

A Smart Self-Organizing Node Deployment Algorithm in Wireless Sensor Networks

Mahsa Sadeghi Ghahroudi, Alireza Shahrabi, and Tuleen Boutaleb
 School of Computing, Engineering and Built Environment
 Glasgow Caledonian University, Glasgow, UK
 {Mahsa.Sadeghi, A.Shahrabi, T.Boutaleb}@gcu.ac.uk

Abstract—Node deployment is one of the fundamental issues in Wireless Sensor Networks (WSNs) which has not only a direct impact on the effectiveness of other operations, such as routing and data fusion, but also on the appropriateness of the provided coverage expected in many applications such as national security, surveillance, military, health care, and environmental monitoring. In mobile sensor networks, the resource-constrained move-assisted sensor nodes are used in an area to maximise the coverage within a reasonable time and energy cost. Recently, a family of algorithms, inspired by the equilibrium of molecules, have been proposed to address the coverage issue. However, these solutions lead to high energy cost and latency due to two major issues. First, almost all the nodes in the network try to move to a new position at each stage. Even worse, the decision made at each node on to which point to move at each stage is purely based on obsolete information, i.e. the current locations of moving neighbouring nodes. In this paper, we propose a new distributed algorithm, called SSND, to efficiently provide the maximum coverage for WSNs that use mobile nodes. SSND avoids to collectively, and blindly, move sensor nodes at each step but to apply an eligibility function to elect a few nodes to move using the valid information to obtain the maximum effect. Our extensive simulation study under various operational conditions shows not only a higher percentage of area coverage by SSND but with much lower power consumption and latency than those of the other protocols recently reported in the literature.

Index Terms—distributed wireless sensor network, energy efficiency, smart node deployment algorithms, Self-Organizing

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have recently gained significant attraction in different areas of science due to their widespread applications. Such applications range from health surveillance to battlefield surveillance and environmental monitoring [1], [2]. One of the problems in WSNs is the sensor nodes deployment. Deployment of sensor nodes influences the coverage, uniformity, and connectivity of the network. An optimised sensor deployment algorithm provides a maximum area coverage which is the goal of most applications [3].

The WSN algorithms for sensor nodes deployment are categorised as centralised or distributed [4]. The centralised method includes a node, i.e. sink, that controls the whole sensor nodes' locations whereas the distributed method does not rely on a centralised node to decide on all the sensor nodes' locations. In the distributed method, a sensor can communicate with its neighbouring nodes and determine its location at every time step based on its local information. The distributed method is fault tolerance, scalable, and cost-

efficient and, hence a more popular approach in wireless sensor networks [5], [6].

The coverage problem in distributed sensor nodes deployment algorithms in WSNs has been widely studied [7]–[10]. Many WSNs, consist of fixed sensor nodes where mobile sensors only cover the holes in the area [11]–[13]. However, the full area coverage is also provided with the iterative movements of identical mobile sensor nodes [14], [15]. In a mobile sensor network, all sensor nodes move at every time step, and this iterative action continues until the sensor reaches its stable state [14], [15].

Inspired from the field of biology and chemistry, some algorithms have been proposed to utilise the local information for the collective movement of mobile sensor nodes, which are categorised into two groups, animal aggregations and equilibrium of molecules. Natural phenomena such as schools of fish, flocks of birds or swarms of bees are some of the examples of animal aggregations [16], [17]. The behaviour of these phenomena likes, schools of fish is mathematically presented [8], and used to improve the coverage in WSNs [9].

In the equilibrium of molecules, the optimal location of sensors for the required coverage mapped to the location of molecules where molecules have the lowest energy and are in the equilibrium state [14], [18], [19]. Recently, a family of algorithms, inspired by the equilibrium of molecules, have been proposed to address the coverage issue [14], [18], [19]. Every sensor node uses the locations of its neighbouring nodes to calculate the partial force in order to determine to move to which point. However, these solutions lead to high energy cost and latency due to two major issues. First, almost all the nodes in the network try to move to a new position at each time step. Even worse, the decision made at each node on to which point to move at each stage is purely based on obsolete information, i.e. the current locations of moving neighbouring nodes.

In this paper, we aim to develop a distributed deployment algorithm, called Smart Self-organization Node Deployment (SSND) for mobile sensor networks. Our main intention is to achieve maximum coverage within an optimise energy consumption and time cost. SSND addresses the collectively and blindly movement of sensor nodes limitation by moving one sensor in every neighbourhood by an eligibility function which is inspired by the equilibrium of molecules while using valid information from sensor nodes. The smart movement of the sensors is the superiority of the SSND where reduces the total sensor movements, latency and power consumption significantly under various operational conditions.

