Dynamic Energy Aware Gur Game based Algorithms for Self Optimizing Wireless Sensor Networks

Nitin, *Senior Member, IEEE*
Department of CSE and ICT
Jaypee University of Information Technology,
Waknaghat, Solan-173234, Himachal Pradesh
delnitin@ieee.org

ABSTRACT

This paper presents, Application of Gur Game Based Algorithm on Wireless Sensor Networks (WSNs) deployed to monitor Homogenous and Heterogeneous Grid in order to achieve Quality of Service (QoS) = 0.40 and 0.50. Further, the objectives of all these algorithms are to maximize the coverage of the sensor area while conserving energy consumed by sensor nodes. This is achieved via carefully activating/deactivating the sensors while maximizing the coverage area.

General Terms

Modeling & Simulation, Algorithms

Indexing terms

Gur Game; Wireless Sensor Networks; Quality of Service; Coverage Area, Energy, Homogeneous Grid; Heterogeneous Grid.

Academic Discipline And Sub-Disciplines

Physical Sciences

SUBJECT CLASSIFICATION

Computer Science and Engineering

COVERAGE

Wireless Sensor Networks

TYPE (METHOD/APPROACH)

Experimental

INTRODUCTION AND MOTIVATION

A WSN consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants [1-3] and to cooperatively pass their data through the network to a main location. The development of WSNs was motivated by military applications such as Battlefield Surveillance; today such networks are used in many industrial and consumer application, such as Environmental Monitoring, Habitat Monitoring, Acoustic Detection, Seismic Detection, Military Surveillance, Inventory Tracking, Medical Monitoring, Smart Spaces and Process Monitoring [1-16]. Sensors are low-powered devices with very little computational capability. WSNs allow sensors to work together in order to cover wide areas comprehensively. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes. The WSNs have many limitations such as Hostile Environment, Random topology, Limited Resources, Design Challenges. In this paper, we are specifically talking about the Random Distribution of the sensors in the Homogeneous and Heterogeneous Grids and Power Restrictions (we want to minimize the energy dissipation as much as possible). In order to handle these two open issues we have taken the help of mathematical Gur Game Based Algorithms and its application on WSNs [15, 16].

PROBLEM DESCRIPTION AND CONTRIBUTIONS

In this paper, we have shown the Application of Gur Game on WSNs. More specifically we have shown the (1) Application of Generalized Gur Game based Algorithm on WSNs monitoring 4 x 4 Homogenous Grid to achieve QoS = 0.40 and 0.50. This approach is Application-Specific. (2) Application of Gur Game based Algorithm on WSNs monitoring 4 x 4 Heterogeneous Grid to achieve QoS = 0.40 and 0.50. This approach is Application-Specific and cannot be generalized because of the degree of Heterogeneity, is extremely high.

For our simulation, purpose we have used Java technology moreover, Gur Game is a random approach and can be efficiently simulated with Java Technology. All simulation results are validated through MATLAB.

The rest of the paper is organized as follows. Section 2 describes some of the previous work already done in this area in terms of energy aware algorithms for WSNs. Section 3 describes the Application of Gur Game Based Algorithms on WSNs (Generalized, monitoring Homogenous and Heterogeneous Grid) is supported by the Simulations results from MATLAB and Java Technology and Limitations for Homogenous and Heterogeneous WSNs Design, followed by Section 4 concludes the paper followed by the references.

APPLICATION OF GUR GAME BASED ALGORITHMS ON WIRELESS SENSOR NETWORKS

A random and greedy mathematical paradigm of the Gur Game has been proposed in [6, 7]. This technique allow the nodes to communicate with the base station at a regular interval of time, and base station send feedback in order to manage nodes on basis of packets received. The result is a robust sensor network that allows the base station to dynamically adjust the resolution of a network based on feedback it receives from the sensors in the network. Recently the authors of [15, 16] introduced self optimizing algorithms to regulate the process of activating sensors while maximizing the number regions covered by sensor nodes. They had proposed a dynamic clustering algorithm that employs the concept of connected dominating sets. They also improved the earlier Ants Algorithm and Genetic Algorithm to take into consideration the dynamic nature of WSNs.

In this paper, we report the results obtained by implementing Gur Game Based Algorithms. Gur Game is a mathematical algorithm, was originally introduced by Brian Tung and L. Kleinrock in [6, 7]. Instead of using number of active nodes, we use ratio between coverage area and number of active nodes as QoS parameter. We have learnt the application of Gur Game based algorithm on WSNs and the reward function for simulation purposes from the authors of the paper [15, 16] however; we have taken a different approach to design the algorithm and its application on the WSNs.

Simulation Scenario 1

In order to achieve the QoS = 0.40 (from a Homogenous 4 x 4 Grid having 16 regions) we have should have 16 regions and these 16 regions must be monitored by the 40 sensors as we know that 16/40 = 0.40 and if we take this as a vice versa we will have 40/16 = 2.5 i.e. each region is monitored by the 2.5 sensors. Further, in order to achieve the QoS = 0.40 (from a Heterogeneous 4 x 4 Grid having 15 regions) we have should have 15 regions and these 15 regions must be monitored by the 37.5 sensors and It is not possible physically and technically. For getting X = 0.40 we have X = 37.5/15 = 2.5 or X = 15/37.5 = 0.40, where 37.5 is the active number of sensors in the 15 regions. Briefly, we cannot achieve 37.5 active sensors for the design we have chosen for our simulations. Therefore, to tackle this problem we have used the round off code (explained in Section-3.6.2) to round off the 37.5 as 37 and just saving nearly 1 more sensor to get active.

Simulation Scenario 2

In order to achieve the QoS = 0.50 (from a Grid having 16 regions) we have should have 16 regions and these 16 regions must be monitored by the 32 sensors as we know that 16/32 = 0.50 and if we take this as a vice versa we will have 32/16 = 2 i.e. each region is monitored by the 2 sensors. Further, in order to achieve the QoS = 0.50 (from a Grid having 15 regions) we have should have 15 regions and these 15 regions must be monitored by the 30 sensors as we know that 15/30 = 0.50 and if we take this as a vice versa we will have 30/15 = 2 i.e. each region is monitored by the 2 sensors.

Testbed, Experimental Setup and Simulation Outputs

We have used Java Technology (i.e. JDK 1.6) for simulations and MATLAB for drawing the Graphs and these two software are running on top of the IBM System x, running with Novell's SUSE Linux Enterprise Server 11. We have used the java.util.Random class allows you to create objects that produce pseudo-random numbers with uniform or gaussian distributions according to a linear congruential formula with a 48-bit seed. To handle the Redundancy we have we have used HashSet method to create a HashSet, which allows no duplicates but in order to ensure correct order, since HashSet does not we have to use Collections method.

