A distributed self-organising node deployment algorithm for mobile sensor networks
Sadeghi Ghahroudi, Mahsa; Shahrabi, Alireza; Boutaleb, Tuleen

Published in:
International Journal of Communication Systems

DOI:
10.1002/dac.5309

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link to publication in ResearchOnline

Citation for published version (Harvard):
A distributed self-organising node deployment algorithm for mobile sensor networks

Mahsa Sadeghi Ghahroudi | Alireza Shahrabi | Tuleen Boutaleb

School of Computing, Engineering and Built Environment, Glasgow Caledonian University, Glasgow, UK

Correspondence
Alireza Shahrabi, Cowcaddens Road, Glasgow, United Kingdom, G4 0BA.
Email: a.shahrabi@gcu.ac.uk

Summary
Providing the appropriate coverage is quite essential for the effective functioning of many applications in Wireless Sensor Networks. Therefore, the efficiency of node deployment algorithms to supply the requested coverage is of high significance. In this paper, inspired by the equilibrium of molecules, a novel node deployment algorithm, called Smart Self-organising Node Deployment (SSND), is proposed to provide maximum coverage. Despite other proposed algorithms, which provide coverage based on the collective movement of nodes with massive energy consumption, SSND moves one sensor in every neighbourhood at each step to reduce the sensor nodes’ movement and hence the energy consumption. The chosen sensor nodes at each time step are distributedly determined by an eligibility function to reduce the non-essential movements while improving the accuracy of the reported locations of neighbours. Our extensive simulation study shows that SSND can achieve up to 30% coverage improvement compared to those of other algorithms in most scenarios and provides an adequate trade-off between coverage and energy consumption.

KEYWORDS
collective movements, distributed wireless sensor network, energy efficiency, self-organising, smart node deployment algorithm

1 | INTRODUCTION

Mobile Sensor Networks (MSNs) have recently attracted considerable attention and are widely used in various applications. Such applications range from environment monitoring to battlefield awareness and medical healthcare.1–3 One of the main challenges of these applications in MSNs is sufficient sensing coverage. The node deployment algorithms provide the coverage for these applications using resource-constrained sensor nodes that are placed in their proper locations by communicating their information through a network. However, sensor nodes are constrained by their resources in storage and processing capabilities due to their limitation in size, energy and power.4,5 Therefore, challenges in data collection, processing of the data and communication are significant in node deployment algorithms.

Most previous research use central deployment algorithms to locate the sensor nodes. In centralised algorithms, a node, i.e., sink, communicates and collects the sensors’ initial locations and propagates their target locations after a decision-making process.6,7 Although centralised algorithms provide required coverage, the single point of failure is not
Another type of node deployment algorithm, i.e., distributed algorithms, locates sensors in their proper locations with no centralised node. In a distributed node deployment algorithm, sensors need to exchange their information with their neighbours, i.e., sensor nodes in their communication range, to decide on their movements and final locations. The main advantage of the distributed node deployment algorithm is the scalability of the network in case of adding or removing sensors while reducing the chance of network failure.

The distributed node deployment algorithms have been widely reported in the literature from different angles. The main feature of all existing distributed node deployment is its iterative process. Every sensor node in a distributed node deployment algorithm runs an iterative algorithm at each time step to find its proper location in the Region of Interest (RoI) before it reaches its stable position. The final coverage of the area is obtained when all the sensor nodes in the RoI are stabilised. There are many research contributions to the field to enhance node deployment algorithms by improving the mathematical models and resource management. In the following, the most novel and pioneer ones are introduced under different groups.

Attributes of electromagnetic particles motivated three algorithms, vector-based (VEC), Voronoi-based (VOR) and Minimax. In these algorithms, every sensor node finds a part of the RoI, called the sensor node’s cell, where all the points in that area are closer to itself than other sensor nodes. As the points in each cell are closest to its associated sensor node and have no intersection with other cells, this sensor node is responsible for full coverage of its cell by finding a proper location within it. Moreover, the union of all cells create the Voronoi diagram, i.e., RoI. Therefore, the target area is fully covered when every cell is fully covered by its associated sensor node.

Although these algorithms provide sufficient coverage, they have shortcomings in some scenarios. Therefore, four other node deployment algorithms, the Maxmin-vertex, Maxmin-edge, Minimax-edge and VEDGE are introduced to improve the weaknesses of VEC, VOR and Minimax algorithms. The VEDGE algorithm, which is the combination of the Maxmin-edge algorithm and the Minimax algorithm, is proposed to increase the coverage in mobile sensor networks. Despite the VEC algorithm presented in Wang et al., the VEDGE algorithm can provide satisfying coverage for the relevantly large number of sensors in the network. Ultimately, the vertex-edge virtual forces (VEVF) algorithm is proposed that drives the sensor nodes by virtual forces. The VEFV algorithm moves sensor nodes in their Voronoi cell based on the virtual forces that have entered from the vertices and boundaries of their cells. Although VEFV is recognised as the most efficient Voronoi-based algorithm, it is not efficient in terms of adopting itself to the environmental conditions, like different initial deployment of sensor nodes, and its performance highly depends on the initial locations of sensors.

Another group of node deployment algorithms is inspired by the equilibrium of the molecules. In the equilibrium of molecules, every molecule reaches its balance and stable location after some time. The lowest energy point for every molecule, i.e., stable state, confirms almost the same distance from other neighbouring molecules. Therefore, any deployment of molecules in an area results in a uniform distribution when molecules reach their stable states. The Distributed Self-Spreading Algorithm (DSSA) uses the equilibrium of molecules concept to provide uniform distribution for sensor nodes to cover the requested area. In the DSSA, partial forces are employed from all neighbouring nodes to a sensor node to simulate the equilibrium of molecules concept and adjust the sensor nodes distance at every time step. Although DSSA provides sufficient coverage in some circumstances, the initial locations of sensor nodes highly affect the final coverage in DSSA. Moreover, a specific level of initial coverage is necessary for DSSA to operate. Furthermore, DSSA experiences a long transition from chaos to an order state.