The rest of the paper is organised as follows. Section II introduces the related work followed by the collective movement limitations in Section III. The SSND algorithm is introduced in Section IV that includes the motivations, node movement criteria and SSND algorithm. The performance evaluation section to describe the simulation specification and results is under Section V while our conclusions are drawn in Section VI.

II. RELATED WORK

In the sensor nodes deployment algorithms the communication of sensor nodes is an important concept. The communication between neighbouring nodes in distributed sensor networks has been studied from different perspectives [6], [14], [15], [20]. In some algorithms, the procedure that sensors communicate with each other is inspired by natural phenomenon behaviours. For example, neighbourhood movement theory which is seen in the animal aggregation movements, like birds migration, is applied in sensor nodes deployment problem in WSNs [16], [17]. In neighbourhood movement theory every sensor moves based on its neighbour's positions. A sensor makes an average of its neighbours' positions and moves towards the calculated point. The distributed movement of the sensors creates a uniform sensor placement with the required coverage.

In another group of studies, the sensor movement is simulated from the equilibrium of molecules, which molecules reach their balance and stable location after some time. In the stable state, the molecules are in their lowest energy point and almost equal distance from other molecules. The dispersion of the molecules with equal distance from their neighbours results in a uniform distribution of the molecules. The uniform distribution has resulted from the equilibrium of molecules that is used in Distributed Self-Spreading Algorithm (DSSA), to cover an area with sensor nodes [14]. A sensor node in every time step moves based on the partial force that is applied from other sensor nodes as the result of the equilibrium of molecules concept. Although in the DSSA the required coverage is achieved, the adequate coverage depends on the initial locations of sensor nodes and specific level of coverage during the initial deployment while suffering from the long transition from chaos to order state.

Another study introduced Self-Organizing node Deployment Algorithm (SODA), that is an algorithm based on the equilibrium of molecules concept [19]. In SODA, despite DSSA, the weakness of initial dependent dependency, sensitivity to the initial coverage and long transition from chaos to order state are resolved. The effective parameters in the environment adjust the irrespective partial force that results in non-uniform deployment in many scenarios, and provide an adjusted partial force in SODA. The partial force is adjusted by the number of sensors in the neighbourhood and the expected density (μ) parameters. The adjustment of partial force based on the local network density improves the sensitivity to initial coverage and as the result final coverage. The density-aware algorithm, SODA that is based on the collective movement of sensor nodes reduces the power consumption and improve the final coverage in comparison to DSSA.

Despite the improvements that are achieved by SODA concerning the area coverage, its energy cost is still high due to collective movement of the nodes and also using the obsolete information when calculating the next point to move at each node. Besides, the chaos that happens by moving the nodes collectively may cause many physical sensor collisions and false movement in the area.

III. LIMITATIONS OF COLLECTIVE MOVEMENT ALGORITHMS

In this section, the limitations of the two proposed sensor deployment algorithms, DSSA [14] and SODA [19] are discussed. Both of the algorithms are inspired by the equilibrium of molecules and are based on collective movement.

A. DSSA limitations

The major limitations of DSSA are dependency to the initial deployment, sensitivity of the final coverage to the initial coverage of sensors, the requirement of a non-single-point-deployment and also the long time transition from chaos to order state. The details of each limitation is further described in the following sections.

1) *Initial deployment dependency*: In DSSA, the performance of the network varies for the same number of sensor nodes. Using the same number of sensors, this algorithm generates different figure for the final coverage by varying the initial locations of sensors. The unknown performance for a specific number of sensors makes the DSSA an unpredictable and uncertain algorithm.

2) *Sensitivity to the initial coverage*: In order to be able to distribute sensors evenly in the area a minimum initial coverage is required by DSSA. If the initial coverage is below a certain percentage, a full coverage is not achievable. In most of the cases, the initial coverage requirement after initial random deployment of nodes, given that there are sufficient number of nodes, is about 90%. Obviously, this requirement is not possible in so many applications.

3) *non-single-point-deployment*: The partial force that influences the movement of the sensors to the new location, $S = \{s_1, s_2, \dots, s_n\}$, where $N = |S|$ is presented in Equation 1. The communication range and sensing range of the sensors are C_R and S_R , respectively and μ represents the expected density. The location of sensors is specified using a 2D vector at time step k , $p_i^k = (x_i^k, y_i^k)$ and the distance between s_i^k and s_j^k is $P_{ij}^k = \sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2}$. The partial force at step k for sensor i and its neighbouring sensor j is a repulsive force calculated as:

$$f_{ij}^k = \frac{|D_i^k|}{\mu^2} (C_R - |p_i^k - p_j^k|) \frac{p_j^k - p_i^k}{|p_j^k - p_i^k|} \quad (1)$$

where, p_i^k is the position of sensor i at time step k , and $D_i^k = \{s_m | \forall s_m \in S, P_{im} < C_R, i \neq m\}$ represents the local density (i.e. the set of neighbours) of sensor i at step k . The new location of the s_i is $p_i^{k+1} = p_i^k + \sum_{s_j \in D_i^k} f_{ij}^k$.