Application of Gur Game Based Algorithm on WSNs for 4 x 4 Homogenous Grid to achieve QoS = 0.40 and QoS = 0.50

Algorithm: GeneralizedGurGame_Sensors_HomogenousGrid

Step1: Enter the number of Sensor nodes[1....N](here $N = 2^n$, where n = 4,6,8,10....so on) which are deployed in Sensor field.

Step2: Enter the number of Regions the field is divided into (M Regions [here M regions are either 16n (where n = 1) or multiples of 16n (where n = 4) onwards] in [P x P Grid, here $P = 2^n x 2^n$, where n = 2,3,4,5....so on]). According to the Algorithm, the size of the Grid can be 4 x 4, 8 x 8, 16 x 16, 32 x 32, 64 x 64 so on. In a specific Grid the regions are numbered from 0 to M-1 i.e. a 4 x 4 Grid will have 16 Regions {numbered from 0 to 15}in which 16 Sensors can be deployed similarly in a 8 x 8 Grid will have 64 Regions {numbered from 0 to 63} in which 64 Sensors can be deployed and so on.

Step3: If Sensor no. < total Sensors goto Step4 **Else** goto Step10.

Step4: Enter **If** Sensor is active or not.

Step5: If Sensor is active goto Step6 **Else** goto Step9.

Step6: If the size of the Grid is 4 x 4 and Sensor index is {1,4,13,16} then total Regions covered is 3 **Else** goto Step7. Similarly, **If** the size of the Grid is 8 x 8 and Sensor index is {1,8,57,64} then total Regions covered is 3 **Else** goto Step7, **If** the size of the Grid is 16 x 16 and Sensor index is {1,16,241,256} then total Regions covered is 3 **Else** goto Step7, **If** the size of the Grid is 32 x 32 and Sensor index is {1,32,993,1024} then total Regions covered is 3 **Else** goto Step7 and so on.

Step7: If the size of the Grid is 4 x 4 and Sensor index is {2,3,5,8,9,12,14,15} then total Regions covered is 5 **Else** total Regions covered is 8. Similarly, **If** the size of the Grid is 8 x 8 and Sensor Index is {2, 3, 4, 5, 6, 7, 9, 16, 17, 24, 25, 32, 33, 40, 41, 48, 49, 56, 58,59,60,61,62,63} then total Regions covered is 5 **Else** total Regions covered is 8, **If** the Size of the Grid is 16 x 16 and the Sensor index is {2....15, 17, 32, 33, 48, 49, 64, 65, 80, 81, 96, 97,112, 113, 128, 129, 144, 145, 160, 161, 176, 177, 192, 193, 208, 209, 224, 225,240,242....255} and so on.

Step8: Regions covered by the Sensor are designated to the active Sensor according to the adjacent Region for every node.

Step9: Increment Sensor no. and goto Step3.

Step10: Count the number of active Sensors.

Step11: Calculate total Regions covered by active Sensors (with repetition).

Step12: Remove duplicates from Step11.

Step13: Calculate the value of X which is the ratio between Regions covered (result of Step12) and active Sensor nodes.

Step14: Calculate v where v=(-0.002)*((X*100-40)*(X*100-40)) [15, 16].

Or

Calculate v where v=(-0.002)*((X*100-50)*(X*100-50)) [15, 16].

Step15: Calculate the Reward function r in this simulation where r=0.2+0.8*pow(e,v) [15, 16].

Step16: If Sensor was active, assign the value of Reward r **Else** Penalized with Probability 1-r.

step17: Compare X with 0.40, **If** X=0.40, QoS is achieved, which implies that each Region is covered by more than 2.5 active Sensor nodes on an average **Else** QoS is not achieved.

Or

Compare X with 0.50, **If** X=0.50, QoS is achieved, which implies that each Region is covered by more than 2.0 active Sensor nodes on an average **Else** QoS is not achieved.

Simulation Results

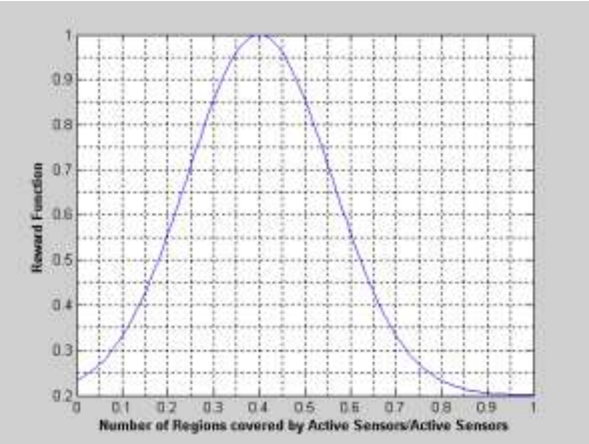


Fig. 1. The MATLAB figure shows the graph for Number of Regions Covered by Active Sensors/Active Sensors on the X-axis drawn against the Reward Function on the Y-axis. The Maximum Reward can be 1 only as the Probability moves from 0 to 1 only. The Reward Function has highest value i.e. 1 when the QoS = 0.40 i.e. there are 2.5 Active Sensors to monitor a Region.

