IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 2, No 2, March 2014
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

DP-UREA: An Algorithm for Autonomous Reorganization of Mobile Sensor Nodes to Improve Coverage

Manisha J. Nene[1], Rajendra S. Deodhar[2], Lalit M. Patnaik[3]

[1]Department of Computer Engineering, Defence Institute of Advanced Technology DIAT, Pune, India 411025.
[2] Armament Research & Development Establishment ARDE, Pune, 411021,
[3] Indian Institute of Science IISc, Banglore, 560012

Abstract

Stochastic deployments of mobile Wireless Sensor Network WSNs are practical in harsh fields, disaster areas, toxic urban regions or hostile environments where human reachability or intervention is impossible, and WSN deployment cannot be performed manually. These deployments may not always lead to effective coverage or; result in a connected network. Hence, distributed deployment schemes where sensor nodes are configured with self-organising and self-adaptive capabilities bring in profound advantages to realise autonomous redeployments of WSNs.

In this paper, we present a Distributed and Parallel- Uncovered Region Exploration Algorithm DP-UREA; for autonomous reorganization of WSN after initial stochastic deployment. Execution of DP-UREA is completely distributed in nature; and all sensor nodes execute the procedure in parallel to self-organise. In DP-UREA, each sensor computes uncovered region around itself and computes a new location to self-displace; with an aim to improve the network coverage locally. All these stochastically deployed sensors simultaneously and cooperatively carry out the movements locally; to achieve the global goal.

Extensive simulation results of DP-UREA and its comparison with other deployment algorithms show that, the performance of D-UREA is efficient with respect to communication and computation; and versatile enough to suit the pervasive needs of cyber-physical systems.

Categories and Subject Descriptors: *[COMPUTER-COMMUNICATION NETWORKS]: Wireless Sensor Network, Autonomous, Self-Organisation, Mobility, Topology, Coverage, Deployment*

Keywords: *Sensor Networks, Redeployment, Self-Adaptive, Coverage, Deployment Algorithm, Mobility, UREA*

1. Introduction

Autonomous networks deployed using WSNs are distributed amorphous computing environments that consist of a large number of nodes and communication links, subject to intermittent failure, likely destruction and limited power resources. Human configuration and control of the entire system is not always possible. An unreachable and dynamic WSN which includes failure-prone nodes will require strategies that are as simple as possible in computations and local communications, to facilitate self-organization. Self-organization is a process of autonomous formation of network connectivity, network addressing and routing structures. Also, the term self-organization in the context of WSNs is to incorporate the capabilities of self-healing to satisfy the application needs. Self-organization of WSN is challenging because of the tight constraints on the bandwidth and energy resources available. Distributed environment and mobility capabilities contribute to the autonomous deployment of WSNs to form pervasive cyber-physical systems. It is important to decentralize these systems without sacrificing its robustness and accuracy.

Mobility of sensors or mobility in target/event introduce new challenges in collaboration management of sensors and information processing in the deployed sensor network. Also, mobility capability can be utilised to benefit the desired placement of sensors in the FoI to achieve various objectives with respect to improvement in network coverage, connectivity and communication path detection.

Coverage is one of the performance evaluation metrics of WSNs. It represents *how well a Field of Interest FoI is monitored by a set of sensors* or *how effective is the sensor network in detecting the intrusion of the objects* in the FoI. Coverage metrics can be considered as a measure of Quality of Service QoS. Coverage achieved with the specific topology is inherently one of the most fundamental essential system parameter.

The application needs prominently dictate the sensor deployment schemes. Deterministic or stochastic deployment schemes for the sensors pose a variety of questions like the optimal number of sensors required to full cover the given Field of Interest FoI, required redundancy of sensor nodes to improve the network lifetime, fault management and many more. While improving network coverage, additional requirements like communication connectivity, K-connectivity, K-coverage, best case and worst case coverage, network lifetime, etc. dominate the decision making for the deployment algorithms. In stochastic deployments, mobility feature of the mobile ad hoc network can be best exploited to reposition the sensors so as to achieve the desired network coverage. Efficient sensor *deployment schemes* are necessary to minimise cost and yet achieve mandated levels of coverage.

In centralised deployment schemes [1] [2], all the sensor nodes are governed and instructed by the centralised server/base station for their reorganisation and relocation in the FoI. These schemes are possible in friendlier environments or FoI where *a priori* information of terrain is available.

In distributed deployment schemes [3] [4] [5] [6], a sensor node takes displacement decision based on the information that is locally available to it within its communication range. There is no central governing server/base station. These schemes are practical in harsh fields, disaster areas, toxic urban regions or hostile environments where human reachability or intervention is impossible, and if the deployment cannot be performed manually. To scatter the sensor nodes by throwing or dropping using aircraft/UAVs are some of the possible options to achieve random/stochastic deployments using WSNs. Further, at a given point of time, when more than one sensors displace/reorganise simultaneously, in the FoI based on the local information available to them within their communication range can be viewed as parallel deployment schemes.