Considering all the principles, for a $s_i \in S$ in a dense area at time step k the local density has the greatest value, $D_k^i = \{s_1, s_2, \dots, s_m\}$, that $s_i \notin D_k^i$. In addition, the $C_R - |p_i^k - p_j^k|$ section of the partial force in dense local area where $s_i \in S$ and $s_j \in D_k^i$ are very close to each other and $\lim_{s_i^k \rightarrow s_j^k} |p_i^k - p_j^k| = 0$ becomes $\lim_{s_i^k \rightarrow s_j^k} C_R - |p_i^k - p_j^k| = C_R$. Therefore, as the distance in between sensors reduces, the partial force increases. The large partial force scatters all the sensors towards the edges and corners of the area and breaks sensor connectivity. Therefore, a single-point-initial-deployment in DSSA that creates a dense area does not lead to a full coverage of the area.

4) *long transition from chaos to order state*: As it is mentioned in the above sections, the initial deployment of sensors is one of the parameters that can significantly affect the partial force. A non-accurate partial force results in many false movements which causes chaos in the area. However, every sensor node stops its movement at the specific time step, and decreases the number of false movements. Therefore, the chaos state diminishes when sensor nodes reach their stable state and transit to order state. This happens very late in DSSA during the last steps of the algorithm where almost the full coverage has been achieved.

B. SODA limitations

Some of the limitations of DSSA is addressed in SODA. The density-adjusted partial force in SODA makes it independent to initial location of the sensor nodes. Also, the need to a minimum initial coverage is lifted, and the long transition from chaos to order state is significantly shortened. Addressing those issues while still relying on the equilibrium of molecules idea makes SODA a reasonable solution to achieve an optimal coverage. However, for many applications, incurring the long delay and high energy consumptions due to collective movements of sensors are not sustainable.

1) *Energy consumption*: In SODA, every sensor moves in every time step until it is stabled. In every time step, almost all the sensors move and consume energy, where is not necessary for all the sensor nodes to move. The collective movement of sensors reduces the energy level and lifetime of all sensors. The consumed energy can be saved by smartly moving only a few nodes to obtain the maximum effect at every time step.

2) *Decision-making using obsolete information*: In the SODA, sensor nodes exchange their locations information prior to their movement in every time step. However, as the location-calculation process happens for all the sensor nodes in every time step at the same time and the sensors all move afterwards, the used information are obsolete.

C. Motivation

Motivated by the shortcomings of the collective movement deployment algorithms discussed in the previous section, the SSND algorithm is proposed to overcome those issues. Our proposed algorithm also uses the concept of the equilibrium of molecules described by Coulomb's law, for the applied forces between sensor nodes. The Coulomb's law states that the electrostatic force in between charges with the same sign

is repulsive and is proportional to the charge's distance. In SSND, sensor nodes are considered with the same charge, and the applied force to sensor s_i from sensor s_j has a direct relation with the distance between them, which is the same as the partial force in DSSA [14]. The key factors of SSND are the accurate movement of sensors in every step to avoid false and collective movement and also the reduction of the consumed energy. Despite collective movement algorithms, in SSND only the node that has the highest eligibility value in its neighbourhood moves to the new location. The eligibility value is calculated in every time step based on the parameters affecting the sensor nodes movement.

The affecting parameters in every movement, are described in the next section.

D. Node movement criteria

To address the blind and collective movements of sensor nodes, we set two criteria to identify an eligible sensor node for movement in each neighbourhood. At each time step, only one node moves to a new location and the other nodes remain in their places. The new location of the moving node is calculated based on the locations of other nodes in the neighbourhood, which are not yet moved and, hence, their information are still valid. The local force eligibility and local density eligibility are the criteria applied in each neighbourhood to control the movement of sensors at every time step.

1) *Local Force Eligibility*: Let F_i^k represents the total amount of force applied between s_i and its neighbouring nodes (Equation 1), which can be calculated by:

$$F_i^k = \sum_{s_j \in D_i^k} f_{ij}^k \quad (2)$$

Definition: $\mathcal{N}_{i,f}^k$ represents the set of neighbouring sensors of sensor i , s_i , whose their total force is lower than that of s_i :

$$\mathcal{N}_{i,f}^k = \{s_m | \forall s_m \in D_i^k, F_i^k \geq F_m^k\} \quad (3)$$

Definition: The j^{th} member of the D_i^k at time step k is shows D_{i-j}^k . Therefore, $s_m \in \mathcal{N}_{i,f}^k$ can also be shown as D_{i-j}^k .