Following the DSSA and under the same category, the Self-Organising node Deployment Algorithm (SODA) is proposed. The shortcomings of DSSA, such as initial deployment dependency, sensitivity to initial coverage, and long transition from chaos to order are all addressed in SODA. The partial force in DSSA is adjusted by effective parameters in SODA, which results in a final uniform deployment in many scenarios. The number of sensors in each neighbourhood, i.e., local density, and the expected density, i.e., the number of neighbouring sensor nodes in uniform distribution of sensors, are the effective parameters to adjust the partial force. These two parameters can present the local density and its difference from the expected density. The partial force in SODA considers the difference between the local density and the expected density to adjust, strengthen or weaken, its force. Therefore, sensitivity to the initial location of sensor nodes during the initial stages and the demanded minimum initial coverage are alleviated in SODA. Moreover, the final coverage is improved while the energy consumption is lower than that of DSSA. However, the energy cost in SODA is still high due to the collective movement of all the sensor nodes at each time step. Even worse, the collected information from neighbouring nodes is no more valid as they are all moving at every time step. Additionally, the collective movement of sensor nodes at every time step results in chaos that could lead to physical collisions and unnecessary movements of sensor nodes.
In this paper, a distributed node deployment algorithm called Smart Self-organising Node Deployment (SSND) for mobile sensor networks is proposed. The main aim of SSND, which is inspired by the equilibrium of molecules, is to achieve maximum coverage with less time cost in the deployment process while energy consumption is also optimised. The SSND algorithm addresses the collective and careless movement of sensor nodes by operating an eligibility function to move only one sensor node in every neighbourhood while collecting accurate location information from the neighbouring nodes. The smart movement of the sensor nodes can therefore lead to lower power consumption, total travelled distance and latency. Moreover, the final obtained coverage does not depend on the initial locations of sensor nodes. The preliminary version of this work was presented in a prior conference paper. The present paper extends the conference version with further details on collective movement algorithms. Moreover, the dependency of the SSND on initial coverage and communication range are evaluated with different groups of algorithms.

The rest of the paper is organised as follows. The model description is presented in Section 2, which is followed by the collective movement algorithms in Section 3. The SSND algorithm, including the motivations, movement criteria and other details of the SSND algorithm, is discussed in Section 4. Finally, the performance evaluation section to explain the simulation specification and results is under Section 5, while the paper has been concluded in Section 6.

2 | MODEL DESCRIPTION

A MSN consists of a set of sensor nodes, \( S = \{s_1, s_2, ..., s_n\} \), where \( s_i \in S, 1 \leq i \leq n, n \in \mathbb{N} \), represents a small, low-cost sensor node that is distributed in the RoI for sensing, collecting and reporting data of interest. Sensing model, network connectivity and localisation are some of the important sensor nodes’ properties in a node deployment algorithm when evaluating an algorithm. Moreover, the minimum number of sensor nodes to cover the target area is essential when comparing different node deployment algorithms. All these properties are discussed in the following sections.

2.1 Sensing, connectivity and localisation properties

Every sensor node senses any intruder within a specified territory, recognised as its sensing model. The sensing model of a sensor can be represented by a circle with a fixed radius of \( R_s \), i.e., disk model, or it can be any other irregular shape. In this paper, the sensing model of sensors is considered to be the disk model.

The communication between sensor nodes can be direct or through a sink node. The communication through a sink node created a centralised connection, while direct communication between sensors presents a distributed connection. In this paper, the direct connection with the communication range \( R_c \) is considered between sensor nodes.

For outdoor systems, the Global Positioning System (GPS) is the most common method for localisation. However, in indoor systems, many techniques such as the strength of the received signal or the time difference of arrival of two different signals are used to enable sensors to determine their locations. In this paper, we assume the sensor nodes know their locations in the area regardless of the environment.

2.2 Minimum number of required sensors in a target area

There are various ways to cover a square-shaped target area, with side \( a \) using identical sensors with radius \( r \). In Figure 1, an example of a node deployment is presented. It is shown that sensors fully cover the target area with some overlapping area, i.e., marked area, between neighbouring nodes. In order to fully cover a square region with \( a^2 \) area using \( n \) sensor nodes, the following inequality should be satisfied.

\[
a^2 \leq \sum_{i=1}^{n} \pi r^2
\]

(1)

Although the exact number of sensors is not required in many applications, a close estimation reduces the cost of the implementation. The circle nature of the sensors’ sensing range does not allow a fixed minimum required number of sensors for a target area. However, a proximity range of the required sensor nodes can be determined. In the
following, the lower and upper bounds for the minimum required number of sensors in a square with side \( a \) are presented by converting circles to closest squares.

### 2.2.1 Lower bound of the minimum number of sensor nodes

The circle that shows the sensing range of a sensor node can be estimated by a square that surrounds it. An inscribed circle with radius \( r \), like the one in Figure 2A, has a diameter equal to the side length of the square. In this scenario, the area of the circle is estimated equal to the area of the square with the side length of \( a_1 = 2r \), where the area of the square is \( a_1^2 = 4r^2 \). The estimated sensing area of a sensor node, i.e., circumscribed square, is bigger than the actual sensing area. Therefore, the number of these squares that fit into the RoI, \( a^2 \), represents the lower bound of the minimum number of required sensors. The lower bound demonstrates that it is impossible to cover the RoI with an equal or less number of sensors and is calculated by dividing the total area of the RoI by the area of this square as presented in

\[
\left\lfloor \frac{a^2}{4r^2} \right\rfloor < n.
\]
2.2.2 | Upper bound of the minimum number of sensor nodes

The circle that presents the sensing range of a sensor node can be approximated by a square that surrounds it. In Figure 2B, the largest square that is surrounded by a circle with radius $r$ is shown. The diagonal of the square is equal to the diameter of the circle. Therefore, the side length of this square is equal to $\frac{a}{2} = \frac{2r}{\sqrt{2}}$. The number of these squares that covers the RoI is the upper bound of the minimum number of sensors that is calculated by dividing the total area of the RoI by the area of this square. The upper bound guarantees the maximum of the minimum required number of sensors that are needed to fully cover the area, which is shown:

$$n \leq \frac{a^2}{2r^2}. \quad (3)$$

As mentioned earlier, the upper and lower bounds of the minimum number of the required sensors are the number of the inscribed and circumscribed squares that fit in the RoI. Therefore, the minimum number of required sensor node range can be presented as

$$\left\lfloor \frac{a^2}{4r^2} \right\rfloor < n \leq \frac{a^2}{2r^2}. \quad (4)$$

This range is used when configuring the simulation setup of the node deployments to provide realistic results when evaluating the algorithm.

