International Journal of Grid and Distributed Computing
Vol.7, No.1 (2014), pp.53-66
http://dx.doi.org/10.14257/ijgdc.2014.7.1.06

A novel intelligent Sleep Wakeup Scheduling algorithm to the Area Coverage problem in Wireless Sensor Networks

Hagar Noori [1] and Reza Rafeh[2]

[1,2]*Department of computer engineering, science and research branch, Islamic Azad University, Arak-Iran,*
[1]*hagar_noori@yahoo.com*, [2]*reza-raafeh@yahoo.com*

Abstract

One major problem in the area of wireless sensor networks is the coverage problem. The coverage problem deals with the ability of the network to cover a certain area or some certain events. In this paper, we focus on the area coverage problem. We use a cluster-based coverage control scheme and propose HSSAC protocol to maintain sensing coverage by keeping a small number of active sensor nodes and a small amount of energy consumption in a wireless sensor network. In this protocol, proper active sensor set can be determined using the harmony search algorithm. Due to the proposed protocol accuracy in selecting the active sensor set, it is able to provide the acceptable coverage rate in sparse deployment. As the result of increasing nodes density, the proposed protocol decreases the number of active nodes in the sensor networks. Thus, the proposed protocol decreases the energy consumption of the networks and prolongs the network lifetime. We have simulated our protocol and simulation results show high efficiency of the proposed protocol.

Keywords: *Area Coverage, Energy Consumption, Harmony Search algorithm, Scheduling, Wireless Sensor Networks*

1. Introduction

The Wireless Sensor Networks (WSNs) are designed to conduct surveillance tasks, such as monitoring an area, several known/unknown targets and so on [1]. The nodes in WSNs are usually powered by batteries with finite capacity and it is always impossible to replenish the power [2]. Therefore, the applications are hindered by limited energy supply, and one design challenge in sensor networks is to save limited energy resources to prolong the lifetime of the WSN [2, 3]. Power saving techniques can generally be classified in two categories [2]:

- Scheduling the sensor nodes
- Adjusting the transmission or sensing range

In this paper, we deal with the area coverage problem by keeping a small number of active sensor nodes and a small amount of energy consumption. We use a cluster-based coverage control scheme and propose HSSAC[1] protocol.

The remaining of the paper is organized as follows: Section 2 presents the related works. Section 3 illustrates the Harmony Search (HS) algorithm briefly. Section 4 presents the network model and Section 5 describes the proposed algorithm. Section 6 presents some simulation results and evaluates the proposed algorithm. The paper concludes with Section 7.

[1] HSSAC: Harmony Search Scheduling Area Coverage

ISSN: 2005-4262 IJGDC

2. Related Works

The Coverage problem deals with the ability of the network to cover a certain area or some certain events [4]. Various coverage formulations have been proposed in literature among which following three are most discussed [4]:

- Point coverage: the objective of point coverage problem is to cover a set of stationary or moving points [4].
- Barrier coverage: barrier coverage can be considered as the coverage with the goal of minimizing the probability of undetected penetration through the sensor network [4].
- Area coverage: monitoring the whole area of the network is the main objective of area coverage problem [4]. In this paper, we focus on the problem of Area Coverage.

By allowing redundant sensors to go into the sleep mode, the energy consumption network is reduced [2]. Thus, an important method for the Area Coverage problem is to find the maximal number of covers in sensor network, where the cover is a set of nodes that can completely cover the target area [2]. The problem of finding the maximal number of covers in a sensor network is addressed in [5]. It is an NP-complete problem [5]. In [6], a centralized solution is proposed to reduce the energy consumption network by turning on some redundant nodes in the sensor network. But, this solution requires a large number of nodes to operate in the active mode. Also, a sensor network must provide satisfactory the network connectivity [7]. But the work in [7] does not provide minimized number of active nodes. A straightforward solution is to use a transition range R_T that is at least twice the sensing range R_S (*i.e.*, $R_T \geq 2R_S$), such that Area Coverage guaranties the network connectivity of active nodes [8]. Yang *et al.*, in [9] addressed the *k*-Connected coverage set problems in wireless sensor network with the objective of minimizing the total energy consumption while achieving *k*-coverage for reliability. Note that a sensor network that achieves *k*-coverage could be *k*-connected. In [10], a set of nodes are made active to maintain coverage while others are put into sleeping modes to conserve energy.