Definition: For each sensor s_i , $s_i \in S$, let's \mathcal{M}_i^k represents the number of times that s_i is elected as an eligible sensor to move up to time step k , ($0 \leq \mathcal{M}_i^k \leq k$).

\mathcal{M}_i^k is a crucial influencing factor on the eligibility of a sensor for movement. The higher the value of \mathcal{M}_i^k the less probability of being elected, i.e. to be eligible, for movement in a neighbourhood.

Now, using these definitions we can define the local force eligibility of sensor i , s_i , at time step k , $F_{i,elig}^k$:

$$F_{i,elig}^k = \mathcal{N}_{i,f}^k - \mathcal{M}_i^k \quad (4)$$

It should be noted that the value of $F_{i,elig}^k$ can be negative or positive. A negative value may be obtained when, for instance, s_i has the minimum force in its neighbourhood at one time step while has already been elected, at least once, as an eligible

node to move. The maximum value of $F_{i_{elig}}^k$ happens when s_i has the maximum force in its neighbourhood and has not been elected to move. Therefore, $-k \geq F_{i_{elig}}^k \geq |D_i^k|$.

A node with a higher value of local force eligibility has a higher chance to be elected as an eligible node to move at a given time step.

2) *Local Density Eligibility*: When calculating the force between s_i and s_j , the value of D_i^k can directly influence the value of f_{ij}^k . However, when prioritising the sensors to move the value of the local density is less important and the difference between μ and D_i^k should be bold.

The absolute value of the difference of μ and $|D_i^k|$ shows the distance that s_i has to eliminate in order to achieve its perfect density, μ . Therefore the parameter, local density eligibility, $D_{i_{elig}}^k$ is introduced as one of the eligibility factors when electing a sensor in sensor s_i neighbourhood to move at time step k :

$$D_{i_{elig}}^k = \mu - |D_i^k| \quad (5)$$

The value of local density eligibility in the best case is zero. By increasing or decreasing the difference between μ and $|D_i^k|$, the eligibility of the sensor increases linearly.

IV. SMART SELF-ORGANIZATION NODE DEPLOYMENT (SSND)

A. Smart Self-organization Node Deployment (SSND)

The main goal of the Smart Self-organization Node Deployment (SSND) algorithm is to elect a node in each neighbourhood to move at each time step. Therefore, instead of moving almost all nodes at each step, only a low percentage of nodes are elected across the whole area to move. The description of SSND algorithm is shown in Algorithm 1.

The sensors in the specified area start their process by executing the initialize procedure in the beginning. After that, every sensor node $s_i \in S$ runs SSND procedure every time step until the sensor stops its movement. The SSND procedure calls the $Eligibility(s_i)$ function that returns the $Elig_i^k$, D_i^k , and F_i^k values. At each time step k and after one-hop exchanging of information, including their eligibility, $Elig_{D_i^k}^k$, the eligible node, with the highest eligibility value can find out itself as the only node in the neighbourhood which has to calculate a new location to move. By moving one node from each neighbourhood at each time step and repeating this over the next time steps, all local densities tend to converge toward the expected density, μ . The sensor stops the movement and marks itself as a stable sensor when one of these two conditions are satisfied [14]:

- *Oscillation Check*: A sensor ends the process of partial forces calculation when it reaches its O_{lim} . The back and forth, movement of a sensor in a defined distance is called oscillation. O_{count} represented the number of the time that the sensor moves back and forth within a specific distance is called, $threshold1$. A sensor stops its movement when it reaches the oscillation limits O_{lim} . A sensor is finally stabilized when $O_{count} \equiv O_{lim}$.

- *Stability Check*: Stability check recognises the sensors that have reached their stable state. A sensor that moves within $threshold2$ distance for a certain number of times becomes