3 | COLLECTIVE MOVEMENT ALGORITHMS

In distributed node deployment algorithms, every sensor decides about its movement independently. Therefore, without considering any priority in sensor nodes, all of them move to their desired locations at every time step before being stabilised. In most of the distributed node deployment algorithms, the independent movements of sensor nodes create a collective movement of sensor nodes at every time step $k$, $k \in \mathbb{N}$. The unnecessary movements of sensor nodes lead to wasting energy and also propagating some inaccurate information about the locations of the moving sensors to their neighbours. In this section, the limitations of the best three collective movement algorithms reported in the literature, namely, DSSA, SODA and VEVF, are discussed. The DSSA and SODA algorithms are inspired by the equilibrium of molecules, and VEVF is based on the Voronoi diagram. These three algorithms provide the required coverage with an acceptable level of power consumption and latency in comparison to other existing approaches; however, they are not applicable in some configurations. In the following, the common limitations of these algorithms are explained. Moreover, in the next sub-sections, the details of each algorithm’s limitations are separately discussed.

**Decision-making using obsolete information:** The most common limitation of the collective movement algorithms is the obsolete information a sensor node receives from its neighbouring nodes at every time step. In collective movement algorithms such as VEVF, DSSA and SODA, every sensor node receives its neighbouring nodes’ locations at the beginning of each time step and calculates its desired location based on the received information. Nevertheless, as almost all sensor nodes move to new locations before the end of this time step, the reported locations of sensor nodes are obsolete as they have already moved to other locations by the end of the time step. Therefore, the decision on the next location of every sensor node in these algorithms is not always the best decision.

**Initial deployment dependency:** The final obtained coverage using DSSA, SODA and VEVF algorithms is highly dependent on the initial deployment of sensor nodes in RoI. Therefore, changing the initial locations of sensors may lead to significantly different results in the final coverage. The different final coverage for a fixed number of sensor nodes makes these algorithms unpredictable. Moreover, the final achieved coverage is not reliable with specified resources, i.e., a particular number of sensor nodes.

3.1 | VEVF algorithm

The VEVF algorithm is a distributed node deployment algorithm that is based on the Voronoi diagram. In VEVF, every sensor node at every time step begins by creating a Voronoi cell based on its location and its neighbours’ locations.
Next, it scans its created cell for holes, calculates a position in its cell to cover that hole and moves towards that point. It is proven that if every sensor in the area can fully cover its own Voronoi cell, the whole Voronoi diagram, RoI, is then fully covered as well. Therefore, full coverage is obtained if every sensor node can accurately calculate the proper location and move towards it. In the VEVF algorithm, the new location of the sensor node, i.e., the proper location at every time step, is obtained by virtual forces that are applied from the vertices and boundaries of the sensor's Voronoi cell. Moreover, sensors only move to their new locations when the covered area of the current cell increases after this movement and terminate the algorithm if they have already fully covered their cell. The provided simulations of the VEVF represent a sufficient coverage in area, however, it is not all aware of the environment conditions and the decision-making process of this algorithm is based on obsolete information. In addition, the VEVF algorithm is sensitive to the sensor nodes' communication range. The details of these limitations are as follow.

**Lack of environmental awareness:** A sensor node in the VEVF algorithm checks its Voronoi cell in every iteration to cover the existing holes and terminates the algorithm when its Voronoi cell is fully covered or cannot increase the cell's coverage. Therefore, if a sensor node adequately covers its current cell, it can terminate the algorithm and stabilise itself in its location. On the other hand, the termination of each sensor node does not depend on other sensor nodes and does not happen at the same time step. Therefore, the Voronoi cell of a stabilised sensor node can change significantly or more holes are created through the process as the locations of its destabilised neighbouring nodes are changing. Although these changes may result in further expansion of the stabilised sensor's cell, it does not necessarily lead to further global coverage of the area, as the sensor is stabilised and not moving to increase its cell coverage. This is due to the fact that sensors only use their local information and do not take into account the potential negative impact of their decisions on their surrounding area. Therefore, the full coverage of the current cell is not an appropriate condition for a sensor to terminate the algorithm.

**Sensitivity to the communication range:** In some scenarios where one or more Voronoi neighbours of a sensor are out of its communication range, the calculated Voronoi cell of that sensor becomes incorrect. Moreover, in the sparse area where two or fewer sensor nodes exist in the communication range of a sensor, there are not enough sensor nodes to create a Voronoi cell. Although these limitations are ignorable in cases where the sensing range is much shorter than the communication range, the inaccurate created Voronoi cells decrease the accuracy of the Voronoi-based algorithms and increase the chance of collision in the area.

### 3.2 DSSA

The DSSA is a distributed node deployment algorithm that is inspired by the equilibrium of molecules. Every sensor node in the DSSA algorithm enters and receives forces, partial forces, to and from its neighbouring nodes at every time step, which is the same behaviour between molecules. Hence, sensor nodes are expected to be evenly distributed across the RoI at the final stage of DSSA, the same as those in the equilibrium state. The value of partial forces that are entered from neighbouring sensor nodes to at every time step depends on the location of , i.e., the distance between and its neighbouring sensors, and the local density, i.e., the number of the neighbouring sensor nodes. At the end of every time step, a total force that is the cumulative entered forces to from its one-hop neighbouring nodes moves the sensor to its desired location. Every sensor node, , moves at every time step independent to other sensor nodes inside or outside of its one-hop neighbourhood. This iterative process continues for every sensor node as long as the termination conditions are not satisfied. The DSSA can achieve the required coverage in most scenarios; however, it suffers from some limitations in some other configurations. In the following, some of these limitations are illustrated in detail.