In [2], we have proposed a centralised deployment algorithm UREA: Uncovered Region Exploration Algorithm for Reorganization of Mobile Sensor Nodes to Maximize Coverage. A distributed algorithm D-UREA employing the same technique and using a single random seed selection strategy is presented in [6].

In this paper, we propose a new distributed deployment algorithm DP-UREA for autonomous redeployment of mobile WSNs after stochastic deployment. In this algorithm, each sensor node explores uncovered region around itself to decide its movement. Further, we describe the deployment scheme to execute DP-UREA in parallel. Using DP-UREA, speed-up in the deployment time is achieved as compared to D-UREA [6].Extensive simulation results are presented to demonstrate its consistency and robustness. Finally, the results are compared with the algorithms proposed to achieve the same objective.

The outline of the paper is as follows: Algorithm DP-UREA is presented in section 2. The deployment scheme is described in section 3. The analysis of the simulation results are presented in section 4 and section 5 concludes the paper.

2. DP-UREA: ALGORITHM

2.1 Preliminaries

Localization Techniques
Location awareness is crucial for WSNs since many applications such as environment monitoring and target tracking depends on knowing the locations of sensor nodes. Due to the ad hoc nature of such networks, each node must determine its location through a location discovery process. For outdoor systems, the Global Positioning System GPS [7] can be used. GPS may not be cost effective or work well indoors.

Many techniques have been proposed to enable each node to determine its location indoors with only limited communication with nearby nodes. Most of these methods exploit received signal strength [8], time difference of arrival of two different signals [9] and angle of arrival [10]. The detailed discussion of these techniques is provided in [11]. In this paper, we assume that the sensors know their locations.

Path Planning
In systems that exploit mobility capability of sensors, finding paths on which these mobile sensors can move to desiring destinations, especially when there exist obstacles or uneven surface in the FoI, is an important concern. This is been studied in the area of robotics [12], [13]. Li et al. [14] studied the problem in sensor networks. They combined the above methods to find the best motion path and modified them to exploit the distributed nature of sensor networks. In this paper, we do not study this problem further; we assume that mobile sensors can move to any location where they are asked to move based on the existing techniques. However, in [15] it is shown that the proposed Uncovered Region Exploration technique is successfully applied in obstacle regions and priority-based regions.

Displacement Model
In systems that exploit mobile sensors, the sensor nodes are provided with the mechanical devices, which determine the velocity with which they move in the FoI and the energy consumed per unit displacement. In this paper, we assume that the mobile sensor nodes are provided with the same mechanical devices, so they all move with the same velocity to the expected new location in the FoI; and have a constant unit of energy utilisation for a unit displacement.

Sensing Model

Each type of sensor has its unique sensing model characterized by its sensing area, resolution and accuracy. The sensing area depends on multiple factors such as strength of signals generated at the source, the distance between the source and the sensor, the attenuation rate in propagation and the desired confidence level of sensing rate of the sensing level.

For instance, consider an application in which a network of acoustic sensors is deployed for detecting mobile vehicles [16]. Due to signal attenuation, sensors closer to a vehicle can detect higher strength of acoustic signals than sensors farther away from the vehicle and, thus, have higher confidence for detecting the vehicle. Therefore, given a confidence level, we can derive a sensing range surrounding each sensor. In this paper, we only consider the isotropic sensing models. Each sensor node is associated with a sensing area which is represented by a circle with the same radius.

Sensing Range and Communication Range

In distributed environment for deployment algorithms, a sensor can exchange location information by broadcasting. Hence, only the sensors that are within its communication range will acknowledge. The uncovered region exploration parameter is computed by it using the acknowledgements received from the neighbours. In this paper, it is assumed that the communication is lossless.

2.2 Algorithm DP-UREA

Notations & Assumptions

Consider a field of Interest (FoI), with a superimposed grid, given by $F = \{(i,j): 1 \le i \le m, 1 \le j \le n\}$. We assume that k sensors $S_1, S_2, \dots, S_k$ are randomly deployed in the *FoI*. We further assume that the locations of the sensors coincide with the grid points in the FoI, and are known to each sensor through some localization technique. All the deployed nodes have sensing range equal to r_s and communication range r_c units.

We assume binary model for sensing by a sensor. A sensing circle of a sensor S located at $(x,y) \in F$ is given by

$$C(S) = \{(i,j) \in F: \sqrt{(i-x)^2 + (j-y)^2} < r_s\}.$$

The set of grid points in the FoI, which lie just outside the sensing circle of S, is denoted by $N(S)$ and is given by

$$N(S) = \{(i,j) \in F: r_s \le \sqrt{(i-x)^2 + (j-y)^2} < r_s + 1\}.$$

Let G denote the coverage grid matrix with the dimensions same as that of FoI and $G(i,j) = 1$ if grid point (i,j) is within the sensing circle of some sensor. Coverage of the network is estimated by the number of covered grid points and is given by $cov = \frac{\sum_{i,j} G(i,j)}{mn}$. $Neighbour(S)$ denotes the set of neighbour nodes of sensor S which lie within the communication range of S. For sensor S located at (x,y), its local window is given by

$$W(S) = \left\{ \begin{array}{c} (i,j) \in F: \\ \max(1, x - r_c) \le i \le \min(1, x + r_c), \\ \max(1, y - r_c) \le j \le \min(1, y + r_c) \end{array} \right\}$$

