Distributed Coverage Algorithms for a Network of Nonidentical Mobile Sensors

HAMID MAHBOUBI, Concordia University
KAVEH MOEZZI, Concordia University
AMIR G. AGHDAM, Concordia University
KAMRAN SAYRAFIAN-POUR, National Institute of Standards and Technology (NIST)

In this work, distributed deployment algorithms are proposed for efficient coverage in a mobile sensor network. The proposed algorithms calculate the position of the sensors iteratively based on existing coverage holes in the field. To this end, the multiplicatively weighted Voronoi (MW-Voronoi) diagram is used to partition the field, as it is assumed that the sensors have different sensing ranges. Under the proposed procedures, the sensors move in such a way that the coverage holes in the network are reduced. Simulation results are provided to demonstrate the effectiveness of the deployment schemes proposed in this paper.

1. INTRODUCTION

Wireless sensor networks have attracted considerable research interest in the past decade, and have found a broad range of applications in various areas [Wang et al. 2010], [Holland et al. 2011], [Zhu and Ding 2011], [Puccinelli and Haenggi 2010]. Recent advances in micro-electro-mechanical systems (MEMS) technology have enabled the design and manufacturing of small and low-cost sensor nodes. Examples of sensor network applications include biomedical engineering, security surveillance, target tracking and environmental monitoring, to name only a few [Moh et al. 2005], [Baumgartner et al. 2009], [Misra and Singh 2012], [Ingelrest et al. 2010]. In particular, a mobile sensor network (MSN) is comprised of a number of wireless nodes, where each node is capable of moving in different directions and communicating with a subset of sensors in order to achieve a global objective. Typical objectives in a mobile sensor network includes monitoring of a moving target [Mahboubi et al. 2012] and energy-efficient area coverage [Mahboubi et al. 2011]. In practice, to achieve the desired goals cooperatively, it is often preferable to use a decentralized decision-making scheme for sensor deployment [Howard et al. 2002]. Furthermore, the deployment strategy needs to be independent of the initial position of the sensors, as such information is usually unavailable [Clouqueur et al. 2003], [Wang et al. 2006]. In addition, the cost-effective resource management techniques are required to prolong the network lifetime [Dietrich and Dressler 2009], [Choi et al. 2012].

In [Cortes et al. 2004], [Cortes 2008], a mobile sensor deployment strategy is introduced to increase network coverage. In [Pavone et al. 2008], distributed control laws are presented to achieve convex equi-partition configuration in mobile sensor networks. An efficient procedure is introduced in [Graham and Cortes 2009] to move the sensors in such a way that the maximum error variance and extended prediction variance are minimized. Distributed control laws are provided in [Cortes and Bullo 2006] for the disk-covering and sphere-packing problems, using non-smooth gradient flows. An algorithm is proposed in [Susca et al. 2008] to monitor an environmental boundary with mobile agents, where the boundary is optimally approximated with a polygon. Distributed gradient-descent coverage algorithms are presented in [Kwok and Mar-
tinez 2010] using the Delaunay graph. In [Cortes et al. 2005], coordination algorithms are presented to increase coverage, and also a performance analysis using a class of aggregate objective functions is provided based on the geometry of the Voronoi partitions and proximity graphs. In [Bullo et al. 2009], [Cortes 2010], the problem of sensor deployment in a network with non-uniform coverage priority is considered. Decentralized control laws for the optimal placement of sensors are presented in [Martinez and Bullo 2006] for target tracking. A coverage strategy is proposed in [Boukerche and Fei 2007] based on a localized Voronoi diagram, where each sensor uses the local information of the neighboring sensors to construct its Voronoi region. In [Luo and Zhang 2008], a real-time coverage map is provided to compute the position of the sensors accordingly. Most of the existing results in sensor networks related to the coverage problem (including the above-mentioned papers) assume that all sensors have the same communication and also sensing ranges. While this can significantly simplify the deployment algorithm design, it may not be a realistic assumption in many real-world applications.

In this paper, three distributed deployment algorithms are presented for a network of nonidentical sensors. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is employed to find the coverage holes, where the weight assigned to each sensor is proportional to its sensing radius [Deza and Deza 2009], [Okabe et al. 2000]. Three proposed algorithms in this work are farthest point boundary (FPB), Maxmin-vertex and Minmax-vertex. These sensor placement algorithms are distributed and perform iteratively. Once each destination is computed, new local coverage area of the corresponding sensor (in the previously constructed MW-Voronoi region) is compared to its preceding local coverage area. If the new local coverage area is larger than the preceding one, the sensor moves to the new destination; otherwise, it remains in its current position. If the increase in the local coverage area of each sensor in an iteration does not exceed a certain threshold, the algorithm is terminated (to ensure a finite number of iterations).

The rest of the paper is organized as follows. In Section 2, some preliminaries and important notions and definitions are presented. Section 3 provides the new algorithms for efficient network coverage, as the main contribution of the paper. Simulations are given in Section 4, and finally the concluding remarks are summarized in Section 5.

2. BACKGROUND

2.1. Voronoi Diagram

Let $\mathbf{S}$ be a set of $n$ distinct nodes $S_1, S_2, \ldots, S_n$ in the 2D plane. The Voronoi diagram partitions the plane into a group of $n$ convex polygons, called Voronoi polygons, such that: (i) Each polygon contains only one node, referred to as the generating node of that polygon, and (ii) any point inside a polygon is closer to its generating node rather than to any other node in the plane [Klein 1989], [Okabe et al. 1992]. To construct the Voronoi polygon of a node, first the perpendicular bisectors of the segments connecting this node to its neighboring nodes are found. The smallest polygon (created by these perpendicular bisectors) containing this node is, in fact, the desired Voronoi polygon. The Voronoi diagram is a basic tool for the analysis and design of sensor deployment techniques.

Represent each sensor in the field as a node and sketch the Voronoi polygons for all sensors as described above, to cover the entire sensing field. Recall that by definition any point inside a polygon is closer to its generating sensor than to any other sensor in the field. Thus, assuming that all sensors have the same sensing capability (in terms of coverage radius), if a sensor cannot detect a certain point inside its corresponding poly-
Distributed Coverage Algorithms for a Network of Nonidentical Mobile Sensors

gon, that point cannot be detected by any other sensor either. This means that in order
to discover the “so-called” coverage holes in the entire network (i.e., the undetectable
points in the sensing field), it suffices that every sensor finds the points it cannot cover
in its own Voronoi region. However, as noted earlier, this fundamental statement is
only true for the case when all sensors have the same sensing range. It can be easily
shown that when the sensing radii of different sensors are not the same, then a point
which is not detected by the sensor corresponding to the polygon containing that point,
may be detectable by the sensor corresponding to a neighboring polygon. Hence, when
the sensors are not identical in terms of sensing radius, the conventional Voronoi dia-
gram is not as useful for effective sensor deployment in the network. The MW-Voronoi
diagram described in the next subsection is used to remedy this shortcoming.

2.2. MW-Voronoi Diagram

Let $S$ be a set of $n$ distinct weighted nodes in the plane denoted by $(S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)$, where $w_i > 0$ is the weighting factor associated with $S_i$, for any $i \in n := \{1, 2, \ldots, n\}$. It is desired now to partition the plane into $n$ regions such that:

— Each region contains only one node, called its generating node, and
— the nearest node, in the sense of weighted distance, to any point inside a region is
the generating node of that region.