**Non-single-point-deployment:** The partial force interacted between every sensor, , and its neighbour, , is calculated as presented in Heo and Varshney:}

\[
\begin{align*}
    f_{ij}^k &= \left| \frac{D}{\mu^2} \right| (R_c - |p_i^k - p_j^k|) \frac{p_j^k - p_i^k}{|p_j^k - p_i^k|},
\end{align*}
\]

where is the position of sensor at time step and \( D = \{ s_m \in S \mid |p_m^k - p_i^k| < R_c \} \) represents the set of neighbours, i.e., local density, at step while \( \mu \) represents the expected density. The communication and sensing range are the same for all the sensor nodes and are and , respectively. Therefore, the new location of the is calculated as 

\[
    p_i^{k+1} = p_i^k + \sum_{j \in D_i^k} f_{ij}^k.
\]
Consider a sensor node $s_i \in S$ at time step $k$ in a dense area. The local density of the $s_i$ at this time step is at its highest due to the high number of the neighbouring nodes, $D_k^i = \{s_1, s_2, ..., s_m\}$. On the other hand, sensor nodes are close to each other in a dense area, $\lim_{k \to \infty} |p_i - p_s^k| = 0$. Therefore, the $R_c - |p_i - p_s^k|$ section of the partial force between these two sensor nodes is becoming close to $R_c$, $\lim_{k \to \infty} R_c - |p_i - p_s^k| = R_c$. Consequently, as the distance between sensors decreases in a dense area, the $R_c - |p_i - p_s^k|$ part of the partial force is increasing to $R_c$ while the $|D_k^i|$ is at its highest, which leads to a very large partial force. The large partial forces that are resulted by dense deployment of sensor nodes move the sensor nodes far away and towards the edges of the area. Besides, the sudden large movements of sensor nodes at the same time break the nodes connectivity. Therefore, not only the single-point-initial-deployment that creates a dense area does not provide full coverage but also breaks the connectivity of network by forcing sensor nodes to move far away from each other.

**Long transition from chaos to order state:** The inaccurate partial forces in DSSA, result in many false movements, which initially cause chaos in the area. The chaos state eventually diminishes as sensor nodes decrease their total number of false movements. The reduction in the number of false movements is due to the gradual reduction of the total movements during the algorithm execution as nodes become stabled. In DSSA, the chaos state happens when the initial deployment is dense or when there are many false long movements by sensor nodes. Although the travelled distance by nodes at each time step is decreased close to the end of the algorithm until they are stabled, the delay in reaching the order state and the extra travelled distance are still considerable. Accordingly, the achieved full coverage in DSSA is costly in terms of time and energy.

### 3.3 | SODA

The SODA and DSSA are both inspired by the equilibrium of molecules. However, the partial force in SODA is adjusted based on the local density of every sensor node at every time step to control the partial force’s value based on the neighbouring nodes. Therefore, the main limitations of DSSA that are resulted from inaccurate partial forces are addressed. However, the improved delay and energy consumption are not yet affordable for many applications, mostly due to two reasons. First, SODA still suffers from the collective movement of sensors at every time step. The collective movement of sensor nodes reduces every sensor node’s energy level and lifetime at every time step. Moreover, neighbouring nodes report obsolete information about their current locations while they are moving at the same time. The propagation of obsolete information leads to false movements and high energy consumption. The redundant movement of sensor nodes can be eliminated by only moving a prior sensor node in every neighbourhood at each time step while significantly reducing the energy consumption. The lower number of movements in each neighbourhood can also reduce the propagation of obsolete information.

### 4 | SMART SELF-ORGANISING NODE DEPLOYMENT (SSND) ALGORITHM

In this section, the details of our distributed node deployment algorithm, Smart Self-organising Node Deployment (SSND) are discussed. First, the motivation to propose SSND and movement criteria are discussed. The details on how to declare a sensor node as a stabled node and hence, to stop running the algorithm are then followed. Also, the used notations in the algorithm are summarised in Table 1.

#### 4.1 | Motivation

The obsolete reported locations of neighbouring nodes, the non-essential collective movement of sensor nodes, sensitivity to communication range and the high energy consumption of the earlier mentioned algorithms have been the inspiration to propose a practical distributed node deployment algorithm, Smart Self-organising Node Deployment (SSND). Our proposed node deployment algorithm, SSND, is developed based on Coulomb’s law that is the foundation of the equilibrium of molecules. In Coulomb’s law, the electrostatic force between two electric charges is repulsive if the charges have the same sign and are attractive when the signs are different. The value of the force is proportional to the distance between charges. In the existence of molecules, the electrostatic force introduced by Coulomb’s law is exchanged...
between every two molecules leading to the equilibrium of molecules after some time. It is stated that molecules in their stable states have the same distance from each other and create a uniformly distributed placement.\textsuperscript{15}

One of the approaches to fully cover an RoI by distributed node deployment algorithm is distributing sensor nodes uniformly when enough sensor nodes exist in the RoI. In SSND, the uniform distribution of sensor nodes is modelled from the equilibrium of molecules phenomena, as particles in the equilibrium also achieve a uniform placement. Sensor nodes are replicated as electric charges, which interact with each other through a force called partial force. The value of partial force has an equivalent relation to sensor nodes’ distance, the same as those of DSSA and SODA.\textsuperscript{15,19} Although the partial force between sensors is the same in all three algorithms, the collective movement of sensors is diminished in SSND by selecting the appropriate sensor in every one-hop neighbourhood to move at every time step. The selected sensor that moves in that time step has the highest eligibility in its neighbourhood, which is calculated by an eligibility function. The reported eligibility for every sensor node is affected by two factors; local force and local density, which will be discussed in the following section.

### 4.2 Movement criteria

To address the blind and collective movements of sensor nodes, we define an eligibility function. Every sensor distributively runs the eligibility function at every time step to check its priority in its neighbourhood. The sensor with the highest priority at each one-hop vicinity is the only sensor node that moves to a new position, while other neighbouring sensors stay in their locations. The new location of the prior sensor at the next time step is computed based on the reported locations of neighbouring nodes at the beginning of the current time step, which stays the same during this time step as they will not move in this time step. Therefore, the calculation of the new location is based on accurate information. Moreover, selecting only the prior sensor node to move reduces the total travelled distance of sensor nodes and the consumed energy significantly. The eligibility function returns a priority value based on its two components: the local force and local density, which are explained further in the next sub-sections.