$LG(S)$ represents the local coverage grid matrix of a sensor node S and is of size of $W(S)$. Let $U(S) = \{(i,j) \in F: (i,j) \in N(S) \text{ and } LG(i,j) = 0\}$ be the set of uncovered grid points just outside the sensing circle of S. For $S_i \in N(S)$, let $U(S_i, W(S))$ denote the set of uncovered grid points around S_i within $W(S)$, as estimated by S.

Algorithm DP-UREA

Input: k randomly deployed sensors $S_1, S_2, \dots, S_k$ with their initial positions (x_i, y_i) for $i = 1, 2, \dots, k$. Sensing radius r_s of each sensor, FoI F with dimensions $m \times n$.
Output: Final sensor positions $(x_{i(new)} y_{i(new)})$ for $i = 1, 2, \dots, k$.

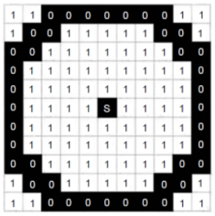
```
1.  Start: Initial stochastic deployment of k mobile nodes in the FoI F;
2.  for t=1 to number of rounds
3.  Communication phase
        3.1  Each node broadcasts its location
        3.2  Each node records  the locations of its neighbours
4.  Computation phase at each node
        4.1  Construct a local grid matrix $LG(S)$ based on the
             information received during communication phase
        4.2  Compute $N(S)$ and $U(S)$.
        4.3  Compute new location for itself.
        4.4  Check the boundary conditions for the new
             location.
        4.5  Based on the information available, compare the
             local coverage at the new location with the
             present local coverage.
        4.6  If there is improvement in the local coverage and
             boundary conditions are satisfied, new location is
             accepted.
5.  Movement phase at each node
        5.1  Move to the new location if found feasible
6.  end for
7.  Resume application specific task
8.  end
```

Details of steps 4.2 to 4.5 are given below.

Step 4.2: Computation *of* $N(S)$ and $U(S)$

$N(S)$ and $U(S)$ are calculated with the help of a data structure: Uncovered Region Exploration. Evaluation and update are the two computational operations performed on this data structure. At any given time, $U(S)$ is computed as follows.

Define a window of size $(2r_s + 1)$. Initialise the window to ones. Set appropriate locations of this window to zero as shown in Figure 1, to indicate $N(S)$ if S is placed at the centre of the window. Place the window on the local coverage grid so as to coincide its centre with sensor S. The positions of zeros in the window that coincide with zeros on the coverage grid; provide uncovered region exploration parameter $U(S)$. The computational complexity to evaluate and update $U(S)$ is $O({r_s}^2)$.



1	1	0	0	0	0	0	0	0	1	1
1	0	0	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	S	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	0
1	0	0	1	1	1	1	1	0	0	1
1	1	0	0	0	0	0	0	0	1	1

Fig.1 Window used for $U(S)$ computation, $r_s = 5$ units

Step 4.3: Computation of new location
S located at $(x, y) \in F$ computes new location co-ordinates (x_{new}, y_{new}) based on uncovered region $U(S)$ computed in step 4.2.

a) Compute $F_x = \sum_{(i,j)\in U(S)}(i - x)$ and $F_y = \sum_{(i,j)\in U(S)}(i - y)$

b) Define new locations of S as

$$x_{new} = x + round\left[\frac{F_x}{F_x + F_y}\right] \quad \text{and}$$
$$y_{new} = y + round\left[\frac{F_y}{F_x + F_y}\right]$$

where round [] function returns the nearest integer.

Step 4.4: Checking boundary conditions
After computation of the new location, each node confirms that new locations are not outside FoI

a. If $x_{new} < 1$ then $x_{new} = 1$ and $x_{new} > m$ then $x_{new} = m$.
b. If $y_{new} < 1$ then $y_{new} = 1$ and $y_{new} > n$ then $y_{new} = n$.

Computational complexity to compute (x_{new}, y_{new}) is O $({r_c}^2)$.

Step 4.5: Confirmation of new locations
A node S accepts new locations (x_{new}, y_{new}) if improvement in the local coverage is expected.

1. Let $count_1 = |U(S)|$ be the count of uncovered grid points around sensor node S.
2. Let $count_2 = |U(S)_{new}|$ be the count of uncovered grid points around proposed new location of S; (x_{new}, y_{new}).
3. If $count_1 \geq count_2$ assign new sensor locations (x_{new}, y_{new}) to sensor S.