The diagram obtained by the partitioning described above is called the \textit{multiplicatively weighted Voronoi diagram} (MW-Voronoi diagram) [Okabe et al. 2000]. Analogous to conventional Voronoi diagram, the mathematical characterization of each region obtained by the partitioning described above is as follows:

$$\Pi_i = \{ Q \in \mathbb{R}^2 \mid w_j d(Q, S_i) \leq w_i d(Q, S_j), \forall j \in n - \{i\}\}$$

(1)

for any $i \in n$, where $d(Q, S_i)$ is the Euclidean distance between $Q$ and $S_i$.

According to (1), any point $Q$ in the $i$-th MW-Voronoi region $\Pi_i$ has the following property:

$$\frac{d(Q, S_i)}{d(Q, S_j)} \leq \frac{w_i}{w_j}, \forall i \in n, \forall j \in n - \{i\}$$

(2)

Some of the following definitions and assumptions are borrowed from [Mahboubi et al. 2013].

\textbf{Definition 2.1.} A pair of sensors in an MW-Voronoi diagram whose MW-Voronoi regions share some boundary points are referred to as \textit{neighboring sensors} (this definition is similar to the one given in [Mahboubi et al. 2013] for the conventional Voronoi diagram). For any sensor $S_i$, $i \in n$, the set of neighboring sensors is denoted by $N_i$.

\textbf{Definition 2.2.} Consider the sensor $S_i$ with the sensing radius $r_i$ and the corresponding MW-Voronoi region $\Pi_i$, $i \in n$, and let $Q$ be an arbitrary point inside $\Pi_i$. The intersection of the region $\Pi_i$ and a circle of radius $r_i$ centered at $Q$ is referred to as the $i$-th \textit{coverage area w.r.t.} $Q$, and is denoted by $\beta^Q_{\Pi_i}$. The $i$-th coverage area w.r.t. the location of the sensor $S_i$ is called the $i$-th \textit{local coverage area} of that sensor, and is denoted by $\beta_{\Pi_i}$.

\textbf{Definition 2.3.} Consider an arbitrary point $Q$ inside the MW-Voronoi region $\Pi_i$, $i \in n$. The area inside the MW-Voronoi region $\Pi_i$ which lies outside the $i$-th coverage area w.r.t. $Q$ is referred to as the $i$-th \textit{coverage hole w.r.t.} $Q$, and is denoted by $\theta^Q_{\Pi_i}$. The $i$-th coverage hole w.r.t. the location of the sensor $S_i$ is called the $i$-th local coverage hole.
hole of that sensor, and is denoted by $\theta_\Pi$. Also, the union of all local coverage holes in the sensing field is referred to as the total coverage hole, and is denoted by $\theta$, i.e. $\theta = \sum_{i=1}^{n} \theta_\Pi$.

**Definition 2.4.** The Apollonian circle of the segment $AB$, denoted by $\Omega_{AB,k}$, is the locus of all points $E$ such that $\frac{AE}{BE} = k$ [Akopyan and Zaslavsky 2007].

To construct the $i$-th MW-Voronoi region, first the Apollonian circles of the neighboring partitions are found for the $i$-th sensor. In other words, the Apollonian circles $\Omega_{S_iS_j, \frac{w_i}{w_j}}$ are found for all $S_j \in N_i$. The smallest region created by these circles which contains the $i$-th node is, in fact, the $i$-th MW-Voronoi region (e.g., see Fig. 1). An example of an MW-Voronoi diagram with 20 sensors is sketched in Fig. 2.

![Fig. 1. The MW-Voronoi region for a sensor $S_1$ with four neighbors $S_2, \ldots, S_5$ [Mahboubi et al. 2010].](image)

The MW-Voronoi diagram is the main tool for sensor deployment in this paper. Each sensor has a sensing area which is a circle whose size is not necessarily the same for all sensors. Consider each sensor in the field as a node with a weight equal to its sensing radius, and sketch the MW-Voronoi region for every sensor. From the characterization of the MW-Voronoi regions provided in (1), it is straightforward to show that if a sensor cannot detect a phenomenon in its corresponding region, no other sensor can detect it either. This means that in order to find the coverage holes in the network, it would suffice to compare the MW-Voronoi region of each node with its local coverage area.

In the remainder of the paper, the MW-Voronoi diagram corresponding to a set of $n$ distinct weighted nodes $(P_1, r_1), (P_2, r_2), \ldots, (P_n, r_n)$, where $P_i$ and $r_i$ are the location and sensing radius of the $i$-th sensor, $i \in n$, will be simply referred to as the MW-Voronoi diagram.

**Notation 2.5.** Consider a circle of radius $r$ centered at $O$, denoted hereafter by $\Omega(O, r)$, and a point $V$ in the plane. The intersection of $\Omega$ and the extension of $VO$
Fig. 2. An example of the MW-Voronoi diagram for a group of 20 nonidentical sensors in a network.

from \( O \) is denoted by \( T_{\Omega(O,r)}^{V} \). The other intersection point of \( \Omega(O,r) \) and \( VO \) (or its extension) is denoted by \( T_{\Omega(O,r)}^{V} \).

Notation 2.6. As mentioned before, the boundary curves of an MW-Voronoi region are the segments of some Apollonian circles. The set of all such Apollonian circles for the \( i \)-th MW-Voronoi region is denoted by \( \Omega_i \). The sets \( \tilde{\Omega}_i \) and \( \tilde{\tilde{\Omega}}_i \) are defined as follows:

\[
\tilde{\Omega}_i = \{ \Omega \in \Omega_i | S_i \in \Omega \}
\]

\[
\tilde{\tilde{\Omega}}_i = \{ \Omega \in \Omega_i | S_i \notin \Omega \}
\]

Assumption 2.7. In this paper, it is assumed that there is no obstacle in the field. Therefore, the sensors can move to any desired location without obstacle avoidance concerns using existing techniques, e.g. [Koditschek 1989], [Lengyel et al. 1990], [Li et al. 2003].

Assumption 2.8. The sensors are capable of localizing themselves with sufficient accuracy in the sensing field (using, for instance, the methods proposed in [Niculescu and Nath 2003], [Intanagonwiwat et al. 2000]).

Assumption 2.9. It is assumed that the graph representing sensors’ communication topology is connected [Kwok and Martinez 2011]. Hence, each sensor can obtain the information about the sensing radii and locations of the other sensors (and in particular its neighbors) through proper communication routes, and consequently calculate its MW-Voronoi region accurately. Note that this is a realistic assumption as the number of sensors in a mobile sensor network is typically large (or more precisely, there is a sufficient number of sensors per area unit) [Wang and Tseng 2008], [Yoon et al. 2011].

3. DEPLOYMENT PROTOCOLS

In this section, three different protocols are developed for a distributed sensor network. The proposed algorithms are iterative, where in each iteration every sensor \( S_i \),
$i \in \mathbb{n}$, first broadcasts its sensing radius $r_i$ and location $P_i$ to other sensors, and then constructs its MW-Voronoi region $\Pi_i$ based on the information it receives from them. It checks the region subsequently to detect the possible coverage holes. If any coverage hole exists, the sensor calculates its target location (but does not move there) in such a way that by moving there the coverage hole would be eliminated, or at least its size would be reduced by a certain threshold. Once the new target location $\hat{P}_i$ is calculated, the coverage area w.r.t. this location, i.e., $\beta_{\Pi_i}^{\hat{P}_i}$ is obtained. If this coverage area is greater than the previous local coverage area, i.e. $\beta_{\Pi_i}^{\hat{P}_i} > \beta_{\Pi_i}^{P_i}$, the sensor moves to the new location; otherwise it remains in its current position. In order to terminate the algorithm in finite time, a proper coverage improvement threshold $\epsilon$ is defined such that if the increase in the local coverage area by each sensor is not sufficiently large (as specified by $\epsilon$), there is no need to continue the iterations.

