order $O(m^4)$.

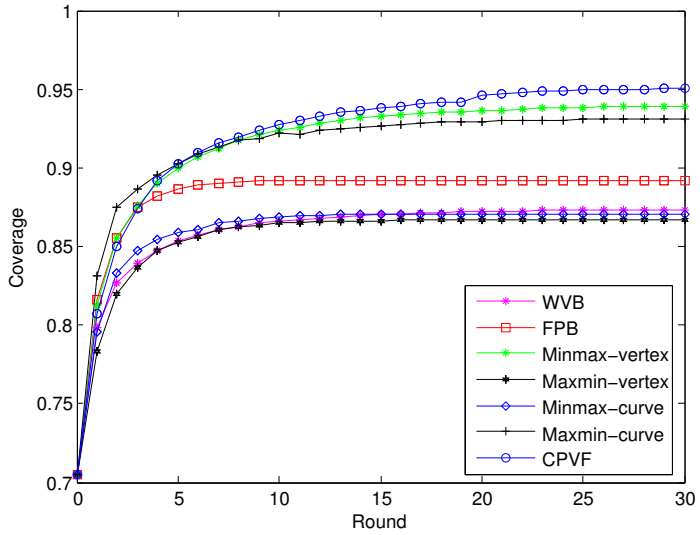

Fig. 9: Coverage factor in each round of different algorithms in Example 3.

VI. Conclusions

Three distributed sensor deployment strategies are developed in this work to improve coverage in a network of mobile sensors. The sensing field is first partitioned using the Voronoi diagram, and the deployment algorithms are developed based on the configuration of Voronoi polygons. The proposed strategies are iterative, where the next candidate position of any sensor is obtained in each iteration based on the distance of the sensor from the edges and vertices of its polygon. Different virtual forces are defined which are applied to the sensor from the vertices and boundaries of the polygon. Each sensor tends to move to a new location under the vector sum of these virtual forces, but it only moves to the new location if its coverage increases. The results are extended to the case of sensors with nonidentical sensing ranges using the notion of the multiplicatively-weighted Voronoi (MW-Voronoi) diagram. Simulations demonstrate the efficacy of the proposed algorithms in increasing the network coverage.

References

[1] S. Zhu and Z. Ding, "Distributed cooperative localization of wireless sensor networks with convex hull constraint," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2150–2161, 2011.

[2] S. Shin, T. Kwon, G.-Y. Jo, Y. Park, and H. Rhy, "An experimental study of hierarchical intrusion detection for wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 744–757, 2010.

[3] T.-C. Chen, T.-S. Chen, and P.-W. Wu, "On data collection using mobile robot in wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 41, no. 6, pp. 1213–1224, 2011.

[4] H.-C. Lin, Y.-C. Kan, and Y.-M. Hong, "The comprehensive gateway model for diverse environmental monitoring upon wireless sensor network," *IEEE Sensors Journal*, vol. 11, no. 5, pp. 1293–1303, 2011.

[5] O. Omeni, A. C. W. Wong, A. J. Burdett, and C. Toumazou, "Energy efficient medium access protocol for wireless medical body area sensor networks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 4, pp. 251–259, 2008.

[6] M. Tubaishat, Z. Peng, Q. Qi, and S. Yi, "Wireless sensor networks in intelligent transportation systems," *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 287–302, 2009.

[7] H. Mahboubi, A. Momeni, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "An efficient target monitoring scheme with controlled node mobility for sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1522–1532, 2012.

[8] G. Wang, G. Cao, and T. F. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[9] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in mobile sensor networks," in *Proceedings of IEEE Multiconference on Systems and Control*, 2011, pp. 1244–1249.

[10] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors," in *Proceedings of IEEE International Conference on Communications*, 2011, pp. 1–6.

[11] C.-Y. Chang, J.-P. Sheu, Y.-C. Chen, and S.-W. Chang, "An obstacle-free and power-efficient deployment algorithm for wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 39, no. 4, pp. 795–806, 2009.

[12] S. Narieda, "Lifetime extension of wireless sensor networks using probabilistic transmission control for distributed estimation," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1832–1838, 2012.

[13] A. Boukerche and X. Fei, "A Voronoi approach for coverage protocols in wireless sensor networks," in *Proceedings of IEEE Global Communications Conference*, 2007, pp. 5190–5194.

[14] A. Kwok and S. Martinez, "Unicycle coverage control via hybrid modeling," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 528–532, 2010.

[15] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimization and Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[16] A. Konstantinidis, K. Yang, and Q. Zhang, "An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks," in *Proceedings of IEEE Global Communications Conference*, 2008, pp. 475–480.

[17] J. Cortes and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2006.

[18] T. M. Cavalier, W. A. Conner, E. del Castillo, and S. I. Brown, "A heuristic algorithm for minimax sensor location in the plane," *European Journal of Operational Research*, vol. 183, no. 1, pp. 42–55, 2007.

[19] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[20] M. Pavone, E. Frazzoli, and F. Bullo, "Distributed policies for equitable partitioning: Theory and applications," in *Proceedings of 47th IEEE Conference on Decision and Control*, 2008, pp. 4191–4197.

[21] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proceedings of 47th IEEE Conference on Decision and Control*, 2008, pp. 3947–3952.

[22] R. Graham and J. Cortes, "Asymptotic optimality of multicenter voronoi configurations for random field estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 153–158, 2009.

[23] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, vol. 2005, 2005, pp. 2542–2547.

[24] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4-5, pp. 436–463, 2005.

[25] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[26] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, "Distributed seeking of nash equilibria in mobile sensor networks," in *Proceedings of the IEEE Conference on Decision and Control*, 2010, pp. 5598–5603.

[27] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2735–2750, 2010.

[28] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, to appear.

[29] E. Deza and M. M. Deza, *Encyclopedia of Distances*. Springer, 2009.

[30] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[31] H. Mahboubi and A. G. Aghdam, "Distributed deployment strategies to increase coverage in a network of wireless mobile sensors," in *Proceedings of American Control Conference*, 2013, pp. 5907–5912.

[32] A. V. Akopyan and A. A. Zaslavsky, *Geometry of Conics*. American Mathematical Society, 2007.

[33] J. Luo, D. Wang, and Q. Zhang, "On the double mobility problem for water surface coverage with mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 146–159, 2012.

[34] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, "Strictly localized sensor self-deployment for optimal focused coverage," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1520–1533, 2011.

[35] S. Yoon, O. Soysal, M. Demirbas, and C. Qiao, "Coordinated locomotion and monitoring using autonomous mobile sensor nodes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1742–1756, 2011.

[36] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 19–24.

[37] G. Wang, G. Cao, P. Berman, and T. F. L. Porta, "A bidding protocol for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 563–576, 2007.

[38] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing range," in *Proceedings of IEEE Global Communications Conference*, 2010, pp. 1–6.

[39] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors: Vertex-based approach," in *Proceedings of American Control Conference*, 2011, pp. 3227–3232.