Figure 2 gives the global view of the execution of DP-UREA:

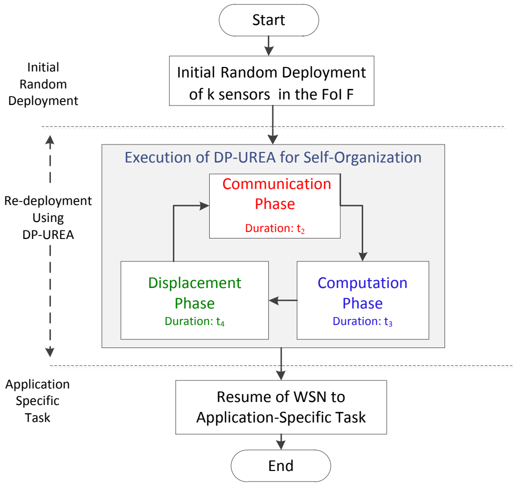


Fig. 2 DP-UREA

The DP_UREA deployment algorithm is completely distributed in nature. A sensor self-displaces only when it identifies an uncovered region around it within its local window. Decision to self-displace is taken by the sensor nodes and their collective action improves the coverage. Due to the distributed nature of execution of DP-UREA; autonomous redeployment WSN is achieved.

The DP-UREA protocol enables parallel execution of the deployment algorithm. On elapse of time t_1 after the initial random deployment, all the sensors participate in the redeployment. All the sensors communicate within their local window to locate neighbours, compute uncovered region around themselves, propose new location to improve the network coverage locally and self-displace. All these activities take place at each node simultaneously.

3. DP-UREA: IMPLEMENTATION ASPECTS

DP-UREA deployment algorithm runs iteratively. In each round, sensors first broadcast their locations and identify their neighbours. They determine the existence of coverage holes by evaluating the data structure *uncovered* *region* around themselves and compute a new location for self-displacement to eliminate or reduce coverage holes. These states enable the self-adaptable capability in each participating sensor node.

During the execution of the algorithm, each sensor node attains various states. A node remains in that state for a specific period of time. The time is estimated based on the evaluated computation and communication complexity. Various states attained by the nodes and the corresponding execution times are given in Table 1.

TABLE 1 STATES OF SENSOR NODES DURING RE-DEPLOYMENT

State Name	Description
<Ready State,t_1>	Randomly deployed sensor nodes that are ready to participate in redeployment process; Node remains in this state from time they are set for deployment till the scheduled time of redeployment.
<Communication State,t_2>	State of a sensor node when it broadcasts its own location and receives information from the neighbors. This state lasts for t_2 clock ticks.
<Computation State, t_3>	State of a sensor node when it computes new locations based on uncovered region around it. This state lasts for t_3 clock ticks.
<Self-Displacement State,t_4>	State of a sensor node when it self-displaces to new computed locations. This state lasts for t_4 clock ticks.
<Release State>	A sensor node quits and terminates its participation in the redeployment process. Resumes to Application-Specific task.
<Sleep State>	A sensor node with its energy level below the energy threshold

The DP-UREA code exists on each node with preconfigured values of t_1, t_2, t_3, t_4 and ROUNDS. These values are upper bound estimations of expected communication, computation, displacement and convergence time.

The ability to self-adapt is realised when the sensors attain various states to perform the assigned tasks. The protocol to self-adapt during reorganisation using DP-UREA is as described below:

DP-UREA Protocol at Sensors S

```
Initial Random Deployment Process
    1. Deployment: Initial Random Deployment Phase
       Set Sensor state to <Ready State, t1>; at the end of $t_1$ clock clicks change state to <Communication State, $t_2$>

    Start: DP-UREA
    2. DP-UREA Communication Phase
       on attaining <communication state, $t_2$>
           a. Sensors broadcast their locations within $r_c$.
           b. Sensors record the location information of their respective neighbours.
           c. On elapse of $t_2$ clock clicks sensor attain <Computation State, $t_3$>
    3. DP-UREA Computation Phase
       On attaining <Computation State, $t_3$>
           a. Sensors construct their local window(S) of size $2r_c+1$ based on the neighbour information
           b. Compute the coverage $Cov_{initial}$
           c. Compute uncovered region around it U(S)
           d. Compute new locations $(x_{new},y_{new})$
           e. Checks the boundary conditions
           f. Compute the expected coverage. $Cov_{final}$
           g. If $(Cov_{final}>Cov_{initial})$ accept $(x_{new},y_{new})$

           h. Increment the iteration counter
           i. After elapse of $t_3$ clock clicks change the state to <Self-Displacement State,$t_4$>
    4. DP-UREA Displacement Phase
       On attaining <Self-Displace State, $t_4$>
           a. Self-displace from (x,y) to $(x_{new},y_{new})$
           b. Maintain the displacement count
           c. If iteration count=ROUNDS change state to <Release State> else change state to <communication state, $t_2$>
    End: DP-UREA
```

Deployment Time

An upper bound clock tick time is set whenever the participating sensors attain important phases of redeployment.