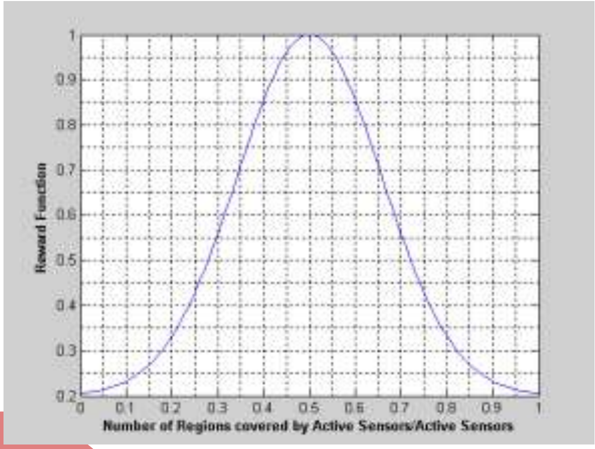
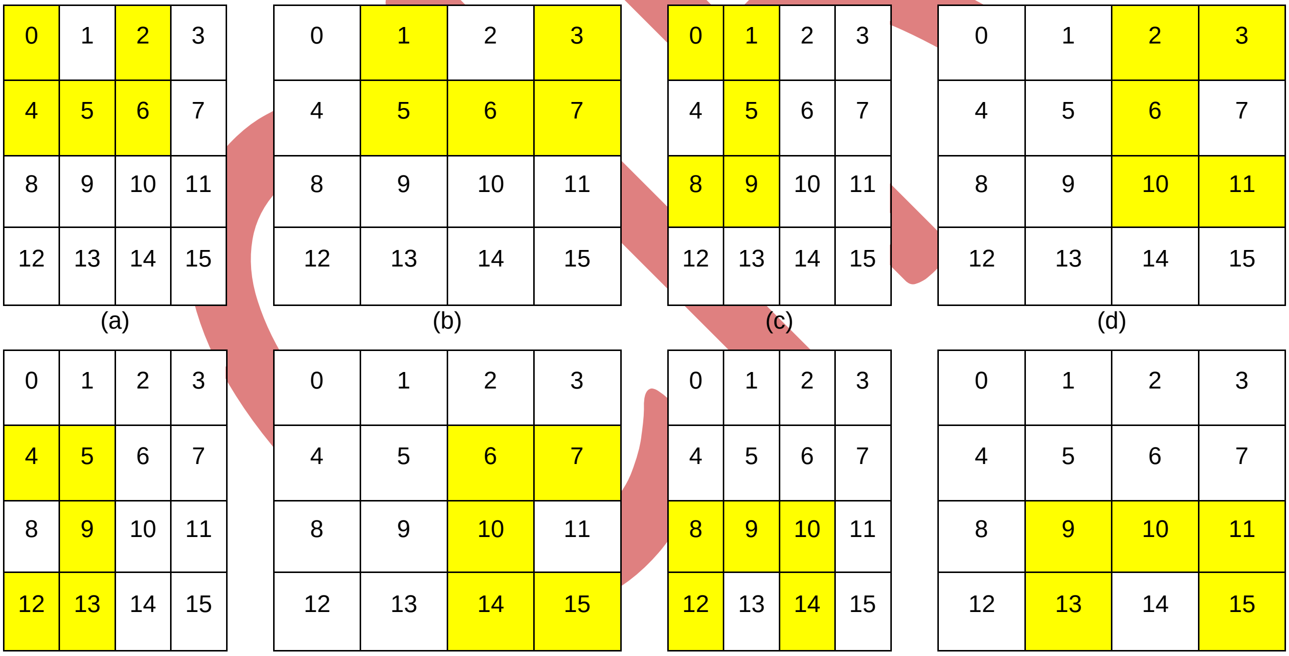


Fig. 2. The MATLAB figure shows the graph for Number of Regions Covered by Active Sensors/Active Sensors on the X-axis drawn against the Reward Function on the Y-axis. The Maximum Reward can be 1 only as the Probability moves from 0 to 1 only. The Reward Function has highest value i.e. 1 when the QoS = 0.50 i.e. there are 2 Active Sensors to monitor a Region.



0	1	**2**	3
4	**5**	**6**	7
8	9	10	11
12	13	14	15

(a)

0	**1**	2	**3**
4	**5**	**6**	**7**
8	9	10	11
12	13	14	15

(b)

0	**1**	2	3
4	**5**	6	7
8	**9**	10	11
12	13	14	15

(c)

0	1	**2**	**3**
4	5	**6**	7
8	9	**10**	**11**
12	13	14	15

(d)

0	1	2	3
4	**5**	6	7
8	**9**	10	11
12	**13**	14	15

(e)

0	1	2	3
4	5	**6**	**7**
8	9	**10**	11
12	13	**14**	**15**

(f)

0	1	2	3
4	5	6	7
8	**9**	**10**	11
12	13	**14**	15

(g)

0	1	2	3
4	5	6	7
8	**9**	**10**	**11**
12	**13**	14	**15**

(h)

Fig. 3. The figure shows the 4 x 4 Homogenous Grid. Only few sensors are activated here to achieve the QoS = 0.40. From here, we can see that some of the sensors are activated to monitor 4 x 4 Grid in order to achieve the QoS = 0.40. From (a) it can be seen that the Sensor 2 is monitoring the Region 0, 2, 4, 5 and 6. From (b) it can be seen that the Sensor 3 is monitoring the Regions 1, 3, 5, 6 and 7. From (c) it can be seen that the Sensor 5 is monitoring the Region 0, 1, 5, 8 and 9. From (d) it can be seen that the Sensor 8 is monitoring the Region 2, 3, 6, 10 and 11. From (e) it can be seen that the Sensor 9 is monitoring the Region 4, 5, 9, 12 and 13. From (f) it can be seen that the Sensor 12 is monitoring the Region 6, 7, 10, 14 and 15. From (g) it can be seen that the Sensor 14 is monitoring the Region 8, 9, 10, 12 and 14 and from (h) it can be seen that the Sensor 15 is monitoring the Region 9, 10, 11, 13 and 15.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(a)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(b)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(c)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(d)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(e)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(f)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(g)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(h)

Fig. 4. The figure shows the 4 x 4 Homogenous Grid. Only few sensors are activated here to achieve the QoS = 0.50. From here, we can see that some of the sensors are activated to monitor 4 x 4 Grid in order to achieve the QoS = 0.50. From (a) it can be seen that the Sensor 1 is monitoring the Region 1, 4 and 5. From (b) it can be seen that the Sensor 2 is monitoring the region 0, 2, 4, 5 and 6. From (c) it can be seen that the Sensor 3 is monitoring the Region 1, 3, 5, 6 and 7. From (d) it can be seen that the Sensor 4 is monitoring the Region 2, 6 and 7. From (e) it can be seen that the Sensor 13 is monitoring the Region 8, 9 and 13. From (f) it can be seen that the Sensor 14 is monitoring the Region 8, 9, 10, 12 and 14. From (g) it can be seen that the Sensor 15 is monitoring the Region 9, 10, 11, 13 and 15 and from (h) it can be seen that the Sensor 16 is monitoring the Region 10, 11 and 14.