#### 4.2.1 Local force component

As mentioned earlier, the SSND algorithm is based on the Coulomb’s law where particles interact with each other with a defined force, partial force. Therefore, every sensor node in SSND enters a partial force to each of its neighbouring nodes. The total amount of applied forces to \( s_i \) from its neighbouring nodes, \( F^k_{i\text{neighbours}} \), can be calculated as

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_{\text{count}} )</td>
<td>Number of time steps that sensor ( i ) has moved less than the threshold2 up to the current time step ( k )</td>
</tr>
<tr>
<td>( O_{\text{count}} )</td>
<td>Number of time steps that sensor ( i ) has moved less than the threshold1 up to the current time step ( k )</td>
</tr>
<tr>
<td>( S_{\text{lim}} )</td>
<td>Maximum number of time steps that sensor ( i ) has to move less than the threshold2 to be able to stabilise</td>
</tr>
<tr>
<td>( O_{\text{lim}} )</td>
<td>Maximum number of time steps that sensor ( i ) has to move less than the threshold1 to be able to stabilise</td>
</tr>
<tr>
<td>( N^k_i )</td>
<td>Number of time steps that ( s_i ) is previously elected as an eligible sensor to move up to time step ( k )</td>
</tr>
<tr>
<td>( N^k_{\text{neighbours}} )</td>
<td>Set of neighbouring sensors of sensor ( i, s_i ), whose their total force is lower than that of ( s_i )</td>
</tr>
<tr>
<td>( F^k_i )</td>
<td>Total partial force of sensor ( s_i ) at time step ( k )</td>
</tr>
<tr>
<td>( F^k_{\text{local}} )</td>
<td>Local force component of eligibility</td>
</tr>
<tr>
<td>( \text{Elig}^k_i )</td>
<td>Eligibility factor of ( s_i ) at time step ( k )</td>
</tr>
<tr>
<td>( D^k_i )</td>
<td>Neighbour set of sensor ( s_i ) at time step ( k )</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Expected density</td>
</tr>
<tr>
<td>( D^k_{\text{local}} )</td>
<td>Local density component of eligibility</td>
</tr>
<tr>
<td>( \text{Elig}^k_i(\text{Elig}^k_i, D^k_i, F^k_i) )</td>
<td>Eligibility function that return ( \text{Elig}^k_i, D^k_i ) and ( F^k_i ) of ( s_i )</td>
</tr>
<tr>
<td>( \text{Eligset}^k_i )</td>
<td>Eligibility set of sensor ( s_i ) at time step ( k ) that includes its neighbour eligibility factors</td>
</tr>
</tbody>
</table>
Therefore, as the sensor nodes distribute around the sensor nodes at time step $t$, its maximum. Therefore, the local force component for every sensor

Definition: $N_{ij}^k$ represents the set of neighbouring sensors of $s_i$, whose their total force is lower than that of $s_i$:

$$N_{ij}^k = \{ s_m | \forall s_m \in D_i^k, F_i^k \geq F_m^k \}. \quad (7)$$

Definition: For each sensor $s_i$, $s_j \in S$, let $M_i^k$ represents the number of times that $s_i$ is previously elected as an eligible sensor to move up to time step $k$, $0 \leq M_i^k < k$.

The total force represents the adequacy of sensors to move. Hence, the higher the total force of a sensor, the higher its chance to be the prior sensor in its one-hop vicinity. Therefore, for every sensor $s_i$, $N_{ij}^k$ is calculated to represent the force adequacy. Moreover, the number of times that every sensor, $s_i$, has selected; $M_i^k$ up to time step $k$ is important to balance the used energy in the network. The higher the value of $M_i^k$ becomes, there is less probability for $s_i$ to be eligible for the next movement in the neighbourhood.

Consequently, the local force component of $s_i$, at time step $k$, $F_{i\text{elig}}^k$, can be illustrated as below:

$$F_{i\text{elig}}^k = |N_{ij}^k| - M_i^k. \quad (8)$$

The value of $F_{i\text{elig}}^k$ can range from positive to negative number. The minimum value of $F_{i\text{elig}}^k$ for the sensor $s_i$ is a negative number that can be obtained when, for example, $s_i$ has been previously elected as the prior node at least one time, and the total entered force to $s_i$ is minimum in its neighbourhood, i.e., $N_{ij}^k = \emptyset$. In other circumstances, when the sensor node $s_i$ has the highest total entered force and has not been selected before this time step, the value of $F_{i\text{elig}}^k$ reaches its maximum. Therefore, the local force component for every sensor $s_i$, at every time step $k$, is equal to, $F_{i\text{elig}}^k$ that can be in a range of

$$|D_i^k| \geq F_{i\text{elig}}^k > -k. \quad (9)$$

Finally, at every time step, the sensor node with a higher local force value has a higher chance of being the eligible node to move in its neighbourhood. However, local force is one of the components of the eligibility function and cannot guarantee any superiority.

4.2.2 Local density component

In a uniformly distributed network, the local density for every sensor node $s_i \in S$, i.e., the number of neighbouring sensor nodes at time step $k$, $|D_i^k|$, is equal to the expected density, $\mu$.\footnote{15} Therefore, as the sensor nodes distribute around the RoI from their initial deployment, the local density leans towards the expected density.

The absolute value of $|\mu - |D_i^k||$ presents the adjustment that the $s_i$ needs to reach its desired density, $\mu$, from its current situation. In order to uniformly distribute the sensor nodes in an area, all of the sensor nodes should have the same local density. Therefore, the local density component, $D_{i\text{reg}}^k$, is considered as one of the eligibility factors to recognise the sensor nodes with a higher density difference at every time step $k$ and elect them to be the prior node in every neighbourhood as shown:

$$D_{i\text{reg}}^k = \mu - |D_i^k|. \quad (10)$$

The value of the local density component in the best case is zero when the local density is equal to the expected density. The local density component of $s_i$ can be a negative value when there are more neighbouring sensor nodes around the $s_i$ than the expected density and is a positive value when there are fewer neighbouring sensor nodes around the $s_i$ at time step $k$.
4.2.3 | Eligibility value

The eligibility value represents the sensor nodes priority in its neighbourhood that has two components: local force and local density, as described in the previous sections. Although both of these two components affect the eligibility value, the local force component has more influence on eligibility, $Elig^k_i$. In Equation (11), the priority factor of $F^k_{i_{elig}}$ is stated by a parameter $\alpha$. Accordingly, for sensor $s_i$ at time step $k$, the $Elig^k_i$ can be computed as

$$Elig^k_i = \alpha \times \frac{F^k_{i_{elig}} - |D^k_{i_{elig}}|}{N}.$$  

(11)

The process of the eligibility function is presented in Algorithm 1.