1. The time set to achieve initial random/stochastic deployment is t_1. Here, the deployed sensors are clustered or sparsely placed in the FoI, leading to uneven network coverage. This is an initial one-time cost.
2. Participating sensor nodes remain in communication phase for duration t_2.
3. The participating sensors compute new locations using the heuristics uncovered region around it, in t_3 time clicks.
4. On confirmation of new locations, each sensor self-displaces a unit distance within t_4 time clicks.

Hence, using DP-UREA; at a single iteration/ROUND; every participating sensor self-displaces in $(t_2+t_3+t_4)$ time. The total taken to self-displace all the sensor nodes at the end of execution of DP-UREA is:

$$t_{DP\ UREA} = (t_2+t_3+t_4) * (Rounds)$$

In D-UREA [6], at a single iteration/ROUND; only one sensor node which is a seed node self-displaces. Therefore, the total time to self-displace all the sensor nodes is:

$$t_{D\ UREA} = (t_2 + t_3 + t_4) * (Rounds)(Total\ no.of\ sensor)$$

Termination

In some applications, the coverage requirement may be met without achieving the maximum coverage. In these cases, it may be prudent to terminate the deployment process before the maximum coverage is reached to save power and reduce deployment time. The value of ROUNDS is decided based on the analysis of convergence, discussed in section 4.

4. PERFORMANCE EVALUATIONS

We analyse the performance of our deployment scheme for the following aspects: deployment quality with respect to the improvement in the *network coverage*, performance consistency and the *robustness*.

Deployment quality is measured by the network coverage and the time (number of rounds) to reach this coverage.

Maximum network coverage is achieved, if the improved coverage reaches the mathematical upper bound using the given number of k sensors; computed as $min(1, \pi r_s{}^2 k/(m \times n))$.

Deployment time is determined by the number of rounds needed and the time of each round. The duration of the each round is primarily determined by the moving speed of the sensors, which is a mechanical attribute of sensors. Thus we use number of rounds to measure the deployment time.

Study 1: To observe the improvement in the network coverage using DP-UREA.

Simulation Setup: We consider $k = 20$ randomly deployed sensor nodes in the FoI of size 50×50 sq. units to run DP-UREA. We assume $r_S = 5$ units and $r_C = 3r_S$.The improvement in the network coverage is observed by taking average of 30 simulation runs.

The average improvement in the network coverage is 99.86176 % within 11 rounds. The performance is consistent and robust as the standard deviation (σ) in its performance is 0.002. The displacement is not more than 4.75 units.

In Fig. 3 the coverage obtained using DP-UREA; as a fraction of maximum achievable coverage for 20 sensors is plotted against the time/iterations. It is observed that the algorithm quickly converges to the mathematical upper bound of the network sensing coverage.

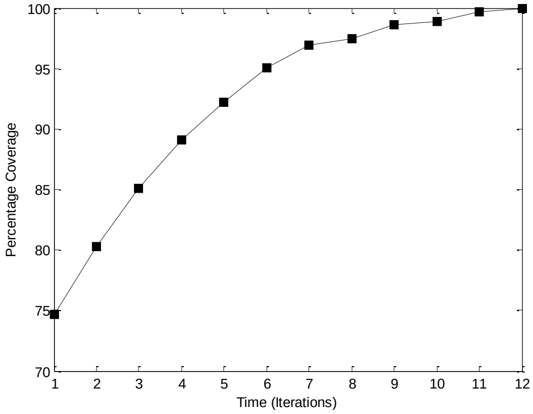


Fig. 3 DP-UREA: Average Coverage (k=20)

Figure 4 shows that the standard deviation (σ) in sensing coverage during each iteration. For 20 nodes the standard deviation (σ) is 0.002. It is observed that the value of (σ) rapidly comes closer to zero as the number of iterations increase before termination. This establishes a fact that DP-UREA consistently converges to the expected solution.

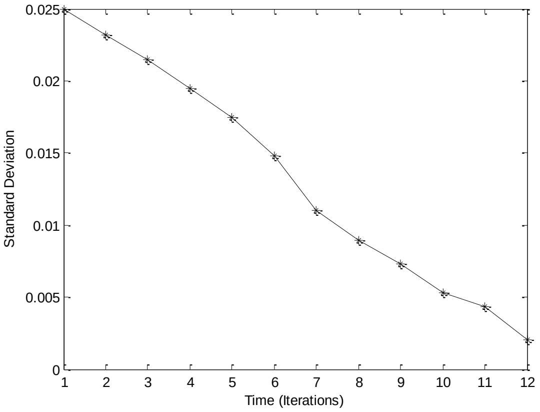


Fig. 4 DP-UREA: Standard Deviation at each Iteration (k=20)

Study 2: To observe the scalability of DP-UREA

Simulation Setup: We consider 40 to 100 randomly deployed sensor nodes, with uniform distribution, in the FoI of size 50×50 sq. units to run the DP-UREA. We assume $r_S = 5$ Units and $r_C = 3r_S$. The performance is analysed by taking average of 30 simulation runs.