Algorithm 1 SSND Algorithm

```

1: procedure INITIALIZE
2:    $\mu = N * \pi * C_R^2 / A$ 
3:    $\mathcal{M}_i^k, S_{count}, O_{count} = 0$ 
4:    $C_R = 4$ 
5:    $S_R = 2$ 
6:    $threshold = 0.1522$ 
7:    $p_i^0 \leftarrow (x_i^0, y_i^0)$ 
8: end procedure

9: procedure SSND
10:   $(Elig_i^k, D_i^k, F_i^k) \leftarrow Eligibility(s_i)$ 
11:   $Eligset_i^k \leftarrow \emptyset$ 

12:  for  $j=1$  to  $|D_i^k|$  do
13:     $(Elig_j^k, D_j^k) \leftarrow Eligibility(D_{i-j}^k)$ 
14:     $add - member(Eligset_i^k, Elig_j^k)$ 
15:  end for

16:  if  $Elig_i^k \geq Max(Eligset_i^k)$  then
17:     $p_i^{k+1} \leftarrow p_i^k + F_i^k$ 
18:     $Increase \mathcal{M}_i^k$ 
19:  end if

20:  if Oscillation then
21:     $Increase O_{count}$ 
22:    if  $O_{count} > O_{limit}$  then
23:       $p_i^{k+1} \leftarrow \frac{p_i^k + p_i^{k-1}}{2}$ 
24:       $s_i$  is stable
25:    end if
26:  end if

27:  if Stability then
28:     $Increase S_{count}$ 
29:    if  $S_{count} > S_{limit}$  then
30:       $p_i^{k+1} \leftarrow p_i^k$ 
31:       $s_i$  is stable
32:    end if
33:  end if
34: end procedure

```

stable. This number is defined as Stability Limit, S_{lim} , and a sensor is stable if $S_{count} \equiv S_{lim}$.

The $Eligibility(s_i)$ function in Algorithm 2 finds the s_i 's sensor nodes neighbour, D_i^k , and after calculating the total force applied to s_i , F_i^k , builds the $N_{i_{if}}^k$ set. This function returns the total force, F_i^k , local density, D_i^k , and the eligibility factor, $Elig_i^k$ values. $Elig_i^k$ is the eligibility value of s_i at time step k that includes the two node movement criteria, local force eligibility, $F_{i_{elig}}^k$, and local density eligibility $D_{i_{elig}}^k$. However, the local force eligibility has more influence on $Elig_i^k$. The priority factor of the $F_{i_{elig}}^k$ in the equation $Elig_i^k$ is specified by the cofactor α . Therefore, the $Elig_i^k$ for s_i at

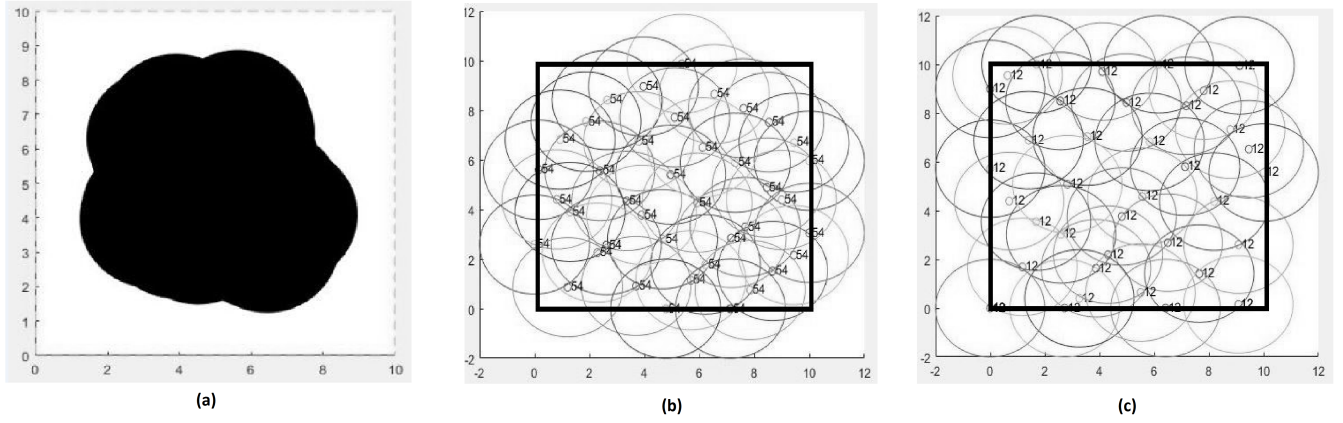


Fig. 1: The distribution of sensors (a) The initial sensor distribution. (b) The final location of sensors after DSSA simulation. (c) The final location of sensors after SSND algorithm.

time step k is:

$$Elig_i^k = \frac{\alpha \times F_{i_{elig}}^k - |D_{i_{elig}}^k|}{N} \quad (6)$$