**Algorithm 1** Eligibility function

```plaintext
1: function ELIGIBILITY(s_i)
2:     $N^k_{i_{ij}}$ ← ∅
3:     $D^k_i$ ← FindNeighbours(s_i)
4:     for $j = 1 : |D^k_i|$ do
5:         $f^k_{ij} = \frac{|p^k_i - p^k_j|}{||p^k_i - p^k_j||}$
6:     end for
7:     $F^k_i = \sum_{s_j \in D^k_i} f^k_{ij}$
8:     for all $s_j \in D^k_i$ do
9:         if $F^k_i \geq F^k_j$ then
10:             add – member($N^k_{i_{ij}}, s_j$)
11:     end if
12: end for
13: $F^k_{i_{elig}} = |N^k_{i_{ij}}| - N^k_{i_{ij}}$
14: $D^k_{i_{elig}} = \mu - |D^k_i|$
15: $Elig^k_i = \frac{\alpha \times F^k_{i_{elig}} - |D^k_{i_{elig}}|}{N}$
16: return $(Elig^k_i, D^k_{i_{elig}}, F^k_i)$
17: end function
```

4.3 | SSND description

The main aim of the Smart Self-organising Node Deployment (SSND) algorithm is to choose a prior node to move at each time step in every neighbourhood. Hence, only a small percentage of sensor nodes is elected in the entire area to move at every time step. The SSND algorithm is explained in detail in the following and presented in Algorithm 2 as well.

Every sensor node $s_i$ executes the SSND procedure at every time step $k$ until it reaches its stable state and stops its movement. The procedure starts after an initialising step where all constant values such as $O_{lim}, S_{lim}, threshold1, threshold2, \mu, R_c, R_s, p^0_i$ are set and counters $O_{count}$ and $S_{count}$ are initialised. Inside the SSND procedure, there is an Eligibility($s_i$) function that finds the $s_i$’s neighbours, $D^k_i$, calculates the applied total force to $s_i$ and forms the $N^k_{i_{ij}}$ set. The Eligibility($s_i$) function returns the calculated values of the $Elig^k_i$, the eligibility factor, $D^k_i$, the local density component and $F^k_i$, total entered force at the end of its process. While the $Elig^k_i$, the eligibility value of $s_i$ at time step $k$ involves two components, local force component, $F^k_{i_{elig}}$, and local density component, $D^k_{i_{elig}}$. After executing the eligibility function, sensor nodes exchange their information, including their eligibility value within their one-hop neighbours. Later, a sensor node with the highest eligibility value in the neighbourhood can acknowledge itself as the most eligible sensor node. Finally, the eligible sensor node at every neighbourhood calculates its new location and moves towards...
that point. This iterative process continues for every sensor node $s_i$, until the sensor approaches its stable state and stops its movement. A sensor node is in the stable state when one of the following conditions is satisfied (same condition as Heo & Varshney\textsuperscript{15}):

**Algorithm 2 SSND algorithm**

1. **procedure** SSND
2. $(E_{lig}^k, D_i^k, F_i^k) \leftarrow \text{Eligibility}(s_i)$
3. $E_{ligset}^k \leftarrow \emptyset$
4. **for** all $s_j \in D_i^k$ **do**
5. $(E_{lig}^{j}, D_j^k, F_j^k) \leftarrow \text{Eligibility}(s_j)$
6. $\text{add} \leftarrow \text{member}(E_{ligset}^k, E_{lig}^{j})$
7. **end for**
8. **if** $E_{lig}^k \geq \text{Max}(E_{ligset}^k)$ **then**
9. $p_i^{k+1} = p_i^k + F_i^k$
10. Increase $M_i^k$
11. **end if**
12. **if** $|p_i^{k+1} - p_i^{k-1}| \leq \text{threshold1}$ **then**
13. Increase $O_{count}$
14. **if** $O_{count} > O_{lim}$ **then**
15. $p_i^{k+1} = \frac{p_i^{k} + p_i^{k-1}}{2}$
16. $s_i\text{ is stable}$
17. **end if**
18. **end if**
19. **if** $|p_i^k - p_i^{k-1}| \leq \text{threshold2}$ **then**
20. Increase $S_{count}$
21. **if** $S_{count} > S_{lim}$ **then**
22. $p_i^{k+1} = p_i^{k}$
23. $s_i\text{ is stable}$
24. **end if**
25. **end if**
26. **end procedure**

**Oscillation check:** A sensor node might move back and forth within a predefined distance, $\text{threshold1}$, during the deployment process. This is measured for every sensor node by counting the number of its occurrence, $O_{count}$. Every sensor node stops its process when its $O_{count}$ reaches the defined oscillation limits, $O_{lim}$.

**Stability check:** In the SSND algorithm, a sensor node that moves less than a specific distance, $\text{threshold2}$, consecutively and after a definite number of times, $S_{lim}$, is considered a stable node. Therefore, every sensor node $s_i$ counts the number of sequential movements that are less than $\text{threshold2}$ by $S_{count}$ from when it starts and compares it with $S_{lim}$ at every time step to recognise its state.

## 5 PERFORMANCE EVALUATION

In this section, the performance of the SSND is compared with the mentioned algorithms, VEVF, DSSA and SODA. The performance evaluation is demonstrated first by discussing the simulation setup, followed by the result of an exemplar deployment of all algorithms. Then, the results of an extensive performance study are presented to compare them in terms of total final coverage, total travelled distance and also their dependency on the initial deployment and communication range. It should be noted that the sensing, computing and communication costs in MSNs are negligible since the mobility cost is multiple order-of-magnitude higher than the other energy costs.\textsuperscript{26} Thus, the total energy consumed by sensor nodes is a function of the total travelled distance by all nodes.
5.1 Simulation setup

The DSSA, SODA, VEVF and SSND algorithms are simulated in a 50 × 50 flat region using Matlab R2019a. The threshold values for oscillation and stability for DSSA, SODA and SSND are considered to be 0.1522, the same as the assumptions widely adopted in the literature. Also, the priority factor, \(\alpha\), for the SSND algorithm is considered to be 2. In the following, three different scenarios are described to evaluate the performance of SSND with the communication ranges of \(R_c = 10, 15, 20\) and the sensing range of \(R_s = 6\). The threshold values and the \(\alpha\) for the SSND algorithm are the same in all scenarios as described above. The sensor nodes in each scenario are homogeneous with the same characteristics defined for each case. The average values are obtained and presented after executing the simulation model for 100 times. All the simulation settings are consistent with those reported in the literature. Moreover, the implemented algorithms are validated using the available data reported in the literature.