As noted above, one of the important characteristics of the sensor deployment strategies proposed in this paper is that each sensor moves to its new destination point only if its coverage area w.r.t. the new location in the old MW-Voronoi region increases. The following theorem is similar to Theorem 1 of [Mahboubi et al. 2013], and shows that the total coverage increases under the proposed algorithms.

**Theorem 3.1.** Consider the set $S$ of $n$ sensors in the plane, and let their positions and sensing radii be denoted by $P = \{P_1, P_2, \ldots, P_n\}$ and $r = \{r_1, r_2, \ldots, r_n\}$, respectively, with the corresponding MW-Voronoi regions $\Pi_1, \Pi_2, \ldots, \Pi_n$. Assume the sensors move to new positions $\hat{P} = \{\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_n\}$ with the corresponding MW-Voronoi regions $\hat{\Pi}_1, \hat{\Pi}_2, \ldots, \hat{\Pi}_n$ such that $\hat{P}_i \neq P_i$ for all $i \in K$, where $K$ is a non-empty subset of $n$. If the $i$-th coverage area w.r.t. $\hat{P}_i$ in the previously constructed MW-Voronoi region $\Pi_i$ is greater than the $i$-th local coverage area in $\Pi_i$ (i.e., $\beta_{\Pi_i}^{\hat{P}_i} > \beta_{\Pi_i}^{P_i}$) for all $i \in K$, then the total coverage in the network increases.

**Proof.** The proof is similar to that of Theorem 1 in [Mahboubi et al. 2013].

**Notation 3.2.** In the remainder of this paper, $\mathcal{V}$ represents an MW-Voronoi diagram with $n$ regions (each one corresponding to a sensor). Furthermore, the number of corners of the $i$-th region is denoted by $m_i$, for any $i \in n$.

In certain cases the vertex-based algorithms can outperform the edge-based ones, as far as coverage is concerned. For example, in Fig. 3 the target location for sensor $S$ in the VOR [Wang et al. 2006], Minimax [Wang et al. 2006], and Maxmin-edge [Mahboubi et al. 2013] strategies are the points $A$, $B$ and $D$, respectively. It is clear in this case that the VOR and Minimax algorithms increase the coverage area, but the Maxmin-edge algorithm does not. This motivates the development of new algorithms for networks of sensors with nonidentical sensing ranges based on the distances of the sensor and the points inside its corresponding region from the vertices of the MW-Voronoi region.

### 3.1. Farthest Point Boundary Strategy (FPB)

In this algorithm, each sensor moves toward the farthest point in its MW-Voronoi region such that any existing coverage hole in its region can be covered. This point is denoted by $X_{i,\text{far}}$ for the $i$-th region. In fact, once a sensor detects a coverage hole, it calculates the farthest point (using the information about its MW-Voronoi region as well as the coverage holes in that region, as it will be shown later) and moves toward it continuously until $X_{i,\text{far}}$ is covered. The following definition is used to calculate the farthest point in each MW-Voronoi region.
Definition 3.3. The corner points of the $i$-th MW-Voronoi region (i.e., the intersection of its boundary curves) are denoted by $V_{i1}, V_{i2}, \ldots, V_{im}$. These points will hereafter be referred to as the MW-Voronoi vertices for the $i$-th MW-Voronoi region (note that a region may have no vertex). It is to be noted that the farthest point in each MW-Voronoi region lies on the boundary of the region.

Lemma 3.4. Let $E$ and $F$ be two points on the circle $\Omega$, and $V$ be an arbitrary point in the plane such that $T^V_\Omega$ is closer to $E$ than to $F$ (see Fig. 4). Then, $VE > VF$.

PROOF. From the law of cosines in triangles $VOE$ and $VOF$, it results that:

$$VE^2 = VO^2 + OE^2 - 2VO \times OE \times \cos \angle VOE$$  \hspace{1cm} (3)

$$VF^2 = VO^2 + OF^2 - 2VO \times OF \times \cos \angle VOF$$  \hspace{1cm} (4)

Since $0 \leq \angle VOF < \angle VOE \leq 180$, hence $\cos \angle VOE < \cos \angle VOF$. From (3), (4) as well as the relations $OE = OF$ and $\cos \angle VOE < \cos \angle VOF$ it can be concluded that $VE > VF$. This completes the proof. \hfill \Box

Lemma 3.5. Given a positive constant $k \neq 1$, let $E$ and $F$ be two points on $\Omega_{AB,k}$ such that $T^A_{\Omega_{AB,k}}$ is closer to $E$ than to $F$ (see Fig. 5). Then, $AE > AF$ and $BE > BF$. 

Fig. 3. An example of a configuration for which the edge-based strategies are not as effective.

Fig. 4. An illustrative figure for Lemma 3.4.
Proof. The proof follows immediately from Lemma 3.4, on noting that \( \frac{AE}{BE} = \frac{AF}{BF} = k \).

Remark 3.6. It is implied from Lemma 3.5 that for any positive constant \( k \neq 1 \), \( T_{\Omega_{AB,k}}^A \) is the farthest point to \( A \) and \( B \), and that \( T_{\Omega_{AB,k}}^A \) is the nearest point to \( A \) and \( B \), among all points on \( \Omega_{AB,k} \). For convenience of notation, \( T_{\Omega_{AB,k}}^A \) and \( T_{\Omega_{AB,k}}^A \) will hereafter be denoted by \( T_{AB,k} \) and \( T_{AB,k} \), respectively.

Lemma 3.7. Let \( D \) be a point and \( AB \) be a segment in the plane. Among all points on \( AB \), the farthest point from \( D \) is either \( A \) or \( B \).

Proof. The proof is straightforward and is omitted here.

Theorem 3.8. Let \( A_i \) be the set of all vertices for the \( i \)-th region \( (i \in n) \) of the MW-Voronoi diagram \( V \), and define the set \( B_i \) as follows:

\[
B_i = \left\{ T_{S_iS_j,k} \mid k = \frac{w_i}{w_j}, 1 \leq j \leq n, S_j \in N_i \right\}
\]

where \( N_i \) is the set of all neighbors of the \( i \)-th sensor. Then the farthest point in the \( i \)-th region belongs to the union of the sets \( A_i \) and \( B_i \); i.e., \( X_{i,\text{far}} \in A_i \cup B_i \).

Proof. As noted earlier, \( X_{i,\text{far}} \) lies on the boundary of the \( i \)-th region. Consider the following two cases:

Case 1: \( X_{i,\text{far}} \) is on the boundary curve \( V_{i1}V_{i2} \) such that \( V_{i1}V_{i2} \in \Omega_{S_iS_j,\frac{w_i}{w_j}} \). If \( T_{S_iS_j,\frac{w_i}{w_j}} \) is on the boundary curve \( V_{i1}V_{i2} \), then according to Remark 3.6, \( X_{i,\text{far}} \in B_i \); otherwise, since among all points on the boundary curve \( V_{i1}V_{i2} \) either \( V_{i1} \) or \( V_{i2} \) is the nearest point to \( T_{S_iS_j,\frac{w_i}{w_j}} \), hence according to Lemma 3.5 \( X_{i,\text{far}} \) is equal to either \( V_{i1} \) or \( V_{i2} \).

This means that \( X_{i,\text{far}} \in A_i \).

Case 2: \( X_{i,\text{far}} \) is on the boundary segment \( V_{i3}V_{i4} \). In this case, it follows from Lemma 3.7 that \( X_{i,\text{far}} \in A_i \).

Therefore, in both cases considered above \( X_{i,\text{far}} \in A_i \cup B_i \).