Figure 5 shows the improved coverage obtained for sets of sensor nodes from 40 to 100. The coverage obtained using DP-UREA varies from 0.966 for 40 nodes (approximately 0.88 using VFA, 0.92 using CLP) to 0.999 for 100 nodes (approximately 0.95 using VFA, 0.989 using CLP) [1][3]. It is observed that the coverage obtained using DP-UREA is significantly improved.

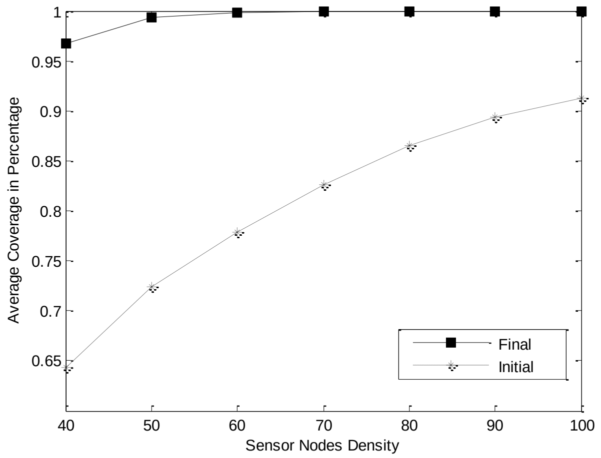


Fig. 5 DP-UREA: Average Coverage

Figure 6 shows that the standard deviation (σ) in the improved coverage obtained in 30 various sets of simulation runs for 40–100 sensor nodes using DP-UREA. The results show that the standard deviation (σ) for 40 nodes using DP-UREA is 0.0006, whereas that in VFA is 0.02. Also, (σ) for 100 nodes using DP-UREA is 0.0001 which is far less than that in VFA, i.e. 0.01.

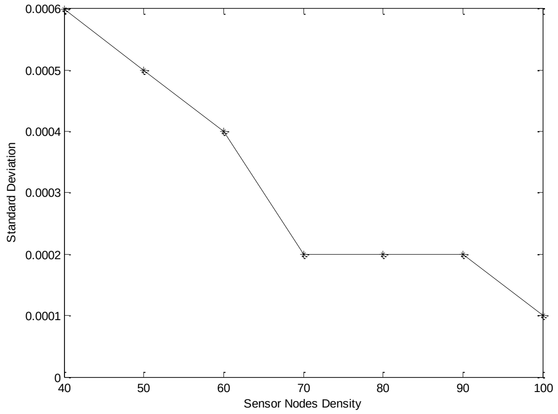


Fig. 6 DP-UREA: Standard Deviation

Smaller values of the standard deviation (σ)in the sensing coverage indicate that DP-UREA is more stable in computing sensing coverage.

Figure 7 shows the average displacement distance required to improve sensor field coverage. It is observed that the average displacement of each node for 40 and 100 sensor nodes is 6.34 and 0.97 units, respectively.

Using VFA, the average displacement of each node for 40 and 100 sensor nodes was 9.2 and 8.4 units, respectively [1][3]. Using VEC, VOR, Minimax [4] the displacement of each node for k=140 in FoI $100x\ 100$ is 4.63 (rounds=12), 4.61 (Rounds=11) and 4.15 units (Rounds=11), respectively; and using DP-UREA is 4 units (Rounds 07).

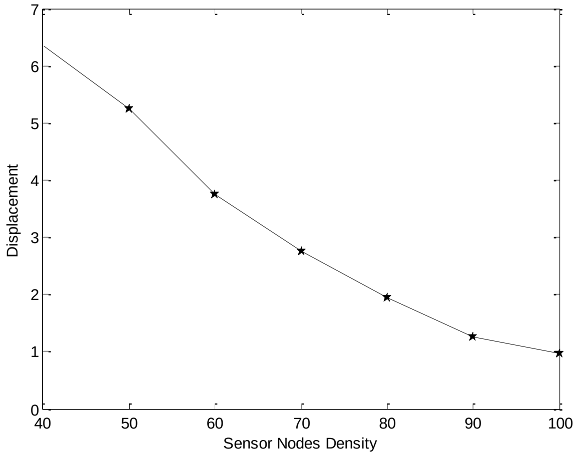


Fig. 7 DP-UREA: Average Displacement

Study 3: To observe the robustness of DP-UREA

Simulation Setup: We consider $40 \text{ to } 100$ randomly deployed sensor nodes, with uniform distribution, in the FoI of size 50×50 sq. units to run the DP-UREA. We assume $r_S = 5 \text{ Units}$ and $r_C = 3 r_S$. The performance is analysed by taking average of 30 simulation runs.

Observations based on the obtained simulation results, as presented in Figure 6, 7 and 8 demonstrate that:

1. DP-UREA quickly converges to the mathematical upper bound of the network sensing coverage. Ref. Figure 3, 5.
2. The value of (σ) rapidly comes closer to zero as the number of iterations increase before termination. This establishes a fact that DP-UREA consistently converges to the expected solution. Ref. Figure 4.
3. Smaller values of the standard deviation (σ) in the computation of sensing coverage using DP-UREA indicate that it is more stable in computing sensing coverage. Ref. Figure 6.