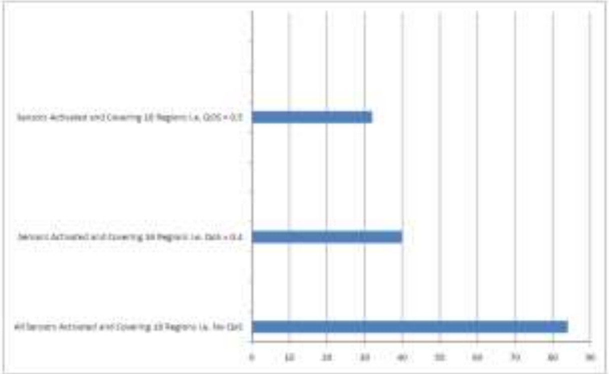


Fig. 5. The figure shows the Total Number of the Sensors Actiavated to Monitor the 4 x 4 Grid for NoQoS i.e in the Worst Case is 84 (X = 16/84 = 0.19) Sensors, the Total Number of Sensors Activated to Monitor the 4 x 4 Grid to achive the QoS = 0.40 (X = 16/40 = 0.40) i.e. in the Better Case is 40 Sensors and the Total Number of Sensors Activated to Monitor the 4 x 4 Grid to achive the QoS = 0.50 (X = 16/32 = 0.2) i.e. in the Best Case is 32.

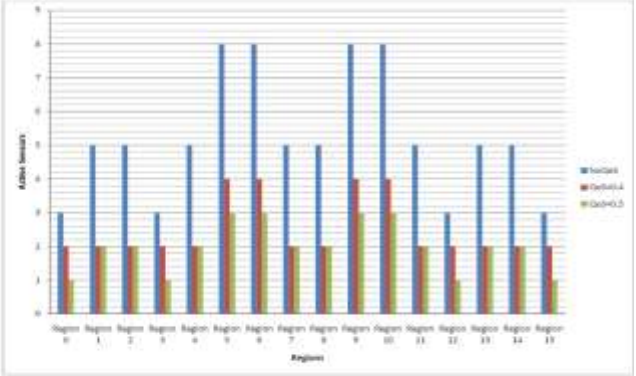


Fig. 6. The figure shows the Number of the Sensors Active in different Regions to Monitor the 4 x 4 Homogenous Grid for NoQoS, the Number of the Sensors Active in different Regions to Monitor the 4 x 4 Homogenous Grid for QoS = 0.40 and the Number of the Sensors Active in different Regions to Monitor the 4 x 4 Homogenous Grid for QoS = 0.50.

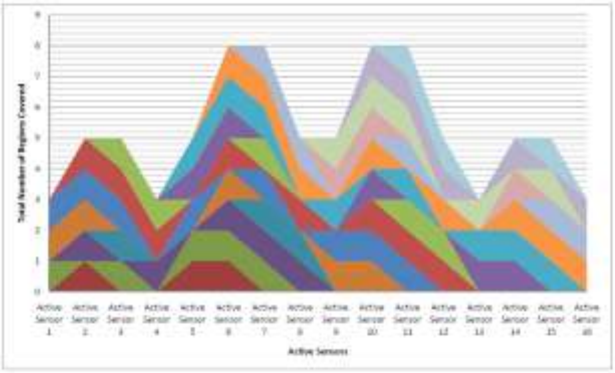


Fig. 7. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Homogenous Grid (with 16 Regions) for NoQoS.

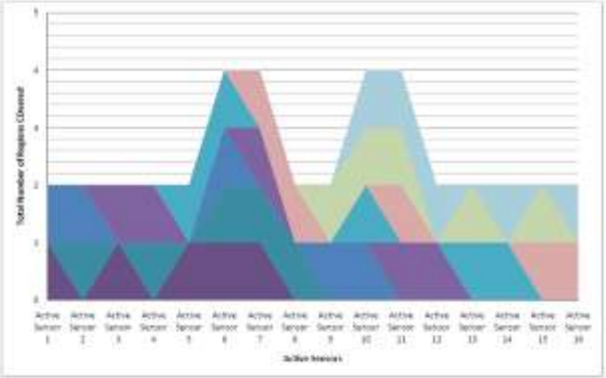


Fig. 8. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Homogenous Grid (with 16 Regions) for QoS = 0.40. The Sensors that are Active are #2, #3, #5, #8, #9, #12, #14, and #15.

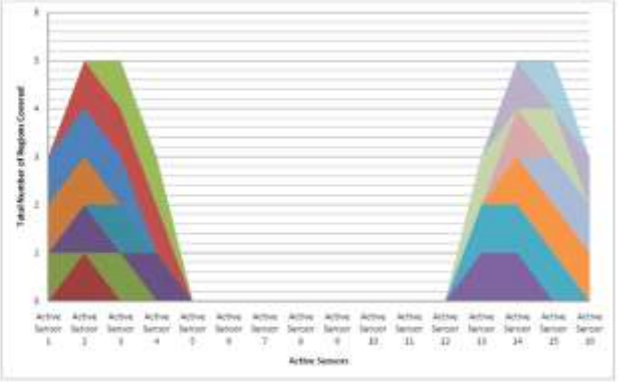


Fig. 9. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Homogenous Grid (with 16 Regions) for QoS = 0.50. The Sensors that are Active are #1, #2, #3, #4, #13, #14, #15 and #16.

Application of Gur Game Based Algorithm on WSNs for 4 x 4 Heterogeneous Grid to achieve QoS = 0.40 and QoS = 0.50

Algorithm: GurGame_Sensors_HeterogenousGrid

```
Step1: Enter the number of Sensor nodes which are deployed in Sensor field.
Step2: Enter the number of Regions[0...14] the field is divided into.
Step3: If Sensor no. < total Sensors goto Step4 Else goto Step12.
Step4: Enter If Sensor is active or not.
Step5: If Sensor is active goto Step6 Else goto Step11.
Step6: If Sensor index is {3,12,15} then total Regions covered is 3 Else goto Step7.
Step7: If Sensor index is {1,4} then total Regions covered is 4 Else goto Step8.
Step8: If Sensor index is {2,7,8,11,13,14} then total Regions covered is 5 Else goto Step9.
Step9: If Sensor index is {5} then total Regions covered is 7 Else if Sensor index is {6,9,10} total Regions covered is 8.
Step10: Regions covered by the Sensor are designated to the active Sensor according to the adjacent Region for every node.
Step11: Increment Sensor no. and goto Step3.
Step12: Count the number of active Sensors.
Step13: Calculate total Regions covered by active Sensors (with repetition).
```

Step14: Remove duplicates from Step13.

Step15: Calculate the value of X which is the ratio between Regions covered (result of Step14) and active Sensor nodes.

Step16: Calculate v where v=(-0.002)*((X*100-40)*(X*100-40)) [15, 16]

Or

Calculate v where v=(-0.002)*((X*100-50)*(X*100-50)) [15, 16]

Step17: Calculate the Reward function r in this simulation where r=0.2+0.8*pow(e,v) [15, 16].