From Theorem 3.8 and on noting that the number of vertices (or boundary curves) of the \( i \)-th region is equal to the number of the \( i \)-th sensor’s neighbors (i.e., \( m_i = \text{dim}(N_i) \)), one can develop the following algorithm of complexity \( O(m_i) \) to calculate the farthest point in the \( i \)-th MW-Voronoi region.
ALGORITHM 1: An algorithm for finding the farthest point in the $i$-th MW-Voronoi region

\begin{algorithm}
\begin{enumerate}
\item for all $S_j \in N_i$
  \begin{enumerate}
  \item calculate $T_{S_iS_j}$, $k$.
  \item if $T_{S_iS_j}$ lies on the boundary of the $i$-th MW-Voronoi region, then
    \begin{enumerate}
    \item record it.
    \end{enumerate}
  \end{enumerate}
\end{enumerate}
\item for $j = 1, 2, \ldots, m_i$
  \begin{enumerate}
  \item record $V_{ij}$.
  \end{enumerate}
\end{enumerate}
\begin{enumerate}
\item The point whose distance from $S_i$ is maximum is the farthest point in the $i$-th MW-Voronoi region.
\end{enumerate}
\end{algorithm}

Since typically an MW-Voronoi region does not have “too many” vertices, the computational complexity of calculating the farthest point is usually not very high.

Fig. 6 shows an operational example of FPB Algorithm. In this example, 27 sensors are randomly deployed in a $50m \times 50m$ flat space: 15 with a sensing radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and 3 with a sensing radius of 9m. Moreover, the communication range of each sensor is assumed to be $10/3$ times greater than its sensing range. In this figure, three snapshots are provided, and in each one both local coverage of the sensors (filled circles) and the MW-Voronoi regions are depicted. The initial coverage is 66.7%, but after the first round it is improved to 77.2%, and the final coverage is 92.9%. As it can be seen, after the final round the sensors are distributed more evenly than the initial deployment, and that the coverage increases considerably.

![Fig. 6](image)

(3.2. Maxmin-Vertex Strategy)
The idea behind the Maxmin-vertex strategy is that normally for a good coverage result, none of the sensors should be too close to any of its vertices. In this strategy, the destination for each sensor is selected as a point inside the corresponding MW-Voronoi region whose distance from the nearest vertex is maximized. This point will be referred
to as the **Maxmin-vertex centroid**, and will be denoted by $\bar{O}_i$ for the $i$-th MW-Voronoi region ($i \in \mathbb{n}$). Let the distance between this point and the nearest vertex to it in the $i$-th region be represented by $\bar{r}_i$. The Maxmin-vertex circle is defined next.

**Definition 3.9.** The Maxmin-vertex circle of a region in the MW-Voronoi diagram $\mathcal{V}$ is defined as the largest circle centered inside that region such that all of the vertices of the region are either outside the circle, or on it. This circle is, in fact, $\Omega(\bar{O}_i, \bar{r}_i)$ for the $i$-th region ($i \in \mathbb{n}$).

**Remark 3.10.** If an MW-Voronoi region has exactly one boundary curve, then this curve is a circle which is also the Maxmin-vertex circle in the Maxmin-vertex strategy.

**Lemma 3.11.** Suppose the $i$-th region ($i \in \mathbb{n}$) of the MW-Voronoi diagram $\mathcal{V}$ has more than one boundary curve. If the Maxmin-vertex circle passes through exactly one vertex, say $V_{i1}$, then $\bar{O}_i$ is $T_{\Omega}^V$ for some $\Omega \in \Omega_i$; otherwise, the Maxmin-vertex circle passes through at least two vertices.

**PROOF.** Let $\bar{V}_{i1}$ be the nearest vertex of the $i$-th MW-Voronoi region to $\bar{O}_i$, and define:

$$\hat{u} := \min_{V \in V_i - \{\bar{V}_{i1}\}} \{d(\bar{O}_i, V)\}, \quad i \in \mathbb{n} \quad (5)$$

where $V_i$ is the set of vertices of the $i$-th MW-Voronoi region in the MW-Voronoi diagram.

![Fig. 7. An example of the Maxmin-vertex circle, when it passes through exactly one vertex.](image)

Suppose $\bar{O}_i$ and $T_{\Omega}^V$ are disjoint for any $\Omega \in \Omega_i$. Suppose also that the Maxmin-vertex circle does not pass through any vertex other than $\bar{V}_{i1}$, and hence the parameter $\delta^* = (\hat{u} - \bar{r}_i)/2$ is strictly positive. There are two possible cases, as discussed below.

**Case 1: $\bar{O}_i$ is inside the $i$-th MW-Voronoi region.** Let $\hat{O}$ be a point inside the $i$-th MW-Voronoi region and on the line $\bar{V}_{i1}\bar{O}_i$, but closer to $\bar{O}_i$, such that the distance between $\bar{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*)$ (see Fig. 8(a)).

**Case 2: $\bar{O}_i$ is on the boundary of the $i$-th MW-Voronoi region.** Suppose $\bar{O}_i$ is on the curve $\epsilon$. Since $\bar{O}_i$ and $T_{\Omega}^V$ are distinct for any $\Omega \in \Omega_i$, one can choose a point $\hat{O}$ on $\epsilon$ such that $d(\hat{O}, \bar{V}_{i1}) > d(\bar{O}_i, \bar{V}_{i1})$ and the distance between $\bar{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*)$ (see Fig. 8(b)).

In both cases, according to the triangle inequality:

$$d(\hat{O}, V) \geq d(\bar{O}_i, V) - \delta \geq \hat{u} - \delta, \quad \forall V \in V_i - \{\bar{V}_{i1}\} \quad (6)$$
From the above relation and on nothing that \( \hat{u} - \delta \geq \bar{r}_i + \delta > \bar{r}_i \) and \( d(\bar{O}, \bar{V}_{i1}) > d(\bar{O}, \bar{V}_{i1}) \), it can be concluded that

\[
\min_{V \in V_i} \{d(\bar{O}, V)\} > \bar{r}_i, \ i \in n
\]

which contradicts the initial assumption that \( \bar{O}_i \) is the Maxmin-vertex centroid. This means that there is at least one more vertex on the Maxmin-vertex circle. \( \square \)

\[\text{Fig. 8. The Maxmin-vertex centroid, when it is: (a) inside an MW-Voronoi region, and (b) on the boundary of an MW-Voronoi region.}\]

**Lemma 3.12.** Consider an MW-Voronoi diagram \( V_i \), and assume that the Maxmin-vertex circle of one of the regions, say region \( i \) (\( i \in n \)), passes through exactly two vertices, say \( \bar{V}_{i1} \) and \( \bar{V}_{i2} \). Then \( \bar{O}_i \) is the intersection point of the perpendicular bisector of \( \bar{V}_{i1} \bar{V}_{i2} \) and the boundary of the \( i \)-th MW-Voronoi region.