Algorithm 2 Eligibility Function

```

1: function ELIGIBILITY( $s_i$ )
2:    $\mathcal{N}_{i_{if}}^k = \emptyset$ 
3:    $D_k^i \leftarrow FindNeighbours(s_i)$ 
4:   for  $j=1 : |D_k^i|$  do
5:      $f_{ij}^k = \frac{|D_k^i|}{\mu^2} (C_R - |p_i^k - p_j^k|) \frac{p_j^k - p_i^k}{|p_j^k - p_i^k|}$ 
6:   end for
7:    $F_i^k = \sum_{s_j \in D_k^i} f_{ij}^k$ 
8:   for  $j=1$  to  $|D_k^i|$  do
9:     if  $F_i^k \geq F_j^k$  then
10:       $add - member(\mathcal{N}_{i_{if}}^k, D_{i-j}^k)$ 
11:    end if
12:  end for
13:   $F_{i_{elig}}^k = \mathcal{N}_{i_{if}}^k - \mathcal{M}_i^k$ 
14:   $D_{i_{elig}}^k = \mu - D_i^k$ 
15:   $Elig_i^k = \frac{\alpha \times F_{i_{elig}}^k - |D_{i_{elig}}^k|}{N}$ 
16:  return ( $Elig_i^k, D_i^k, F_i^k$ )
17: end function
    
```

V. PERFORMANCE EVALUATION

A. Simulation Setup

The DSSA, SODA and SSND algorithms are simulated in a 10 x 10 region using Matlab. The initial coverage of the area is not limited to any specific deployment, and for every run, a random deployment of sensors with different initial coverage is considered. The S_R and C_R are assumed 2. The *threshold* for oscillation and stability is considered to be 0.1522, the same as those used in DSSA performance study [14]. The new introduced parameter, Priority factor α is considered 2 in the simulation.

B. Results

In Figure 1, an initial deployment of 40 sensor nodes is shown under section (a). The final sensor deployments of the DSSA and SSND algorithm for this initial deployment are presented in Figure 1 (b) and (c), respectively. The distribution of the sensors in the Figure 1 (c), is more uniform in compare to Figure 1 (b). The uniform deployment of SSND algorithm has achieved by 12 sensor movements, while the DSSA deployment needed 54 sensor movements. Therefore, SSND algorithm resulted in more coverage and efficient energy usage even with sufficient number of sensor nodes for both DSSA and SSND algorithm.

A comprehensive study of SSND algorithm for average results of 60 runs is reported in the following area coverage, energy consumption, and total number of moved sensor sections.

1) *Area Coverage:* The simulated results of final area coverage of DSSA, SODA, and SSND algorithm are shown in Figure 2. The final coverage of an area for all the algorithms is based on the same initial deployment. As expected, the percentage of coverage increases by increasing the network size (i.e. the number of sensors). However, the incremental behaviour of the final coverage descends as the network size increases. In Figure 2 a leap is seen from 10 to 30 sensors in DSSA, SODA and SSND algorithm.

In the SSND algorithm, an angle of the coverage function's slope (the leap from 10 to 30 sensors) is $\approx \frac{\pi}{4}$ while in SODA