We propose HSSAC protocol to maintain sensing coverage by keeping a small number of active sensor nodes and a small amount of energy consumption. In proposed protocol, proper active sensor set determine using the harmony search algorithm. Due to the proposed protocol accuracy in selecting the active sensor set, it is able to provide the full coverage in sparse deployed. Moreover, as the result of increasing nodes density, the proposed protocol decreases the energy consumption of the sensor networks and prolongs the network lifetime.

3. Harmony Search Algorithm

Harmony Search (HS) algorithm, originated by Geem *et al.*, [11], is based on natural musical performance processes. The HS algorithm consist five steps. The Block diagram of HS algorithm is shown in Figure 1 [12-14].

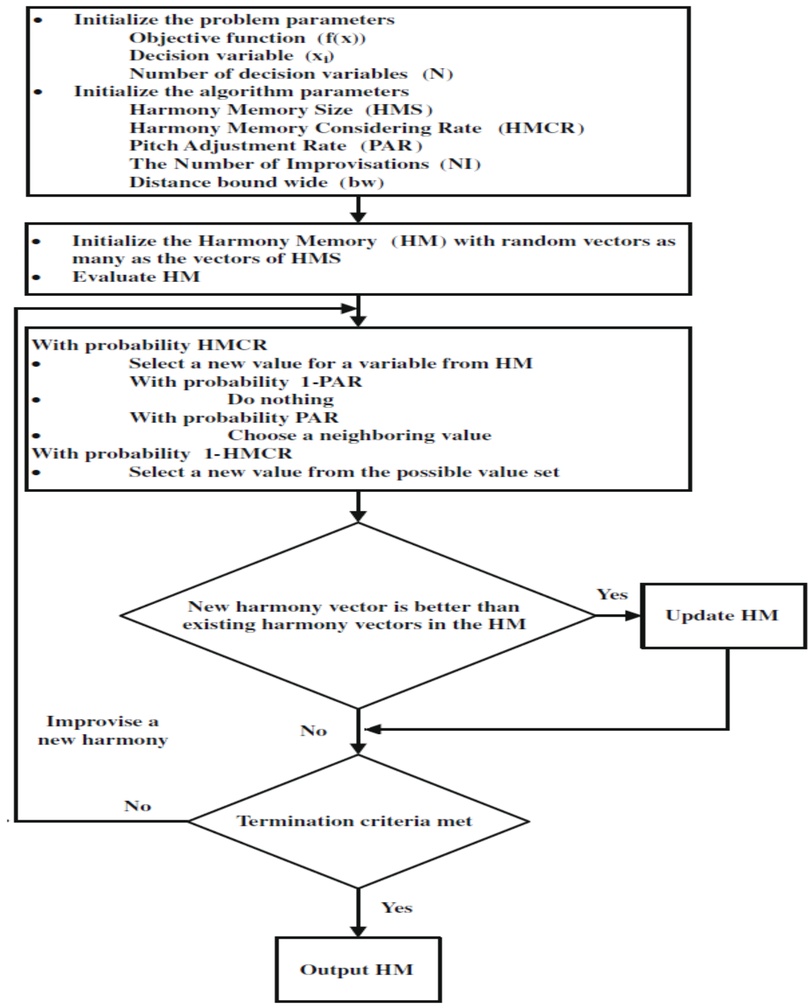


Figure 1. Block Diagram of HS Algorithm

4. The Network Model

In this section, we present the model and the assumptions used in this paper.

4.1. Coverage Problem Formulation

Define that the size of the target area is A_{total}, the size of the monitoring area is A_{area}, the sensor set is $S= \{n_1, n_2, \dots , n_N\}$ and the active sensors set is a subset of S.