**Proof.** Suppose \( \bar{O}_i \) is not the intersection point of the perpendicular bisector of \( \bar{V}_{i1} \bar{V}_{i2} \) and the boundary of the \( i \)-th MW-Voronoi region, i.e., \( \bar{O}_i \) is inside the \( i \)-th region. Define:

\[
\hat{u} := \min_{V \in V_i - \{\bar{V}_{i1}, \bar{V}_{i2}\}} \{d(\bar{O}_i, V)\}, \ i \in n
\]

Since \( \Omega(\bar{O}_i, \bar{r}_i) \) passes through exactly two vertices, thus \( \delta^* = (\hat{u} - \bar{r}_i) / 2 \) is strictly positive. Let \( \bar{O} \) be a point on the perpendicular bisector of \( \bar{V}_{i1} \bar{V}_{i2} \) and outside the triangle \( \bar{V}_{i1} \bar{V}_{i2} \bar{O}_i \), but closer to \( \bar{O}_i \), such that the distance between the points \( \bar{O}_i \) and \( \bar{O} \) is equal to a given value \( \delta \in (0, \delta^*) \) (see Fig. 9). Using the triangle inequality, one can write:

\[
d(\bar{O}, V) \geq d(\bar{O}_i, V) - \delta \geq \hat{u} - \delta, \ \forall V \in V_i - \{\bar{V}_{i1}, \bar{V}_{i2}\}
\]

Using (9) along with the relations \( \hat{u} - \delta \geq \hat{u} - \delta^* = \bar{r}_i + \delta^* > \bar{r}_i \) and \( d(\bar{O}, \bar{V}_{i1}) = d(\bar{O}, \bar{V}_{i2}) >\]
Fig. 9. An illustrative figure used in the proof of Lemma 3.12.

If $\bar{r}_i$, one arrives at:

$$\min_{V \in V_i} \left\{ d(\tilde{O}, V) \right\} > \bar{r}_i, \quad i \in n$$

which contradicts the initial assumption that $\tilde{O}_i$ is the Maxmin-vertex centroid. This completes the proof. □

**Definition 3.13.** For convenience of notation, the circle passing through two vertices $V_p$ and $V_q$ of region $i$ in the MW-Voronoi diagram $V$, centered at the intersection of the perpendicular bisector of $V_pV_q$ and the boundary curve $V_vV_i$, is denoted by $\Omega_{k,l}^{p,q}$, $k, l, p, q \in m_i$. Also, the circle passing through three vertices $V_p, V_q$ and $V_r$ of region $i$ is denoted by $\Omega_{p,q,r}$, for $p, q, r \in m_i$. In addition, the circle passing through one vertex $V_r$ of MW-Voronoi region $i$, centered at $T_{\Omega}V_r$, is denoted by $\Theta_{V_r, \Omega}$, for any $r \in m_i$ and $\Omega \in \Omega_i$.

**Theorem 3.14.** Consider an MW-Voronoi diagram $V$, and suppose that the $i$-th region ($i \in n$) has more than one boundary curve. Let $\hat{C}_i$ and $\check{C}_i$ be the sets of all circles $\Omega_{p,q}^{k,l}$, $\forall k, l, p, q \in m_i$ and $\Theta_{V_r}^{\Omega}$, $\forall r \in m_i$, $\Omega \in \Omega_i$, respectively, whose centers are on the boundary of the $i$-th region, and do not enclose any of the vertices of this region. Let also $\check{C}_i$ be the set of all circumspheres of any three vertices, centered inside the $i$-th MW-Voronoi region or on its boundary, which do not enclose any of the vertices of this region. Define $C_i = \hat{C}_i \cup \check{C}_i \cup \check{C}_i$. Then the circle $\Omega(\tilde{O}_i, \bar{r}_i)$ belongs to $C_i$, and it is the largest circle in this set.

**Proof.** If $\Omega(\tilde{O}_i, \bar{r}_i) \notin \check{C}_i$, then according to Lemma 3.11 the Maxmin-vertex circle passes through at least two vertices. If it passes through exactly two vertices, say $V_1, V_2$, then according to Lemma 3.12, there exist $k, l \in m_i$ such that $\Omega(\tilde{O}_i, \bar{r}_i) = \Omega_{k,l}^{1,2}$. Hence, in this case $\Omega(\tilde{O}_i, \bar{r}_i) \in C_i$, and from Definition 3.9, $\tilde{r}_i = \max \{ r | \Omega(O, r) \in C_i \}$. If, on the other hand, the Maxmin-vertex circle passes through three or more Voronoi vertices, then it is the circumsphere of those vertices. Therefore, $\Omega(\tilde{O}_i, \bar{r}_i) \in C_i$, and again it is deduced from Definition 3.9 that $\tilde{r}_i = \max \{ r | \Omega(O, r) \in C_i \}$. □
Using the result of Theorem 3.14, the following algorithm of complexity $O(m_i^4)$ is developed to find the Maxmin-vertex centroid in the $i$-th MW-Voronoi region.

**ALGORITHM 2**: An algorithm for finding the Maxmin-vertex centroid in the $i$-th MW-Voronoi region

\begin{verbatim}
begin
   1) for $p = 1, 2, \ldots, m_i - 2$
      for $q = p + 1, p + 2, \ldots, m_i - 1$
         for $r = q + 1, q + 2, \ldots, m_i$
            calculate $\Omega_{p,q,r}$
            if $\Omega_{p,q,r}$ is centered inside the $i$-th MW-Voronoi region or on its boundary and does not
                enclose any of the vertices of the region, then
                record it.
            end
         end
      end
   end

   2) for $p = 1, 2, \ldots, m_i - 1$
      for $q = p + 1, p + 2, \ldots, m_i$
         calculate $\Omega_{p,q,i}$
         if $\Omega_{p,q,i}$ does not enclose any of the vertices of the $i$-th region, then
            record it.
         end
      end
   end

   3) for $r = 1, 2, \ldots, m_i$
      calculate $\Theta_{\Omega}$
      if $\Theta_{\Omega}$ does not enclose any of the vertices of the $i$-th region, then
         record it.
   end

   4) The center of the largest circle is the Maxmin-vertex centroid in the $i$-th MW-Voronoi region.
end
\end{verbatim}

As in the case of calculating the farthest point, since typically an MW-Voronoi region does not have "too many" vertices, the computational complexity for calculating the Maxmin-vertex centroid is usually not very high. Consider the initial deployment of Fig. 6(a), and this time let the Maxmin-vertex algorithm be used. After the first round, the coverage is improved to 75.5%, and finally it reaches 89.5% (see Fig. 10).

### 3.3. Minmax-Vertex Strategy

The idea behind the Minmax-vertex technique is that normally for optimal coverage, each sensor should not be "too far" from any of its Voronoi vertices. The Minmax-vertex strategy selects the target location for each sensor as a point inside the corresponding MW-Voronoi region whose distance from the farthest vertex is minimized. This point will be referred to as the Minmax-vertex centroid, and will be denoted by $\hat{O}_i$ for the $i$-th region ($i \in n$). Furthermore, the distance between this point and the farthest vertex from it in the $i$-th region will be represented by $\hat{r}_i$. The Minmax-vertex circle is defined next.
Definition 3.15. The Minmax-vertex circle of an MW-Voronoi region is defined as the smallest circle centered inside the region such that all of the vertices of the region are either inside the circle or on it. This circle is, in fact, $\Omega(\hat{O}, i)$, for the $i$-th region ($i \in n$).

Remark 3.16. If an MW-Voronoi region has exactly one boundary curve, then this curve is a circle which is also the Minmax-vertex circle for that region in the Minmax-vertex strategy.

Lemma 3.17. If an MW-Voronoi region has more than one boundary curve, then the corresponding Minmax-vertex circle passes through at least two vertices.

Proof. Let $\hat{V}_{i1}$ be the farthest vertex to $\hat{O}$ on the boundary of the $i$-th MW-Voronoi region, and define:

$$\hat{z} := \max_{V \in V_i \setminus \{\hat{V}_{i1}\}} \{d(\hat{O}, V)\}, \quad i \in n$$

(11)

Suppose that the Minmax-vertex circle does not pass through any vertex other than $\hat{V}_{i1}$, and hence $\delta^* = (\hat{r}_i - \hat{z})/2$ is strictly positive. There are two possible cases, as discussed below.