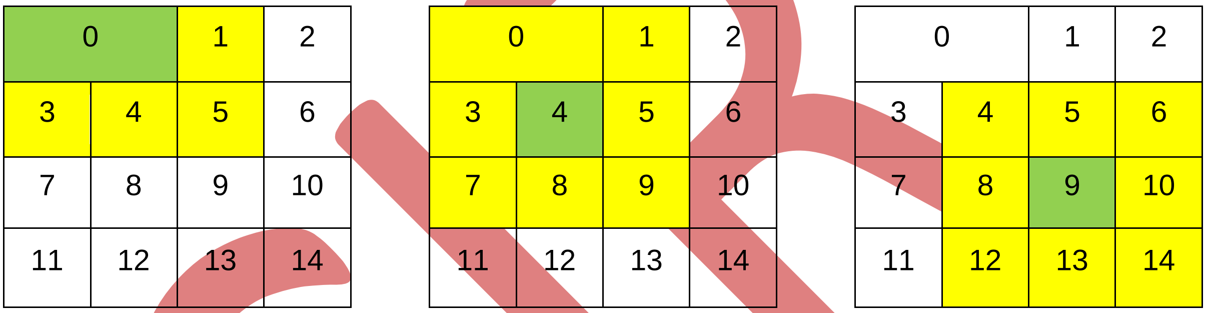
Step18: If Sensor was active, assign the value of Reward r **Else** Penalized with Probability 1-r.

Step19: Compare X with 0.40, **If** X=0.40, QoS is achieved, which implies that each Region is covered by more than 2.5 active Sensor nodes on an average **Else** QoS is not achieved.

Or

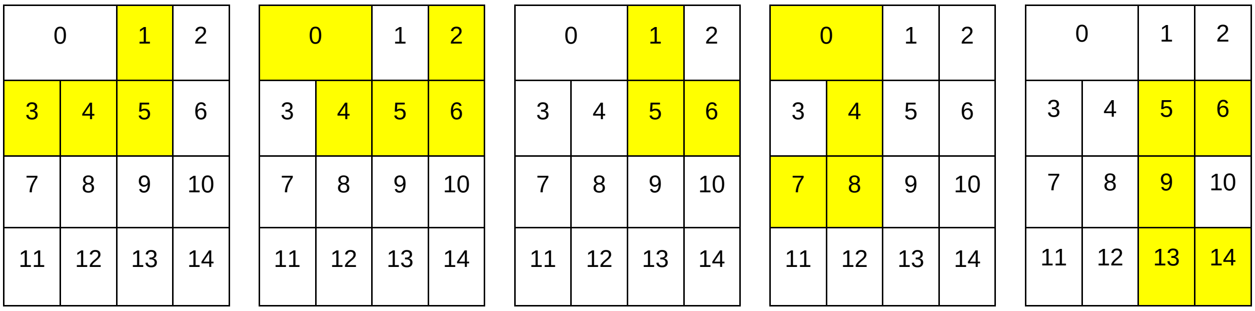
Compare X with 0.50, **If** X=0.50, QoS is achieved, which implies that each Region is covered by more than 2.0 active Sensor nodes on an average **Else** QoS is not achieved.

Simulation Results



(a) (b) (c)

Fig. 10. From here, we can see that all the sensors are active and monitoring the entire 4 x 4 Grid. We have initially activated all the 16 Sensors and it will have the high redundancy. Similarly, Sensor 1 that is placed in the Region 0 is covering or monitoring Region 1, 4, 5 and 3, Sensor 2 that is placed in the Region 1 is covering or monitoring Region 0, 2, 4, 5 and 6, Sensor 3 that is placed in the Region 2 is covering or monitoring Region 1, 5 and 6 and Sensor 4 that is placed in the Region 3 is covering or monitoring Region 4, 7, 8, and 0. From (b) it can be seen that the Sensor 5 that is placed in the Region 4 is covering or monitoring Region 0, 1, 3, 5, 7, 8 and 9. Similarly, Sensor 6 that is placed in the Region 5 is covering or monitoring Region 0, 1, 2, 4, 6, 8, 9 and 10, Sensor 7 that is placed in the Region 6 is covering or monitoring Region 1, 2, 5, 9 and 10, Sensor 8 that is placed in the Region 7 is covering or monitoring Region 12, 11, 8, 4 and 3 and Sensor 9 that is placed in the Region 8 is covering or monitoring Region 3, 4, 5, 7, 9, 11, 12 and 13. From (c) it can be seen that the Sensor 10 that is placed in the Region 9 is covering or monitoring Region 4, 5, 6, 8, 10, 12, 13 and 14, Sensor 11 that is placed in the Region 10 is covering or monitoring Region 5, 6, 9, 13 and 14, Sensor 12 that is placed in the Region 11 is covering or monitoring Region 7, 8 and 12, Sensor 13 that is placed in the Region 12 is covering or monitoring Region 7, 8, 9, 11 and 13, Sensor 14 that is placed in the Region 13 is covering or monitoring Region 8, 9, 10, 12 and 14 and Sensor 15 that is placed in the Region 14 is covering or monitoring Region 9, 10 and 13.



(a) (b) (c) (d) (e)