In this paper, we will deal with the nodes deployed randomly. We assume that the nodes are static once deployed, and each one knows its own location which can be achieved by using some location system [15].

Since the relationship of coverage and connectivity has been proved in [8], the transmission range of sensor nodes is assumed to be at least twice the sensing range. Then coverage can imply connectivity [2]. Thus, we will only focus on the coverage problem (see Figure 2).

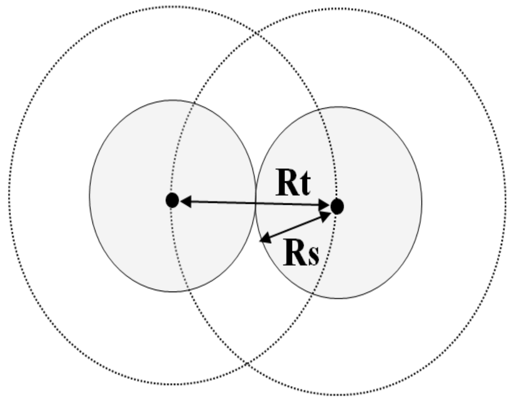


Figure 2. The Transmission Radius of Sensor Nodes is assumed to be at Least Twice the Sensing Radius. then Coverage can Imply Connectivity

The coverage model of the node n_i is supposed as a circle centered at its coordinates (x_i , y_i) with radius r_i [2]. Then, the probability of event e is covered by the sensor node ni is equal to a two-valued function $P_{cov}(e,n_i)$:

$$P_{cov}(e,n_i) = \begin{cases} 1 & \text{If } (x-x_i)^2 + (y-y_i)^2 \le r_i^2 \\ 0 & \text{else} \end{cases} \tag{1}$$

We assume that any random event e is independent of the others. Thus, it can be considered that the event e is covered by the sensor set if any sensor in the active senor set covers it.

Also, we define the coverage rate of the sensor set, $Coverage_{rate}$, as the proportion of the monitoring area A_{area} to the total area A_{total} [2]:

$$Covrage_{rate} = A_{area} / A_{total} \tag{2}$$

4.2. The Cluster-based Architecture

We use a cluster-based coverage control scheme in this paper, which is scheduled into rounds. In each round, firstly, the target area is divided into several equal squares. Then the node in each square having the largest energy will be chosen as the cluster-head. The procedure of selecting the cluster-head is the same works by Jia and Chena in 2006 and 2009 [2, 16]. The cluster-head will choose a set of active nodes to do the sensing job.

4.3. Energy Consumption Analysis

4.3.1. The Energy Consumption Analysis: we only consider the energy consumed by the sensing task, thus when the sensor node is in sleep mode, the consumed power is considered as zero [2].

4.3.2. The Energy Consumption Models: The energy consumed by a wireless sensor node for sensing function is proportional to R_S^2. R_S is the sensing range of the sensor node [2, 17].

Thus, the coverage energy consumption of the sensor set, which is related to the sum of the sensor's sensing radius squared, is defined as:

$$E_{total} = u. \sum_{i=1 \text{ to } N} (f_i \times R_i^{\,2}) \tag{3}$$

In (3), u is a factor and the amount of function f_i is calculated according to (4):

$$f_i = \begin{cases} 0 & \text{if node } n_i \text{ is in sleep mode} \\ 1 & \text{if node } n_i \text{ is an active node} \end{cases} \tag{4}$$

5. Proposed Protocol

In this section, we try to maintain sensing coverage by keeping a small number of active sensor nodes and a small amount of energy consumption. As mentioned before, the cluster-head will choose a set of active nodes to do the sensing job.

In proposed algorithm, at first a primary population of the active sensors set, are selected randomly. Each the active sensors set represents one configuration of sensor network. We consider the mentioned population as Harmony Memory. Then we try to improve the Harmony Memory by creating a new the active sensors set. If the new active sensor set is better than the worst active sensors set existing in Harmony Memory, the worst one will be replaced with it. The process of providing new active sensors set and replacing the worst active sensors set with the new one continue until meeting termination criteria. Algorithm continues until achieving a certain number of iteration or if the fitness value of one of the active sensors sets exceeds the threshold value. Finally, the best active sensors set existing in the Harmony Memory is selected as the response.