Case 1: $\hat{O}$ is inside the $i$-th MW-Voronoi region. Let $\hat{O}$ be a point inside the $i$-th MW-Voronoi region and on the line $\hat{V}_{i1}\hat{O}$, such that the distance between $\hat{O}$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*)$ (see Fig. 11(a)).

Case 2: $\hat{O}$ is on the boundary of the MW-Voronoi region. Suppose $\hat{O}$ is on the curve $\epsilon$. Let $\hat{O}$ be a point on $\epsilon$ or inside the $i$-th MW-Voronoi region such that $d(\hat{O}, \hat{V}_{i1}) < d(\hat{O}, \hat{V}_{i1})$, and the distance between $\hat{O}$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*)$ (see Fig. 11(b)).

In both cases, according to the triangle inequality:

$$d(\hat{O}, V) \leq d(\hat{O}, i) + \delta \leq \hat{z} + \delta, \quad \forall V \in V_i \setminus \{\hat{V}_{i1}\}$$

(12)

From the above relation and on noting that $\hat{z} + \delta \leq \hat{r}_i - \delta < \hat{r}_i$ and $d(\hat{O}, \hat{V}_{i1}) < d(\hat{O}, \hat{V}_{i1})$, it can be concluded that

$$\max_{V \in V_i} \{d(\hat{O}, V)\} < \hat{r}_i, \quad i \in n$$

(13)
Fig. 11. Minmax-vertex centroid, when it is: (a) inside an MW-Voronoi region, and (b) on the boundary of an MW-Voronoi region.

which contradicts the initial assumption that \( \hat{O}_i \) is the Minmax-vertex centroid. This means that there is at least one more vertex on the Minmax-vertex circle. □

Lemma 3.18. Consider an MW-Voronoi diagram \( \mathcal{V} \), and assume that the Minmax-vertex circle of one region, say region \( i \) (\( i \in n \)), passes through exactly two vertices, say \( \hat{V}_{i1} \) and \( \hat{V}_{i2} \). Then \( \hat{O}_i \) is the intersection point of the perpendicular bisector of \( \hat{V}_{i1} \hat{V}_{i2} \) and the boundary of the \( i \)-th MW-Voronoi region.

**Proof.** Suppose \( \hat{O}_i \) is not the intersection point of the perpendicular bisector of \( \hat{V}_{i1} \hat{V}_{i2} \) and the boundary of the \( i \)-th MW-Voronoi region, i.e., \( \hat{O}_i \) is inside the \( i \)-th region. Define:

\[
\hat{z} := \max_{V \in \mathcal{V}_i - \{\hat{V}_{i1}, \hat{V}_{i2}\}} \{d(\hat{O}_i, V)\}, \ i \in n
\]  

(14)

Since \( \Omega(\hat{O}_i, \hat{r}_i) \) passes through exactly two vertices, thus \( \delta^* = (\hat{r}_i - \hat{z})/2 \) is strictly positive. Let \( \tilde{O} \) be a point on the perpendicular bisector of \( \hat{V}_{i1} \hat{V}_{i2} \) and inside the triangle \( \hat{V}_{i1} \hat{V}_{i2} \tilde{O}_i \), but closer to \( \hat{O}_i \), such that the distance between the points \( \hat{O}_i \) and \( \tilde{O} \) is equal to a given value \( \delta \in (0, \delta^*) \) (see Fig. 12). Using the triangle inequality, one can write:

\[
d(\tilde{O}, V) \leq d(\tilde{O}, \hat{V}_{i1}) + \delta \leq \hat{z} + \delta, \ \forall V \in \mathcal{V}_i - \{\hat{V}_{i1}, \hat{V}_{i2}\}
\]  

(15)

Using (15) along with the relations \( \hat{z} + \delta \leq \hat{z} + \delta^* = \hat{r}_i - \delta^* < \hat{r}_i \) and \( d(\tilde{O}, \hat{V}_{i1}) = d(\tilde{O}, \hat{V}_{i2}) < \hat{r}_i \), one can conclude that:

\[
\max_{V \in \mathcal{V}_i} \{d(\tilde{O}, V)\} < \hat{r}_i, \ i \in n
\]  

(16)

which contradicts the initial assumption that \( \hat{O}_i \) is the Minmax-vertex centroid. This completes the proof. □
Theorem 3.19. Given an MW-Voronoi diagram $\mathcal{V}$, let $\hat{W}_i$ be the set of all circles $\Omega_{p,q}^{k,l}$, $\forall k, l, p, q \in m_i$, whose centers are on the boundary of the $i$-th region, and all vertices of the region are either inside or on them. Let also $\tilde{W}_i$ be the set of all circumcircles of any three vertices, centered inside or on the $i$-th region, with all vertices of the region either inside or on them. Define $W_i := \hat{W}_i \cup \tilde{W}_i$. Then the circle $\Omega(O_i, r_i)$ belongs to $W_i$, and it is the smallest circle in this set.

Proof. According to Lemma 3.17, the Minmax-vertex circle passes through at least two Voronoi vertices. If it passes through exactly two Voronoi vertices, say $V_{i1}, V_{i2}$, then according to Lemma 3.18, there exist $k, l \in m_i$ such that $\Omega(O_i, r_i) = \Omega_{k,l}^{1,2}$. Hence, in this case $\Omega(O_i, r_i) \in W_i$, and from Definition 3.15, $r_i = \min \{ r \mid \Omega(O, r) \in W_i \}$. If, on the other hand, the Minmax-vertex circle passes through three or more Voronoi vertices, then it is the circumcircle of those vertices. Therefore, $\Omega(O_i, r_i) \in W_i$, and again it is deduced from Definition 3.15 that $r_i = \min \{ r \mid \Omega(O, r) \in W_i \}$. □

Using the result of Theorem 3.19, the following algorithm of complexity $O(m_i^2)$ is developed to calculate the Minmax-vertex centroid of the $i$-th MW-Voronoi region.

As in the two methods presented earlier, since typically a Voronoi region does not have "too many" vertices, the computational complexity for calculating the Minmax-vertex centroid is normally not very high. Using this algorithm with the initial setting of Fig. 6(a), the coverage after the first round is improved to 79.9%, and it finally reaches 97.1% (see Fig. 13).

Theorem 3.20. The proposed algorithms (FPB, Minmax-vertex and Maxmin-vertex) are convergent.

Proof. Let the positions and sensing radii of the sensors in the $k$-th round be denoted by $P(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$ and $r(k) = \{r_1(k), r_2(k), \ldots, r_n(k)\}$, respectively. Denote also the MW-Voronoi regions in the $k$-th round by $\Pi_1(k), \Pi_2(k), \ldots, \Pi_n(k)$, and the corresponding total covered area of the field by $\beta(k)$. If the $k$-th round is not the final round, then some sensors move and change their locations in the next round. Assume that the $i$-th sensor, $i \in n$, moves to the new location $P_i(k+1) \neq P_i(k)$; if the coverage area w.r.t. this location is greater than the previous local coverage area, i.e. $\beta_{\Pi_i(k)} > \beta_{\Pi_i(k)}$, then according to Theorem 3.1 the total coverage in the network
ALGORITHM 3: An algorithm for finding the Minmax-vertex centroid in the \( i \)-th MW-Voronoi region

\[
\text{begin} \\
1) \text{for } p = 1, 2, \ldots, m_i - 2 \\
    \text{for } q = p + 1, p + 2, \ldots, m_i - 1 \\
    \text{for } r = q + 1, q + 2, \ldots, m_i \\
    \text{calculate } \Omega_{p,q,r}. \\
    \text{if } \Omega_{p,q,r} \text{ is centered inside the } i \text{-th MW-Voronoi region or on its boundary, and all} \\
    \text{the corresponding vertices are either inside the region or on its boundary, then} \\
    \text{record it.} \\
\text{end} \\
\text{end} \\
\text{end} \\
2) \text{for } p = 1, 2, \ldots, m_i - 1 \\
    \text{for } q = p + 1, p + 2, \ldots, m_i \\
    \text{calculate } \Omega_{p,q}^{k,l}. \\
    \text{if } \text{all vertices of the } i \text{-th region are either inside it or on the boundary of } \Omega_{p,q}^{k,l}, \text{then} \\
    \text{record it.} \\
\text{end} \\
\text{end} \\
\text{end} \\
3) \text{The center of the smallest circle is the Minmax-vertex centroid in the } i \text{-th} \\
\text{MW-Voronoi region.} \\
\text{end}
\]