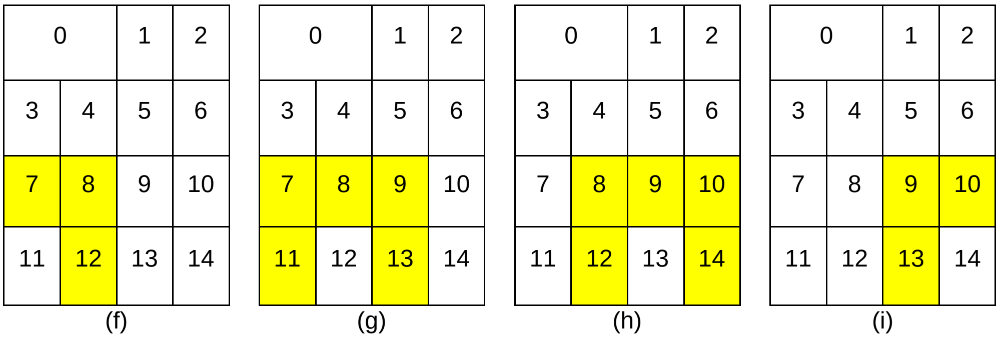


Fig. 11. The figure shows the 4 x 4 Heterogenous Grid. Only few sensors are activated here to achieve the QoS = 0.40. From here, we can see that some of the sensors are activated to monitor 4 x 4 Grid in order to achieve the QoS = 0.40. From (a) it can be seen that the Sensor 1 that is placed in the Region 0 is covering or monitoring Region 1, 4, 5 and 3. From (b) it can be seen that the Sensor 2 that is placed in the Region 1 is covering or monitoring Region 0, 2, 4, 5 and 6. From (c) it can be seen that the Sensor 3 that is placed in the Region 2 is covering or monitoring Region 1, 5 and 6. From (d) it can be seen that the Sensor 4 that is placed in the Region 3 is covering or monitoring Region 4, 7, 8, and 0. From (e) it can be seen that the Sensor 11 that is placed in the Region 10 is covering or monitoring Region 5, 6, 9, 13 and 14. From (f) it can be seen that the Sensor 12 that is placed in the Region 11 is covering or monitoring Region 7, 8 and 12. From (g) it can be seen that the Sensor 13 that is placed in the Region 12 is covering or monitoring Region 7, 8, 9, 11 and 13 and from (h) it can be seen that the Sensor 14 that is placed in the Region 13 is covering or monitoring Region 8, 9, 10, 12 and From (i) it can be seen that the Sensor 15 that is placed in the Region 14 is covering or monitoring Region 9, 10 and 13.

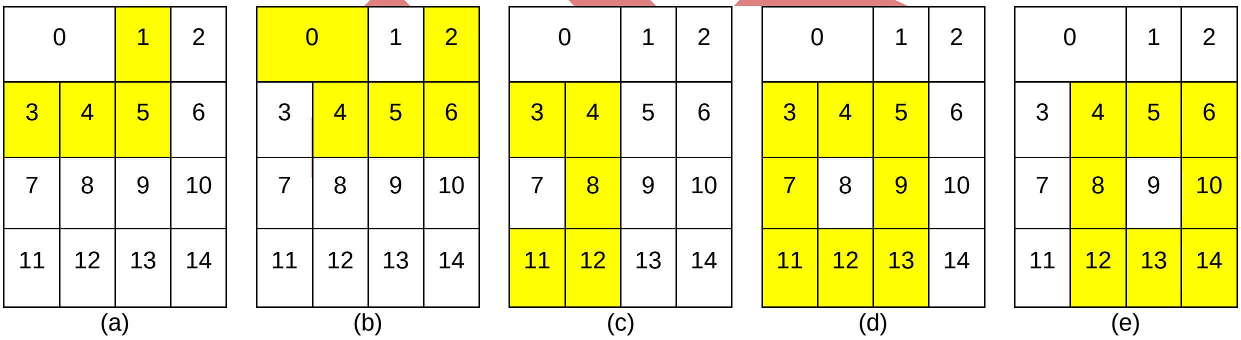


Fig. 12. The figure shows the 4 x 4 Heterogenous Grid. Only few sensors are activated here to achieve the QoS = 0.50. From here, we can see that all the sensors are active and monitoring the entire 4 x 4 Grid. We have initially activated all the 16 Sensors and it will have the high redundancy. From (a) it can be seen that the Sensor 1 that is placed in the Region 0 is covering or monitoring Region 1, 4, 5 and 3. From (b) it can be seen that the Sensor 2 that is placed in the Region 1 is covering or monitoring Region 0, 2, 4, 5 and 6. From (c) it can be seen that the Sensor 8 that is placed in the Region 7 is covering or monitoring Region 12, 11, 8, 4 and 3. From (d) it can be seen that the Sensor 9 that is placed in the Region 8 is covering or monitoring Region 3, 4, 5, 7, 9, 11, 12 and 13. From (e) it can be seen that the Sensor 10 that is placed in the Region 9 is covering or monitoring Region 4, 5, 6, 8, 10, 12, 13 and 14.

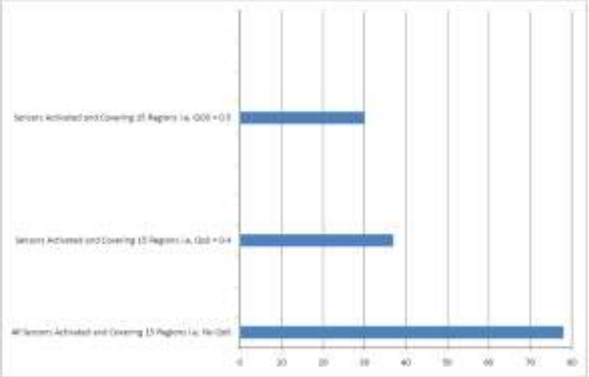


Fig. 13. The figure shows the total number of the Sensors Actiavated to Monitor the 4 x 4 Grid (with 15 Regions) for NoQoS i.e in the Worst Case is 78 (X = 15/78 = 0.19) Sensors, the total number of Sensors Activated to Monitor the 4 x 4 Grid (with 15 Regions)

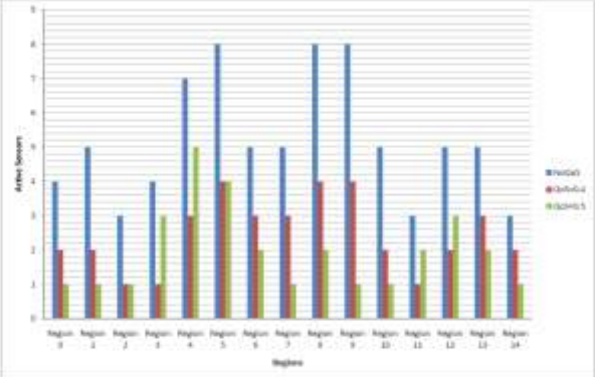


Fig. 14. The figure shows the Number of the Sensors Active in different Regions to Monitor the 4 x 4 Heterogenous Grid for NoQoS, the Number of the Sensors Active in different Regions to Monitor the 4 x 4 Heterogenous Grid for QoS = 0.40 and the

to achive the QoS = 0.40 (X = 15/37 = 0.405 ≈ 0.40) i.e. in the Better Case is 37 Sensors and the total number of Sensors Activated to Monitor the 4 x 4 Grid (with 15 Regions) to achive the QoS = 0.50 (X = 15/30 = 0.2) i.e. in the Best Case is 30.

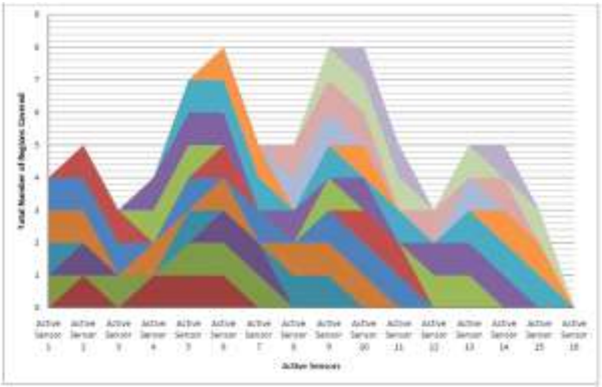


Fig. 15. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Heterogenous Grid (with 15 Regions) for NoQoS. All Sensors are Active at this time.