Phase1. Initialization:
- **Step1**: Initialize the problem and algorithm parameters.
- **Step2**: Initialize the harmony memory with the active sensors sets randomly.

Phase1. Repeating main loop of algorithm until meeting termination criteria:
- **Step3**: Improvise the new active sensors set.
- **Step4**: Update the harmony memory.
- **Step5**: Check the stopping criterion.

So, the proposed algorithm is presented briefly in five steps as follows:

5.1. Step1: Initialize the Problem and Algorithm Parameters

We use binary representation in which each bit corresponds to one sensor node. “1“means that corresponding sensor is an active node; otherwise, it is an inactive node.

Each the active sensors set, $A = (a_1, a_2, a_3, \dots , a_N)$, represents one configuration of sensor network such a way that the status of each ith node equals to a_i (*i.e.*, 0 or 1).

In this step, the optimization problem is specified as follows:

$$\text{Maximize } f(A) = C(A) \times (1/ (E_{total}(A) + \varepsilon_1)) \tag{5}$$

In (5), the amount of function $C(A)$ is calculated according to (6):

$$C(A) = \begin{cases} 1 & \text{if } A \text{ provides the acceptable coverage rate (a predetermined value)} \\ \varepsilon_2 & \text{else} \end{cases} \tag{6}$$

Where A is the active senor set; N is the number of nodes, a_i is the status of the sensor node n_i (*i.e.*, 1 or 0). Also ε_1 and ε_2 shows two very small positive numbers that should be selected properly such a way that $f(A)$ function value doesn't exceed the threshold value.

As mentioned before, $E_{total}(A)$ is the coverage energy consumption of the sensor set A based on (3). Therefore, $f(A)$ function value is in inverse ratio to the coverage energy consumption.

The HS algorithm parameters are also initialized in this step. These are the harmony memory size (HMS), or the number of the active sensors sets in the harmony memory; harmony memory considering rate (HMCR); pitch adjusting rate (*PAR*); and the number of improvisations (*NI*), or stopping criterion.

5.2. Step2: Initialize the Harmony Memory

The Harmony Memory (HM) is a memory location where all the active sensors sets are stored. In this step, we consider a Harmony Memory consists of one HMS group of the active sensors sets according to Figure 3. Each the active sensors set, $A^i = (a_1^i, a_2^i, a_3^i, \dots, a_N^i)$, represents one configuration of sensor network such a way that the status of each jth node equals to a_j^i. The fitness value of this configuration is shown by $f(A^i)$.

$$\mathrm{HM} = \left[\begin{array}{cccccccc|c} 0 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 & f(\mathrm{A}^1) \\ 1 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 & f(\mathrm{A}^2) \\ \vdots & & \vdots & & & \vdots & & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & f(\mathrm{A}^{HMS-1}) \\ 1 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & f(\mathrm{A}^{HMS}) \end{array}\right]$$

Figure 3. HM for HSSAC Algorithm

So, one N×HMS matrix of binary numbers will be obtained. This matrix is initialized randomly.

The fitness value of each the active sensors set A^i is represented by $f(A^i)$ and it is calculated by (7):

$$f(\mathrm{A}^i) = C(A^i) \times (1/(E_{total}(R^i) + \varepsilon_1)) \quad A^i = (a_1^i, a_2^i, a_3^i, \dots, a_N^i)\ , i = 1, 2, 3, \dots, \mathrm{HMS} \quad (7)$$

The amount of function $C(A^i)$ is calculated based on (6) and $E_{total}(A^i)$ is the coverage energy consumption of the sensor set A^i based on (3).

5.3. Step3: Improvise the Active Sensors Set

Generating the new active sensors set is called 'improvisation'. A new active sensors set, $A' = (a'_1, a'_2, \dots, a'_N)$, is generated based on three rules: memory consideration, pitch adjustment and random selection.