Fig. 13. Snapshots of the execution of the movement of the sensors under the Minmax-vertex algorithm. 
(a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

increases in this round, i.e. \( \beta(k+1) > \beta(k) \). On the other hand, the total covered area
is upper-bounded by the overall area of the field, from which the convergence of the algorithms is implied. \( \square \)

It is worth mentioning that the convergence of the proposed algorithms may not
be achieved in finite time. As mentioned earlier, in order to terminate the algorithm
in finite time, a proper coverage improvement threshold \( \epsilon \) is defined such that the
algorithm will continue after the \( k \)-th round only if there is a sensor in the network
whose coverage increases at least by \( \epsilon \) in the following iteration, i.e. \( \exists i \in n : \beta_{i}(k+1) \geq \)
\( \beta_{P_i(k)} \) + \( \epsilon \). Note that the choice of \( \epsilon \) involves a trade-off between network coverage and deployment time. The following theorem provides an upper-bound on the number of rounds required to run the algorithm, as a function of \( \epsilon \).

**Theorem 3.21.** Consider a set of \( n \) mobile sensors \( S \), randomly deployed in a 2D field. Using any of the proposed algorithms with the coverage improvement threshold \( \epsilon \), the number of required rounds to run the algorithm is at most \( \frac{A_{total}}{\epsilon} \), where \( A_{total} \) is the overall area of the field.

**Proof.** Let the number of rounds required to run the algorithm in order to meet the termination condition be denoted by \( \zeta_f \). Let also the total uncovered area of the field in the \( k \)-th round be represented by \( \theta(k) \), and note that \( \beta(k) = A_{total} - \theta(k) \). Denote the position of the sensors and their corresponding MW Voronoi regions in the \( k \)-th round by \( P(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\} \) and \( \Pi_1(k), \Pi_2(k), \ldots, \Pi_n(k) \), respectively. From the properties of the MW-Voronoi diagram, one can conclude that:

\[
\theta(k) = \sum_{i=1}^{n} \theta_{P_i(k)}, \quad \forall 1 \leq k \leq \zeta_f \tag{17}
\]

Define the moving set of the \( k \)-th round as the largest subset of \( S \) that moves in the \( k \)-th round, and denote the indices of the sensors in this set by \( \text{Indx}(k) \). Note that at least one sensor moves in the \( k \)-th round, i.e. \( \text{Indx}(k) \neq \emptyset, \forall 1 \leq k \leq \zeta_f \). Note also that the \( i \)-th sensor, \( i \in \text{Indx}(k) \), moves in the \( k \)-th round if \( \beta_{P_i(k+1)} = \beta_{P_i(k)} + \epsilon \). This means that:

\[
\theta_{P_i(k+1)} \leq \theta_{P_i(k)} - \epsilon, \quad \forall i \in \text{Indx}(k) \tag{18}
\]

On the other hand, some of the points in \( \theta_{P_i(k+1)} \) might also be covered by another sensor located at \( P_j(k+1) \), for some \( j \in n \setminus \{i\} \). Hence:

\[
\theta(k+1) \leq \sum_{i=1}^{n} \theta_{P_i(k+1)} \tag{19}
\]

From the last two relations and on noting that for any \( i \in n \setminus \text{Indx}(k) \) the \( i \)-th sensor does not move (which implies \( \beta_{P_i(k+1)} = \beta_{P_i(k)} \)), one arrives at:

\[
\theta(k+1) \leq \sum_{i=1}^{n} \theta_{P_i(k)} - |\text{Indx}(k)| \epsilon \tag{20}
\]

It is now concluded from (17) and (20) that:

\[
\theta(k+1) \leq \theta(k) - |\text{Indx}(k)| \epsilon \leq \theta(k) - \epsilon \tag{21}
\]

or equivalently:

\[
\beta(k+1) \geq \beta(k) + |\text{Indx}(k)| \epsilon \geq \beta(k) + \epsilon \tag{22}
\]

which implies that using the underlying sensor relocation scheme, in each round the total covered area increases by at least \( \epsilon \). Therefore, the total amount of increased coverage from the first round to the termination round is greater than or equal to \( \zeta_f \epsilon \). Since the total covered area is always less than or equal to \( A_{total} \), hence \( A_{total} \geq \zeta_f \epsilon \) or equivalently \( \frac{A_{total}}{\epsilon} \geq \zeta_f \). \( \square \)

**Remark 3.22.** The importance of using the MW-Voronoi diagram for nonidentical sensors is that it guarantees the convergence of the proposed deployment algorithms.
Note that the monotonically increasing characteristic of the total covered area is guaranteed for the MW-Voronoi partitioning, and not necessarily for conventional Voronoi partitioning. This means that using existing sensor deployment strategies (which are mainly for identical sensors) may lead to non-convergent sensor movements if the sensors are not identical.

**Assumption 3.23.** It is implicitly assumed that a synchronization protocol (similar to the one in [Rentel and Kunz 2008]) is implemented to guarantee that all sensors start each round at the same time. Furthermore, the coverage rounds are assumed to be sufficiently long, so that each sensor can complete the process of calculating the new location and moving there (if necessary) in one round.

In the next section, it will be shown that all proposed algorithms result in a satisfactory final coverage. In addition, the performance of these algorithms will be compared in terms of the energy consumption of the sensors and deployment speed in reaching the desired coverage level for the sensing field.

### 4. Simulation Results

The three algorithms proposed in Section 3 are applied to a flat space of size $50\text{m} \times 50\text{m}$ in this section. In each simulation, the algorithm is terminated when none of the sensors’ coverage in its corresponding MW-Voronoi region is improved by more than $0.1\text{m}^2$ in the next move. The results presented in this section for field coverage are all the average values obtained by using 20 random initial deployments for the sensors.

Assume first the same 27 sensors of the example given in Section 3. The coverage factor (defined as the ratio of the covered area to the overall area) of the sensors in each round is depicted in Fig. 14 for the three algorithms proposed in this paper. It can be observed from this figure that all three strategies result in a satisfactory coverage level of the sensing field in the first few rounds of the corresponding algorithms. The resultant curves also show that the Minmax-vertex algorithm performs better than the other algorithms as far as coverage is concerned.

![Network coverage per round for 27 sensors.](image)

**Fig. 14.** Network coverage per round for 27 sensors.

It is desired now to compare the performance of the proposed algorithms in terms of the number of deployed sensors $n$. To this end, consider three more setups: $n=18$, 36, and 45 (in addition to $n=27$ discussed above). Let changes in the number of identical
sensors in the new setups be proportional to the changes in the total number of sensors (e.g., for the case of \(n=18\) there will be 10 sensors with a sensing radius of 6m, 4 with a sensing radius of 5m, 2 with a sensing radius of 7m, and 2 with a sensing radius of 9m). Fig. 15 provides the coverage results for different number of sensors. It can be seen from this figure that the network coverage in Minmax-vertex algorithm is larger than that in the other algorithms for different number of sensors.