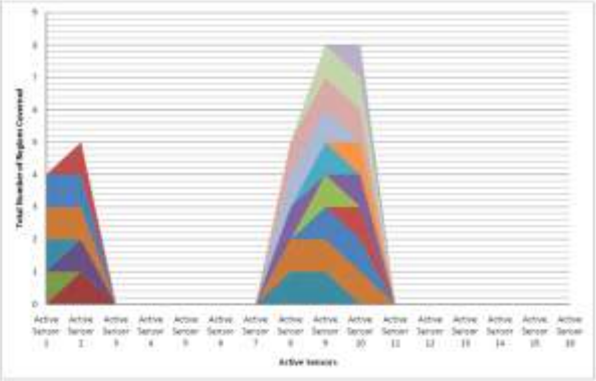


Fig. 17. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Heterogenous Grid (with 15 Regions) for QoS = 0.50. The Sensors that are Active are #1, #2, #8, #9, and #10.

Number of the Sensors Active in different Regions to Monitor the 4 x 4 Heterogenous Grid for QoS = 0.50.

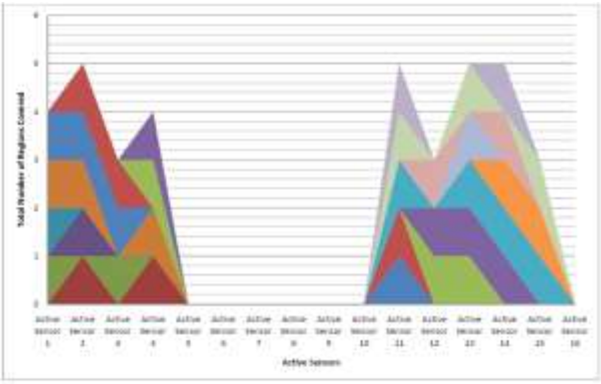


Fig. 16. The figure shows the Total Number of Regions Covered by the Active Sensors to Monitor the 4 x 4 Heterogenous Grid (with 15 Regions) for QoS = 0.40. The Sensors that are Active are #1, #2, #3, #4, #11, #12, #13, #14 and #15.

Result Analysis

Homogeneous Grid

The simulations results achieved by the application of Gur Game based Algorithms on 4 x 4 Homogenous Grid presented in figure (3-9) shows that the number of sensors to be activated in order to achieve the QoS = 0.40 is same as the number of sensors needed to activate and to achieve the QoS of 0.50. Therefore, we are achieving high QoS = 0.50 in comparison to QoS = 0.40 at least expense of energy dissipation and number of sensors.

Heterogeneous Grid and its Limitations

In order to achieve the QoS of 0.40 we have to activate the sensors and control will be in the user's hands as it is Application–Specific. The number of sensors to be activated to monitor the Homogenous Grid is 9 and the number of sensors to be activated to monitor the Heterogeneous Grid is 5 only so we are saving lots of sensors and achieving high QoS.

Explanation for Heterogeneous WSN Design

1. The programs with the outputs i.e. for the QoS of 0.40,
2. The programs with the outputs i.e. for the QoS of 0.50,
3. The programs with the worst case outputs i.e. for the NoQoS (The total number of sensor active during Homogeneous was 84 and during the Heterogeneous it comes out to be 78 only-this is just because of the design).

We know that in case of Homogeneous Design

- For getting X = 0.50 we should have X = 32/16 = 2 or X = 16/32 = 0.50, where 32 is the active number of sensors in the 16 regions and hence we are achieving the QoS.
- For getting X = 0.40 we should have X = 40/16 = 2.5 or X = 16/40 = 0.40, where 40 is the active number of sensors in the 16 regions and hence we are achieving the QoS.

Now for Heterogeneous Design

- For getting X = 0.50 the story is simple as X = 30/15 = 2 or X = 15/30 = 0.50, where 30 is the active number of sensors in the 15 regions and hence we are achieving the QoS.
  Further, one of the biggest problems here is that according to the Heterogeneous Design, we have 15 regions and 16 sensors is:
- For getting X = 0.40 the story is difficult as X = 37.5/15 = 2.5 or X = 15/37.5 = 0.40, where 37.5 is the active number of sensors in the 15 regions and this is not possible during the coding and for the design we have chosen. Briefly, we cannot achieve 37.5 active sensors. Therefore, to tackle this problem we have used the following code to roundoff the 37.5 as 37 and just saving nearly one more sensor to get active.

```
DecimalFormat format = new DecimalFormat("#.##");
format.setRoundingMode(RoundingMode.FLOOR);
  String su = format.format(X);
   X1 = Float.parseFloat(su);
   if (Float.compare(X1,(float) 0.40)==0)
      X=X1;
   else
      X=X;
System.out.println("x="+X);
```

Therefore, after this code appears we have 37.5/15 ~ 37/15 = 0.40 and hence achieving the QoS. We have to use code to make our assumptions to run otherwise we cannot do anything for achieving the X = 0.40.

In this paper, we introduce self optimizing algorithms to regulate the process of activating sensors while maximizing the number regions covered by sensor nodes. Specifically, we have used Gur Game based Algorithms and its Application on WSNs for 4 x 4 Homogenous and Heterogeneous Grid to achieve the QoS = 0.40 and 0.50. d.

The simulations results achieved by the application of Gur Game based Algorithms on 4 x 4 Heterogeneous Grid presented in figure (10-17) shows that the number of sensors to be activated in order to achieve the QoS = 0.40 which is very high in comparison to the number of sensors needed to activate and to achieve the QoS = 0.50. Therefore, we are achieving high QoS = 0.50 in comparison to QoS = 0.40 at very least expense of energy dissipation and number of sensors. The design presented is here is only for 4 x 4 Heterogeneous architecture i.e. Application-Specific or in other words we have only explored the Application of Gur Game based Algorithm on WSNs monitoring 4 x 4 Heterogeneous Grid to achieve QoS = 0.40 and 0.50. This approach is Application-Specific and cannot be generalized, as the degree of Heterogeneity, is extremely high.