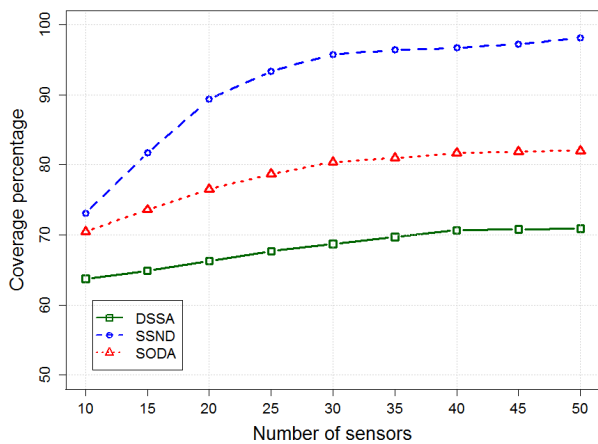


Fig. 2: The area coverage in a 10 x 10 region

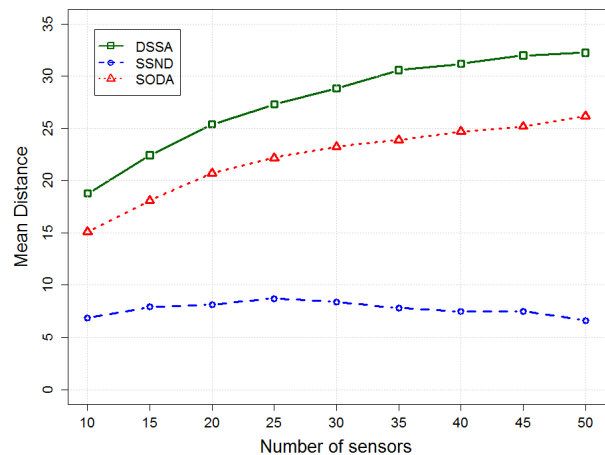


Fig. 4: Mean distance of sensor movement in a 10 x 10 region

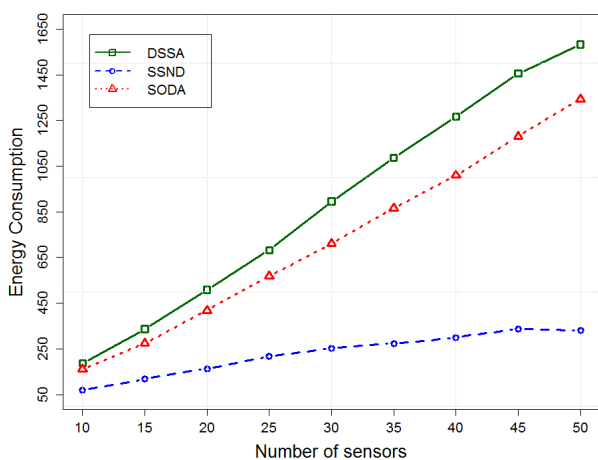


Fig. 3: Total energy consumption in a 10 x 10 region

and DSSA is $\approx \frac{\pi}{7}$ and $\frac{\pi}{18}$, respectively. The slope of this leap in SSND algorithm is almost as twice as SODA that shows a faster improvement in coverage percentage. The SSND algorithm overtakes the 10% improvement of the SODA, in compare to DSSA, with 25% improvement in total coverage percentage.

2) *Energy consumption*: Figure 3 shows the total energy consumption of the sensors. The energy consumption is calculated based on the total distance that sensors take from their origins location to their final location. Considering every unit of the distance takes 1j energy, the total energy consumption is equal to the total distance of sensors.

Figure 3, shows that the total energy consumption increases linearly by the network size (i.e. the number of sensors). However, the increase rate of the total energy consumption for the SSND algorithm is much lower in comparison with DSSA and SODA. The improvement of the SSND algorithm is more than 75% in compare to DSSA while SODA has achieved 25% less energy consumption.

The linear behaviour of Figure 3 shows the possibility of the same energy consumption for a sensor in different number of sensors scenario. In Figure 4 the mean distance of sensors for

DSSA, SODA and SSND algorithms are shown in different number of sensors scenario to investigate the energy usage of a sensor in different number of sensors in the area. In DSSA, and SODA the mean distance of a sensor node increases as the number of sensors in the area increases. However, in SSND algorithm, the mean distance for a sensor increases to the point (saturated point) that the environment is saturated by the sufficient number of sensors. After the saturated point, the increase of the number of sensors does not affect the coverage as sufficient number of sensors can cover the area with their smart movements. Therefore, the mean distance decreases as more number of sensors can cover an area more quickly and with less distance to be taken by every sensor nodes in compare to collective movement algorithms.

3) *Total number of moved sensors*: The total energy consumption does not specify the number of moved sensors that move in the whole process. Therefore, the consumed energy can be used by a sensor that takes a long distance or equally distributed to all the sensors. The number of moved sensors shows the distribution of energy usage in the network which is vital to monitor the stability of the network. In Figure 5, as increasing the network size, the number of moved sensors increase in DSSA and SODA. Accordingly, in DSSA and SODA, all the sensors move, even when the area is saturated, and there is no need for all the sensors to move. Whereas, in the SSND algorithm the number of moved sensors remain the same by increasing the number of sensors. The intelligent behaviour of SSND shows that a sensor moves only if its movement improves the performance of the network, coverage, and the chances for a sensor to move and improve the coverage decrease as the number of sensors in the area increase. Therefore, only needed number of sensors move to cover the area.

VI. CONCLUSION

Area coverage is one of the most fundamental issues in WSNs. The coverage is influenced by the sensor technical specification, the network topology, and most importantly by the sensor deployment algorithm. An optimal coverage

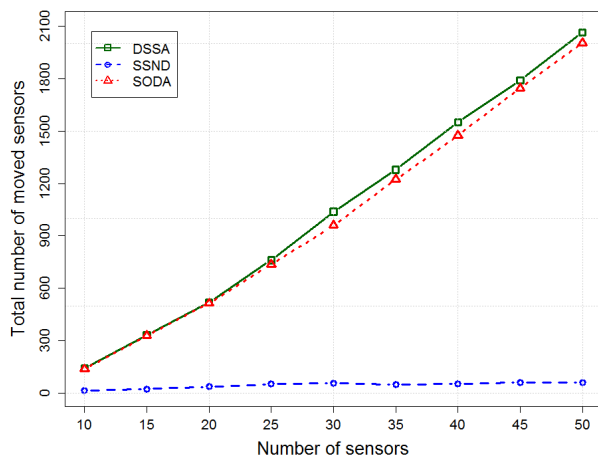


Fig. 5: The number of moved sensors in a 10 x 10 region

provided by a deployment algorithm in any WSN can lead to high overall performance of the network.