![Coverage Chart](image)

Fig. 15. Network coverage for different number of sensors using the proposed algorithms.

The time it takes for the sensors to provide the desired coverage level is another important measure of the efficiency of the algorithms. Since the deployment time of the sensors in each round is almost equal for all algorithms, the number of rounds required for the sensors to reach a certain coverage level is used to evaluate the time efficiency. It is shown in Fig. 16 that in all three algorithms the number of rounds required to meet the termination condition specified earlier, increases by increasing the number of sensors up to a certain value (which varies for different algorithms), and then starts to decrease by adding more sensors. This is mainly because when there are a small number of sensors in the field, the MW-Voronoi regions are large in comparison with the corresponding sensing circles. Hence, there is a good chance that each sensor's local coverage area is completely inside its MW-Voronoi region, which means that the sensor does not need to move in order to increase its coverage area. On the other hand, when there are a large number of sensors in the field, there is a good chance that each sensor covers its MW-Voronoi region (and hence there are no coverage holes), which implies that the termination condition will be satisfied in a short period of time. It can be seen from Fig. 16 that in the Minmax-vertex algorithm the number of rounds required for the termination of the algorithm is larger than the other strategies. When the number of sensors in the field is not large, the number of rounds in the FPB algorithm is smaller than the other algorithms. Hence, the FPB algorithm is more efficient in such cases, as far as the deployment time is concerned. On the other hand, if there are a large number of sensors in the field, the Maxmin-vertex outperforms the other two algorithms in terms of deployment time.

Another important means of assessing the performance of the sensor deployment algorithms is the energy consumption of the sensors. The consumed movement energy of each sensor is known to be directly related to its traveling distance, as well as the
number of times it stops (the latter one is related to the static friction). Thus, to compare the proposed methods in terms of energy consumption, the traveling distance and the number of movements should be taken into consideration. Fig. 17 depicts the average moving distance for different number of sensors, using the three algorithms. This figure shows that by increasing the number of sensors, the average moving distance is decreased in all scenarios. This is due to the fact that the MW-Voronoi regions become smaller when the number of sensors increases. Note that a decrease in the size of an MW-Voronoi region translates to a smaller distance between the corresponding sensor and its destination point in that region. This in turn leads to a decrease in the average moving distance. It is shown in Fig. 17 that the traveling distance in the FPB algorithm is shorter than that in the other two algorithms. It can be seen from Fig. 17 that the average moving distances for all three algorithms are more or less the same when there are a large number of sensors in the field. The number of movements versus the number of sensors is depicted in Fig. 18, where it is shown that if the number of sensors increases from 18 to 27, the number of movements increases as well. It can also be observed from this figure that when the number of sensors increases beyond 36, the number of movements decreases. This is due to the fact that for large number of sensors the MW-Voronoi regions become smaller, and hence the sensors will likely cover their MW-Voronoi regions and will not need to move. As it can be observed from Fig. 18, when there are a relatively large number of sensors in the field, the number of movements in Maxmin-vertex algorithm is less than that in the other algorithms.

Let the required energy for traveling 1m (without stopping) be 8.268J (or 0.210J/inch) [Yoon et al. 2011], [Rahimi et al. 2003]. Consider two scenarios, where the energy required to stop a sensor and then overcome its static friction after a complete stop is equal to the energy required to continuously move the sensor 1m (first scenario) and 4m (second scenario) [Wang et al. 2006], [Wang et al. 2007]. Tables I and II provide a summary of the energy consumption results for these two cases. Define $\alpha$ as the ratio of energy consumption due to one stop followed by one move from complete stop to energy consumption due to one meter move. Now, if there are a large number of sensors in the field and the power required to overcome static friction of a sensor is much larger than that required to move it (per unit), the Maxmin-vertex algorithm outperforms the other two algorithms in terms of energy consumption. If, on the other
hand, the power required to overcome static friction of a sensor is much smaller than that required to move it, then regardless of the number of sensors the FPB algorithm performs better than the other two algorithms in terms of energy consumption.

Remark 4.1. Note that the algorithms introduced in this paper differ only in the way the new locations of the sensors are determined. As mentioned before, the complexity of the algorithm to find the new location of the $i$-th sensor in the FPB strategy is of order $O(m_i)$, while it is of order $O(m_i^4)$ in the Minmax-vertex and Maxmin-vertex algorithms. Hence, the FPB algorithm outperforms the other two algorithms as far as the computational complexity is concerned.

The above discussion is summarized below (logic and’s in these statements are capitalized):

(1) The Minmax-vertex algorithm is more preferable as far as network coverage is concerned.
Table I. The energy consumption in Joule for different number of sensors using the proposed algorithms for the case when the energy required to stop a sensor and then overcome static friction after a complete stop is equal to the energy required to move the sensor 1m non-stop.

<table>
<thead>
<tr>
<th>n</th>
<th>FPB</th>
<th>Minmax-vertex</th>
<th>Maxmin-vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>47.4234 J</td>
<td>77.8335 J</td>
<td>50.9095 J</td>
</tr>
<tr>
<td>27</td>
<td>44.7489 J</td>
<td>95.0093 J</td>
<td>56.8741 J</td>
</tr>
<tr>
<td>36</td>
<td>42.9542 J</td>
<td>68.6495 J</td>
<td>48.2570 J</td>
</tr>
<tr>
<td>45</td>
<td>28.1041 J</td>
<td>37.0676 J</td>
<td>27.3611 J</td>
</tr>
</tbody>
</table>

Table II. The energy consumption in Joule for different number of sensors using the proposed algorithms for the case when the energy required to stop a sensor and then overcome static friction after a complete stop is equal to the energy required to move the sensor 4m non-stop.

<table>
<thead>
<tr>
<th>n</th>
<th>FPB</th>
<th>Minmax-vertex</th>
<th>Maxmin-vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>83.8715 J</td>
<td>153.3546 J</td>
<td>83.8437 J</td>
</tr>
<tr>
<td>27</td>
<td>97.3876 J</td>
<td>232.9012 J</td>
<td>115.5310 J</td>
</tr>
<tr>
<td>36</td>
<td>105.4810 J</td>
<td>163.1114 J</td>
<td>105.1340 J</td>
</tr>
<tr>
<td>45</td>
<td>71.0150 J</td>
<td>84.8015 J</td>
<td>58.4763 J</td>
</tr>
</tbody>
</table>

(2) The Maxmin-vertex algorithm outperforms the other two algorithms when there are a large number of sensors in the field, AND:
— the deployment time is the main concern.
— the energy consumption is the main concern, AND the power required to overcome the static friction of a sensor is much larger than that required to move it (per unit).
(3) The FPB algorithm is more desirable when:
— the deployment time is the main concern AND the number of sensors in the field is not large.
— the energy consumption is the main concern, AND the power required to overcome the static friction of a sensor is much smaller than that required to move it (per unit).
— the computational complexity is concerned.

5. CONCLUSIONS

This paper presents efficient sensor deployment algorithms to improve coverage in mobile sensor networks. It is assumed that the sensing radii of different sensors are not the same. A multiplicatively weighted Voronoi (MW-Voronoi) diagram is then employed to develop three distributed deployment algorithms accordingly. Using these algorithms, the sensors move iteratively to reduce coverage holes in the sensing field. The algorithms proposed here take the general characteristics of an ideal sensor configuration into account (e.g., each sensor should not be too far or too close to any of the vertices of its corresponding MW-Voronoi region). Simulation results are pretested to compare the performance of the proposed approaches for different number of sensors.

REFERENCES