5.2 Results

In Figure 3A, the initial deployment of 40 sensor nodes with communication range \(R_c = 15\) and sensing range \(R_s = 6\), is shown (the points are the locations of sensor nodes and the circles around them are their sensing range). The rest of this figure presents the final deployments of sensor nodes for DSSA, VEVF, SODA and SSND algorithms. Figure 3E represents the SSND final deployment that has covered more than 87\% of the RoI. The final deployments of DSSA, VEVF and SODA for this initial deployment are shown in Figure 3B–D with coverage percentages of 69.20\%, 77.42\% and 71.74\%, respectively. The uniform distribution of the SSND algorithm results in higher final coverage, while the DSSA, SODA and VEVF have terminated their algorithms before reaching a proper final coverage.

In the following, the result of the DSSA, SODA, VEVF and SSND algorithms are compared in three different scenarios. First, the effect of scalability is studied by analysing the total travelled distance of sensors when the communication and sensing ranges are fixed. Afterwards, the impact of the initial coverage percentage on the final coverage is discussed. Finally, the outcome of the initial coverage with a different communication range and the effect of different communication ranges on the final coverage are presented.

![Figure 3](image.png)

**Figure 3.** An example of final coverage of DSSA, VEVF, SODA and SSND algorithms based on the same initial deployment of sensor nodes: (A) the initial coverage of the sensors, (B) the final coverage of DSSA, (C) the final coverage of VEVF algorithm, (D) the final coverage of SODA and (E) the final coverage of SSND algorithm.
5.2.1 | Scalability

In this scenario, $R_s$ and $R_c$ are assumed 6 and 15, respectively. The sensors are randomly placed in a limited area of the RoI ($50 \times 50$), and the same limited initial coverage is considered for 20, 30, 40 and 50 sensors in the RoI. Figure 4 shows the final provided coverage of DSSA, SODA, VEVF and SSND algorithm under the mentioned condition. As it is shown in Figure 4, the final coverage of all four algorithms increases by increasing the network size, i.e., the number of sensors. However, the growth rate of the final coverage is higher in a smaller number of sensors. Moreover, the growth rate of the SSND coverage is higher than those of DSSA, SODA and VEVF algorithms. For instance, for 20 sensors in RoI, SSND provides 61.76% final coverage that increases to 91.72% for 50 sensors, while in VEVF, it is increased from 57.04% to 75.87%. Therefore, the final coverage of SSND is 15.85% higher than that of VEVF. Besides, the improvement of SSND in final coverage increases to 25.12%, and 29.12% in comparison to SODA and DSSA, respectively.

In Figure 5, the total travelled distance of a different number of sensor nodes under the mentioned scenario is presented. In DSSA, SODA and SSND, the total travelled distance increases by increasing the network size, i.e., the number of sensors. However, it is not the same for VEVF. In VEVF, the total travelled distance increases by increasing the number of sensors from 20 to 30, and then, this value diminishes by further increasing the number of sensors. As discussed in Mahboubi et al., the reason for the VEVF behaviour is the relatively large Voronoi cells in comparison to the sensing range for the small number of sensors during the initial deployment.

In Figure 5, DSSA has the largest total travelled distance, while VEVF has the lowest. Also, the total travelled distance increases by increasing the network size in SODA, DSSA and SSND; however, it does not prove the higher travelled distance by every sensor in the network as the number of sensors increases at the same time. The mean travelling distance, the average travelling distance by a sensor, can demonstrate the distribution of the total travelled distance between sensor nodes.

In Figure 6, the mean travelling distance of sensor nodes in different network sizes is presented. As expected in SSND and VEVF, the mean travelling distance decreases as the network size increases due to less need for every sensor to move in the RoI. In a network with more sensor nodes, the movement overhead is distributed among a higher number of nodes and, hence, less distance for every node to travel. However, in DSSA and SODA, more sensor nodes in the network create larger forces that make sensor nodes move unreasonably. Although, in the end, these algorithms provide higher coverage than in scenarios with a lower number of nodes, they experience long-distance movements during the process.
5.2.2 | The effects of initial coverage

Most studies in coverage problems evaluate the node deployment algorithms by running several simulations with random initial positions for sensor nodes.\textsuperscript{9,12,17,18} Although a random initial deployment provides a strong understanding of the algorithm’s performance, it is not a realistic deployment for most of the applications. In many cases, such as disaster areas and chemical sensitive environments, manual or random initial deployment is impossible. Besides, random initial locations of sensor nodes cannot provide uniformly distributed locations because of natural phenomena such as wind and also obstacles, like trees and buildings.\textsuperscript{12} In this section, the initial location of sensor nodes is limited to a small area for DSSA, SODA, SSND and VEVF.

The total coverage and total travelled distance of 40 sensor nodes with $R_c = 15$ and $R_s = 6$ under the above-mentioned conditions are presented in Figures 7 and 8, respectively. The x axis in Figure 7 shows the initial coverage percentage of the initial deployment, from around $20\%$ to $50\%$, and the y axis provides the final coverage of the relevant deployment. It is shown in Figure 7 that DSSA, SODA and VEVF provide more coverage as the initial coverage
percentage increases, despite SSND that provides almost the same final coverage, around 88\%, regardless of initial coverage percentage. The more scattered initial deployment for those algorithms, i.e., higher initial coverage percentage, results in higher final coverage. However, the eligibility function of SSND smoothens the final coverage by acknowledging the different local densities. In SSND, sensor nodes with higher differences between their local densities and expected density, i.e., sensor nodes in a dense area, have higher eligibility to move and are encouraged to increase the final coverage. On the other hand, a scattered initial deployment, close to 50\%, has closer local densities to the expected density in comparison to a dense initial deployment, hence, sensor nodes have less eligibility value to move and increase the final coverage. Therefore, the effect of the local density in the eligibility function diminishes as the initial coverage percentage increases. Thus, the provided coverage for higher initial coverage, e.g., 50\%, in SSND is less advantageous compared to other algorithms when the nodes are initially more scattered.