Inspired by the equilibrium of molecules, a family of deployment algorithms, such as DSSA and SODA, have been recently proposed to maximise the area coverage. However, the achieved coverage is very costly in terms of the energy consumption and incurred latency. This is mainly due to the movement of almost all nodes at each step and also because of using obsolete information to calculate the next point of movement by each node.

In this paper, we proposed a Smart Self-organization Node Deployment (SSND) for mobile sensor networks to overcome the collective movement algorithms limitations. The SSND algorithm provides full coverage independence of the initial location of sensors, and the initial coverage with significant decrease in energy consumption. The improved performance is achieved by avoiding the collective movement of sensors while smart sensors use reliable information to decide about their next locations.

In our performance study, the DSSA, SODA and SSND algorithms have been simulated using Matlab, and their final percentage of coverage, the total energy consumption, and the number of moved sensors have been compared against each other. The obtained simulation results confirm the advantages of SSND to achieve a more uniform distribution of nodes after applying the algorithm and hence a better coverage. The SSND solution has improved the final percentage of coverage by 25% and the energy consumed is reduced by more than 75%. This is mainly attributed to the fact that one node at each neighbourhood is enough to move while using reliable information at each time step.

REFERENCES

[1] H. Smeets, C.-Y. Shih, T. Meurer, and P. J. Marrón, "Demonstration abstract: a lightweight, portable device with integrated usb-host support for reprogramming wireless sensor nodes," in *Proceedings of the 13th international symposium on Information processing in sensor networks*, pp. 333–334, IEEE Press, 2014.

[2] H. Wu, L. Cheng, C. Wu, and L. Chen, "Robot assisted maintenance strategy in wireless sensor networks," in *Computer Science & Education (ICCSE), 2012 7th International Conference on*, pp. 285–288, IEEE, 2012.

[3] D. Tao and T.-Y. Wu, "A survey on barrier coverage problem in directional sensor networks," *IEEE sensors journal*, vol. 15, no. 2, pp. 876–885, 2015.

[4] T. J. Chowdhury, C. Elkin, V. Devabhaktuni, D. B. Rawat, and J. Oluoch, "Advances on localization techniques for wireless sensor networks: A survey," *Computer Networks*, vol. 110, pp. 284–305, 2016.

[5] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[6] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization.," *IEEE Trans. Signal Processing*, vol. 62, no. 6, pp. 1424–1437, 2014.

[7] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, vol. 27. Princeton University Press, 2009.

[8] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, vol. 48, no. 6, pp. 988–1001, 2003.

[9] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[10] G. Shi and Y. Hong, "Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies," *Automatica*, vol. 45, no. 5, pp. 1165–1175, 2009.

[11] C. Costanzo, V. Loscrí, E. Natalizio, and T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network," *Computer Communications*, vol. 35, no. 9, pp. 1047–1055, 2012.

[12] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 301–311, 2013.

[13] M. Erdelj, V. Loscri, E. Natalizio, and T. Razafindralambo, "Multiple point of interest discovery and coverage with mobile wireless sensors," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2288–2300, 2013.

[14] Y. Zhang and L. Wang, "A distributed sensor deployment algorithm of mobile sensor network," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pp. 6963–6968, IEEE, 2010.

[15] A. V. Savkin, T. M. Cheng, Z. Xi, F. Javed, A. S. Matveev, and H. Nguyen, *Decentralized coverage control problems for mobile robotic sensor and actuator networks*. John Wiley & Sons, 2015.

[16] G. Flierl, D. Grünbaum, S. Levins, and D. Olson, "From individuals to aggregations: the interplay between behavior and physics," *Journal of Theoretical biology*, vol. 196, no. 4, pp. 397–454, 1999.

[17] K. Warburton and J. Lazarus, "Tendency-distance models of social cohesion in animal groups," *Journal of Theoretical Biology*, vol. 150, no. 4, pp. 473–488, 1991.

[18] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005.

[19] M. Sadeghi Ghahroudi, A. Shahrabi, and T. Boutaleb, "An efficient self-organizing node deployment algorithm for mobile sensor networks," in *UBICOMM 2018, The Twelfth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, IARIA, 2018.

[20] A. Baranzadeh and V. Nazarzehi, "A decentralized formation building algorithm with obstacle avoidance for multi-robot systems," in *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on*, pp. 2513–2518, IEEE, 2015.

[21] C. Coulomb, "First memoire on electricity and magnetism. second memoir on electricity and magnetism. third memoir on electricity and magnetism," *History of the Royal Academy of Sciences*.