Study 4: Analysis of convergence

The analysis of convergence of DP-UREA for the value of ROUNDS is assigned as follows:

Case 1:

If ($NetworkCoverage = ((\pi r_s{}^2 k)/(mn))$ then for binary detection model, the upper bound value for coverage is $min(NetworkCoverage, 1)$ where k is the total number of sensors deployed, r_s is the sensing radius and FoI is (m × n) sq. units. The algorithm terminates when the upper bound criterion is met. Incorporating a check for termination criteria at an individual sensor node level will increase the computational overheads.

Case 2:
From Figure 8, it is observed that for initial few iteration of control transfers the coverage rapidly increases. As the improvement in the coverage comes closer to the mathematical upper bound, the number of control transfers required to obtain small improvement in the coverage is significantly large. Each control transfer consumes energy for computation and possible displacement. Hence algorithm was terminated when 95% of the possible coverage is achieved.

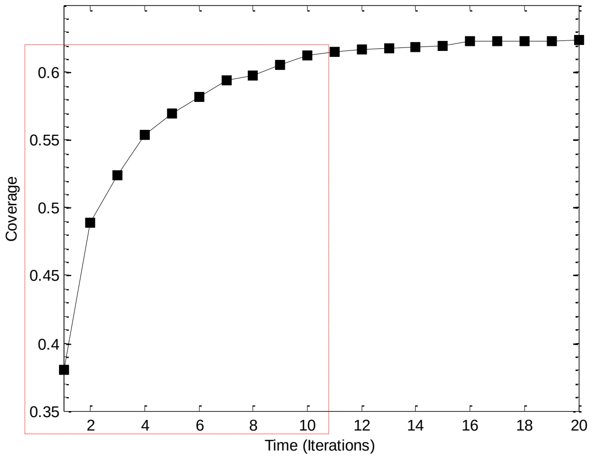


Figure 8 DP-UREA Average Coverage at Individual Iteration, k=20

Average number of control transfers required to achieve 95% of the possible coverage for $k = 20$; hence, the estimation of the value for ROUNDS is 11. Similarly, with the required density of sensor nodes; estimation of the value for ROUNDs can be done.

It is observed that when the network contains large number of sensor nodes, small numbers of control transfers are needed because initial coverage is high. Also, when number of nodes is less, the algorithm terminates after a small number of control transfers as non-overlapping deployment is quickly achieved.

For practical application of the algorithm the value of ROUNDs can be set based on the value obtained in simulation.

Study 5: Performance comparison of DP-UREA with the existing state-of-art.

In Table 2, the performance of DP-UREA is compared with the existing state-of-art, with similar simulation platform and scenarios.

TABLE 2 DP-UREA: PERFORMANCE COMPARISON

Parameter	Type of Deploy-ment	r_s	Percentage Cover-age	Round	σ
Comparison of DP-UREA & VFA, CLP [1][3], $k = 20, FoI = 50x50, r_s = 5$					
DP-UREA	Distributed and Parallel	$3r_s$ $2r_s$	100%100%	09 10	0.0002 0.0002
VFA	Centralised	*	100%	28	*
CLP	Distributed	$3r_s$ $2r_s$	79.61% 76.43%	20	*
Comparison of DP-UREA, VFA & CLP [1][3], $k = 50, FoI = 50x50 r_s = 50$					
DP-UREA	Distributed and Parallel	$3r_s$ $2r_s$	99.9% 99.9%	06 07	0.0002 0.0002
VFA	Centralised	*	92%	*	*
CLP	Distributed	$3r_s$ $2r_s$	98.9% 97%	40 48	*
Comparison of DP-UREA, SI [ ], $k = 09, FoI = 100x100, r_s = 16$					
DP-UREA	Distributed and Parallel	$3r_s$ $2r_s$	100% 100%	07 08	#
SI	Distributed	-	80%	4462	*
Comparison of DP-UREA, Vor, Vec, Minimax [3], $k = 140\ FoI = 100x100, r_s = 16$					
DP-UREA	Distributed and Parallel	20	99.8%	07	#
Vec	Distributed	20	95.79%	12	*
Vor	Distributed	20	97.46%	11	*
Minimax	Distributed	20	97.87%	11	*

*Not Available NA, # Not Computed

TABLE 3COMPARISONS OF DEPLOYMENT ALGORITHMSPARAMETERS

Parameters	DP-UREA	VFA [1]	CLP [3]	VEC,VOR, Minimax [5]
Heuristics	Uncovered Region	Virtual Forces	Crystal Lattice Formation	Voronoi
Deployment Scheme	Distributed and Parallel	Centralised	Distributed	Distributed
Movement Strategy influenced by	Move towards uncovered region	Impact of forces laid by sensor positions	Hexagonal positions	Towards farthest vertex
Extensible to deployment schemes	Centralised, Distributed, Parallel	Centralised	Distributed	Distributed

The Table 3 compares the parameter with respect to heuristics, deployment schemes, strategy that influence the movement of sensor node etc.