In Figure 8, the total travelled distances in different initial coverage are presented. Similar to coverage in Figure 7, the performance of DSSA, SODA and VEVF algorithms depend on the initial coverage percentage. As shown in the
figure, the provided initial coverage and the total travelled distance have an inverse relationship in DSSA, SODA and VEVF algorithms. Despite this, SSND has almost no dependency on the initial coverage. It is also shown that the lowest total travelled distances are reported for VEVF algorithm. However, the reduction in the total travelled distance by increasing the initial coverage percentage is higher than those in SSND. The informative decision-making process of SSND, eligibility function, reduces the influence of initial deployment to total movement of sensor nodes and makes the reduction negligible.

5.2.3 | The effects of communication range

The communication range is one of the important physical features of every sensor node in the area. Moreover, in many applications, the implementation cost or other restrictions limits the flexibility in choosing sensor nodes with higher or lower communication ranges. Therefore, a node deployment algorithm should identify its performance dependency on the communication range. In this section, first, the performance of all four algorithms are compared with a lower communication range for 40 sensor nodes with different initial coverage to investigate its effect. Afterwards, the final coverage of all algorithms under a different communication range is presented.

Initial coverage: In this scenario, the effect of initial coverage on the final coverage is investigated by each of the DSSA, SODA, SSND, and VEVF algorithms with $R_c = 10$. The final coverage of 40 sensor nodes for different initial coverage in DSSA, SODA, SSND and VEVF, are presented in Figure 9. The final coverage of all four algorithms is increased with the same behaviour as the one presented in Figure 7 with $R_c = 15$ under the same initial deployment conditions. However, the obtained final coverage of all the algorithms for $R_c = 10$ is higher, in this case around 10%, than of those of $R_c = 15$. The lower final coverage for $R_c = 15$ has resulted from the unnecessary higher communication range that can violate the local decision-making advantages by inessential extension of the neighbourhood. Therefore, the final coverage of DSSA, SODA and SSND increases while reducing the communication range to a proper range. However, in VEVF, the shorter communication range results in inaccurate construction of Voronoi cells and affects the detection of holes, which is discarded in this scenario. Therefore, the superiority of provided coverage in SSND with a lower communication range over other algorithms is clear, even when no inaccurate Voronoi cell is considered in the simulation of VEVF.

Total coverage: As mentioned in the previous scenario, the communication range of the sensor nodes affects the total coverage of the node deployment algorithms. The effect of communication range on the total coverage for DSSA, SODA and SSND is shown in Figure 10. The VEVF has been neglected in this graph as it only performs correctly with high communication ranges.

![Figure 9](https://example.com/figure9.png)

**Figure 9** Total coverage of 40 sensor nodes with $R_c = 10$ in a $50 \times 50$ region
The total coverage of SSND, SODA and DSSA algorithms is increased by decreasing the communication range. However, the SSND has the highest final coverage under different communication ranges, always above 80% and close to 100% when \( R_c = 10 \), with considerable differences in comparison to DSSA and SODA. Higher communication range in these algorithms adds farther sensor nodes in the neighbouring set leading to more uncertainty in local decision-making. In contrast, the nearest neighbouring sensor nodes can provide more accurate information about the local condition to improve the decision-making process and hence the final coverage. Therefore, despite the VEVF algorithm, which depends on the high values of communication range to provide the required coverage by constructing accurate Voronoi cells, DSSA, SODA and SSND can achieve the required coverage with lower communication range and less deployment cost in comparison to higher communication range.

6 | CONCLUSION

A family of node deployment algorithms inspired by the equilibrium of molecules, such as DSSA and SODA, has been recently reported in the literature to maximise the area coverage in MSNs. Although this family provides maximum coverage under some circumstances, the obtained coverage is costly in many scenarios, mainly due to high energy consumption and entailed latency. Conversely, another family of node deployment algorithms that is based on the Voronoi diagram is recently proposed to cover the holes in the area with less energy consumption. However, the achieved coverage is not sufficient in many configurations, like in dense deployment. Moreover, the short communication range of sensor nodes in this family results in a faulty structure of the Voronoi diagram and also increases the chance of sensor nodes collisions. Furthermore, both families suffer from the collective movement of sensor nodes at every time step, which leads to high energy consumption.

In this paper, we have proposed a novel Smart Self-organising Node Deployment (SSND) algorithm based on the equilibrium of molecules for mobile sensor networks to overcome the limitations listed above. In the SSND, an eligibility function prioritises one sensor node at each neighbourhood to move at each time step, in order to reduce the total travelled distance of sensor nodes and hence the consumed energy. The lower total movement of sensor nodes does not decrease the final coverage of the area, and the SSND algorithm provides proper coverage with reasonable energy consumption irrespective of the initial locations of sensor nodes and the initial coverage. Furthermore, sensor nodes in SSND can have shorter communication ranges without affecting the accuracy of the algorithm. The results of our extensive simulation study have confirmed the superiority of the SSND algorithm. It provides close to 30% higher coverage in different configurations with affordable energy consumption in comparison to DSSA, SODA and VEVF. Moreover, the final deployment is uniform regardless of the initial deployment. The achieved results confirm the fact that by using valid information, moving only one sensor node in every one-hop vicinity is enough for full coverage.
There are some interesting open problems that require further investigation. The ability to bypass obstacles has not been envisaged in any deployment algorithm reported in the literature. This can be a possible future line of study. Also, no mechanism is applied to the deployment algorithms to avoid a collision. These algorithms can be enhanced by applying safe trajectory planning. Finally, extending these algorithms to provide three-dimensional coverage will be another interesting line of research for the future.

ORCID
Alireza Shahrabi https://orcid.org/0000-0002-6583-0617

REFERENCES