CONCLUSION

In this paper, we introduce self optimizing algorithms to regulate the process of activating sensors while maximizing the number regions covered by sensor nodes. Specifically, we have used Gur Game based Algorithms and its Application on WSNs for 4 x 4 Homogenous and Heterogeneous Grid to achieve the QoS = 0.40 and 0.50.

The simulations results achieved by the application of Gur Game based Algorithms on 4 x 4 Homogenous Grid presented in figure (3-9) shows that the number of sensors to be activated in order to achieve the QoS = 0.40 is same as the number of sensors needed to activate and to achieve the QoS of 0.50. Therefore, we are achieving high QoS = 0.50 in comparison to QoS = 0.40 at least expense of energy dissipation and number of sensors.

The simulations results achieved by the application of Gur Game based Algorithms on 4 x 4 Heterogeneous Grid presented in figure (10-17) shows that the number of sensors to be activated in order to achieve the QoS = 0.40 is very high in comparison to the number of sensors needed to activate and to achieve the QoS of 0.50. Therefore, we are achieving high QoS = 0.50 in comparison to QoS = 0.40 at very least expense of energy dissipation and number of sensors.

The objectives of all these algorithms are to maximize the coverage of the sensor area while conserving energy consumed by the sensor nodes. This is achieved via carefully activating/deactivating the sensors while maximizing the coverage area. The simulations programs have been designed in such a way that it makes the activation and deactivation of the sensor to be controlled/handled either by the user or by the system itself. In order to achieve the much desired redundancy in covering the monitored regions, the algorithm attempts to find the best assignment of active sensors while maximizing the number of regions covered by sensor nodes.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless Sensor Networks: A Survey, Computer Networks, 38(4), pp. 393-422, 2002.

[2] R. Kay and F. Mattern, The Design Space of Wireless Sensor Networks, IEEE Wireless Communications 11(6), pp. 54-61, 2004.

[3] A. Tiwari, Energy-efficient Wireless Sensor Network Design and Implementation for Condition-based Maintenance, ACM Transactions on Sensor Networks 3(1), pp. 1-23, 2007.

[4] S. Hadim and M. Nader, Middleware Challenges and Approaches for Wireless Sensor Networks, IEEE Distributed Systems Online **7**(3), pp. 1-23, 2006.

[5] http://en.wikipedia.org/wiki/Wireless_sensor_network

[6] R. Iyer and L. Kleinrock, QoS Control For Sensor Networks, In Proceedings of the IEEE International Conference on Communications, pp. 517-521, 2003.

[7] B. Tung and L. Kleinrock, Using Finite State Automata to Produce Self-Optimization and Self-Control, IEEE Transactions on Parallel and Distributed Systems 7(4), pp. 439-448, 1996.

[8] C. Intanagonwiwat, R. Govindan and D. Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, In Proceedings of the MOBICOM, pp. 56-67, 2000.

[9] Q. Li, J. Aslam, and D. Rus, Online Power-aware Routing in Wireless Ad-hoc Networks, In Proceedings of the MOBICOM, pp. 97-107, 2001.

[10] S. Dulman, T. Nieberg, P. Havinga and P. Hartel, Multipath Routing for Data Dissemination in Energy Efficient Sensor Networks, In Proceedings of CTIT-02-20, pp. 6-163, 2002.

[11] T. He, J. Stankovic, C. Lu and T. Abdelzaher, SPEED: A Real-Time Routing Protocol for Sensor Networks, In Proceedings of the IEEE Distributed Computing Systems, pp. 46-55, 2003.

[12] B. Krishnamachari, Y. Mourtada and S. Wicker, The Energy-Robustness Tradeoff for Routing in Wireless Sensor Networks, In Proceedings of the IEEE ICC, pp. 1833-1837, 2003.

[13] A. Cerpa and D. Estrin, Ascent: Adaptive Self-configuring Sensor Networks Topologies, IEEE Transactions on Mobile Computing, 3(3), pp. 1-14, 2004.

[14] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastava, Topology Management for Sensor Networks: Exploiting Latency and Density, In Proceedings of the 3rd ACM International Symposium on Mobile Adhoc Networking & Computing, pp. 135-145, 2002.

[15] S.I. Nayer and H. Ali, A Dynamic Energy-Aware Algorithm for Self-Optimizing Wireless Sensor Networks, Self Organizing Systems, Lecture Notes in Computer Science 5343, pp. 262-268, 2008.

[16] S.I. Nayer and H. Ali, On Employing Distributed Algorithms and Evolutionary Algorithms in Managing Wireless Sensor Networks, Proceedings of the International Workshop on Theoretical and Algorithmic Aspects of Wireless Ad Hoc, Sensor, and P2P Networks, Chicago, Illinois, 2004.

Author' biography with Photo

Dr. Nitin is Ex First Tier Bank Professor, University of Nebraska at Omaha, NE, USA. His permanent affiliation is with Jaypee University of Information Technology (JUIT), Waknaghat, Solan-173234, Himachal Pradesh, INDIA as a Associate Professor in the Department of Computer Science & Engineering and Information & Communication Technology. He was born on October 06, 1978, in New Delhi, INDIA.

In July 2001, he received the B.Engg. in Computer Science & Engineering [Hons.] and M.Engg. in Software Engineering from Thapar Institute of Engineering and Technology, Patiala, Punjab, INDIA in March 2003. In 2008, he received his Ph.D. in Computer Science & Engineering from JUIT, INDIA. He has completed his Ph.D. course work from University of Florida, Gainesville, FL, USA.

He is a IBM certified engineer and Senior Member-IEEE & IACSIT, Life Member of IAENG, and Member-SIAM & ACIS. He has 125 research papers in peer reviewed International Journals & Transactions, Book Chapters, Symposium, Conferences and Position. His research interest includes Social Networks especially Computer Mediated Communications & Flaming, Interconnection Networks & Architecture, Fault-tolerance & Reliability, Networks-on-Chip, Systems-on-Chip, and Networks-in-Packages, Application of Stable Matching Problems, Stochastic Communication and Sensor Networks. Currently he is working on Parallel Simulation tools, BigSim using Charm++, NS-2 using TCL. He is the Co-founder of High-end Parallel Computing and Advanced Computer Architecture Lab at JUIT. He is Associate Editor of Journal of Parallel, Emergent and Distributed Systems, Taylor and Francis, UK. He is referee for the Journal of Parallel and Distributed Computing, Elsevier Sciences, Computer Communications, Elsevier Sciences, Computers and Electrical Engineering, Elsevier Sciences, Mathematical and Computer Modelling, Elsevier Sciences. WSEAS Transactions, The Journal of Supercomputing, Springer and International Journal of System Science, Taylor & Francis.