5. SUMMARY AND CONCLUSION

In this paper, we present a Distributed and Parallel-Uncovered Region Exploration Algorithm DP-UREA; for autonomous reorganisation of WSN after initial stochastic deployment. Execution of DP-UREA is completely distributed in nature; and all sensor nodes execute the procedure in parallel to self-organise. In DP-UREA, each sensor computes *uncovered region* around itself and computes a new location to self-displace; with an aim to improve the network coverage locally. All these stochastically deployed sensors simultaneously and cooperatively carry out the movements locally; to achieve the global goal.

Based on the results obtained from the extensive simulations; the following conclusions are made: DP-UREA quickly converges to the mathematical upper bound of the network sensing coverage with the given sensor node density. The value of (σ) rapidly comes closer to zero as the number of iterations increase before termination. This establishes a fact that DP-UREA consistently converges to the expected solution. Smaller values of the standard deviation (σ) in the computation of sensing coverage using DP-UREA indicate that it is more stable in computing sensing coverage.

While employing simple yet efficient technique with reduced overheads of computations and network communications, simulation results clearly show that DP-UREA is consistent and robust. DP-UREA is independent of any tuning parameters, which affects the placement scheme and the computation of network coverage. The proposed *Uncovered Region Exploration* based deployment algorithms are efficiently extended and simulated for centralised UREA [2], distributed D-UREA [6] and distributed-parallel DP-UREA redeployments schemes in WSNs; hence, increases the impact factor of the proposed work.

References

[1] Y. Zou, K. Chakrabarty. Sensor deployment and target localization in distributed sensornetworks, ACM Trans. Embedded Comput. Syst. (TECS) 3(1), 2004, pp. 61–91.

[2] Nene M, Deodhar R, Patnaik L. UREA: An Algorithm For Maximisation Of Coverage In Stochastic Deployment Of Wireless Sensor Networks. International Journal of Parallel, Emergent and Distributed Systems,Taylor & Francis 2012; 27(3): pp. 249–274.

[3] P. Wang, T. Hou, R. Yan, Maintaining Coverage by Progressive Crystal-Lattice Permutation in Mobile Wireless Sensor Networks, IEEE Proceedings of ICSNC, November 1-3, Tahiti, France, IEEE Computer Society Press, ISBN: 0-7695-2699-3, 2006, pp. 42-48

[4] Guiling Wang; Guohong Cao; Tom La Porta, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on* , vol.5, no.6, June 2006, pp.640,652

[5] N. Heo, P.Varshney. A Distributed Self Spreading Algorithm for Mobile Wireless Sensor Networks. IEEE Wireless Comm. and Networking, Piscataway NJ, 2003, pp 1597-1602

[6] Nene M, Deodhar R, Patnaik L. D-UREA: Distributed Uncovered Region Exploration Algorithm For Reorganization Of Sensor Nodes To Maximize Coverage. 10th IEEE International Conference Pervasive Computing and Communications Workshops (PERCOM workshops) 2012 pp. 883–888

[7] "US Naval Observatory (USNO) GPS Operations," http://tycho.usno.navy.mil/gps.html, Apr. 2001

[8] Patwari, N.; Ash, J.N.; Kyperountas, S.; Hero, A.O.; Moses, R.L.; Correal, N.S., "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE* , vol.22, no.4, July 2005, pp.54,69

[9] A. Savvides, C. Han, M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiCom*, 2001. ACM, New York, NY, USA, 166-179

[10] D. Niculescu, B. Nath. Trajectory based forwarding and its applications. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (MobiCom '03). ACM, New York, NY, USA,2003, 260-272

[11] L. Hu and D. Evans. 2004. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (MobiCom '04). ACM, New York, NY, USA, 45-57.

[12] D. Koditschek, "Planning and Control via Potential Functions," *Robotics Rev. I*, 1989. pp. 349-367

[13] Lengyel, M. Reichert, B. Donald and D. Greenberg, "Real-Time Robot Motion Planning Using Rasterizing Computer Graphics Hardware," *Proc. SIGGRAPH*, 1990.

[14] Qun Li, Michael De Rosa, Daniela Rus. Distributed algorithms for guiding navigation across a sensor network. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (MobiCom '03). ACM, New York, NY, USA, 313-325

[15] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *Proc. ACM MobiCom*, 2001. ACM, New York, NY, USA, 139-150

[16] S. Meguerdichian, S. Slijepcevic, V. Karayan, Miodrag Potkonjak. 2001. Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing* (MobiHoc '01). ACM, New York, NY, USA, 106-116.

[17] K. Lee. An Automated Sensor Deployment Algorithm Based on Swarm Intelligence for ubiquitous Environment. IJCSNS International Journal of Computer Science and Network Security, Vol. 7. 12, December 2007