In the memory consideration, the status value of the first sensor node (a'_1) for the new active sensors set is chosen from any of the values in the specified HM range ($a'^1_1, a'^2_1, a'^3_1, \dots, a'^{\mathrm{HMS}}_1$). Values of the other sensing radiuses ($a'_2, a'_3, \dots, a'_N$) are chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, while (1 – HMCR) is the rate of randomly selecting one value from the possible range of values (i.e. 1 or 0).

For example, a HMCR of 0.85 indicates that the HS algorithm will choose the sensor status value from historically stored values in the HM with an 85% probability or from [0, 1] with a (100-85)% probability.

Every bit of the new active sensor set, $a'_i \in A'=(a'_1, a'_2, \dots, a'_N)$, obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment.

The value of PAR sets the rate of doing pitch adjustment, while value of (1-PAR) sets the rate of doing nothing. When the pitch adjustment operator is applied, if a'_i bit value is 0 is changed into 1 and vice versa.

In Step 3, HM consideration, pitch adjustment or random selection is applied to each variable of the new active sensors set in turn.

5.4. Sec Step4: Update Harmony Memory

If the new active sensor set, $A'=(a'_1, a'_2, \dots, a'_N)$, is better than the worst active sensors set existing in Harmony Memory, judged in terms of the objective function value $f(R')$, the worst one will be replaced with it.

5.5. Check Stopping Criterion

If the stopping criterion (*i.e.*, maximum number of improvisations) is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated. After ending the computation, the active sensors set with most objective function is selected from the HM active sensors sets.

6. Simulation Result

In this section, our proposed algorithm is simulated using NS2 simulator [18]. To evaluate the proposed protocol, it is compared with OGDC protocol [8]. In the simulation, we assume a target area with a size of 50×50 m^2. We deploy the sensor nodes randomly in the target area. The number of nodes, N, in different configurations are considered as 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340 and 350 respectively. Our proposed protocol is compared with OGDC protocol with two sensing radiuses with size of $R=4$ and $R=6$ m. Other assumptions are shown in Table 1.

The Figure 4 and Table 2 show the coverage rate for different configuration of the sensor networks when all sensor nodes are in active mode. For example in Figure 5, the sensor nodes are distributed in the environment randomly.

Table 1. Parameters Values

NI	HMS	HMCR	PAR	ε_1	ε_2	Acceptable coverage rate
300	50	0.85	0.02	1	0.001	95.00%

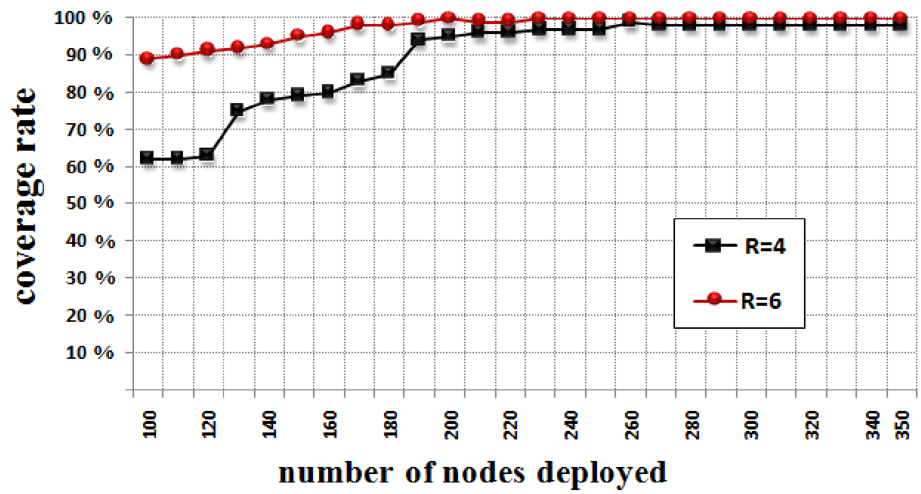


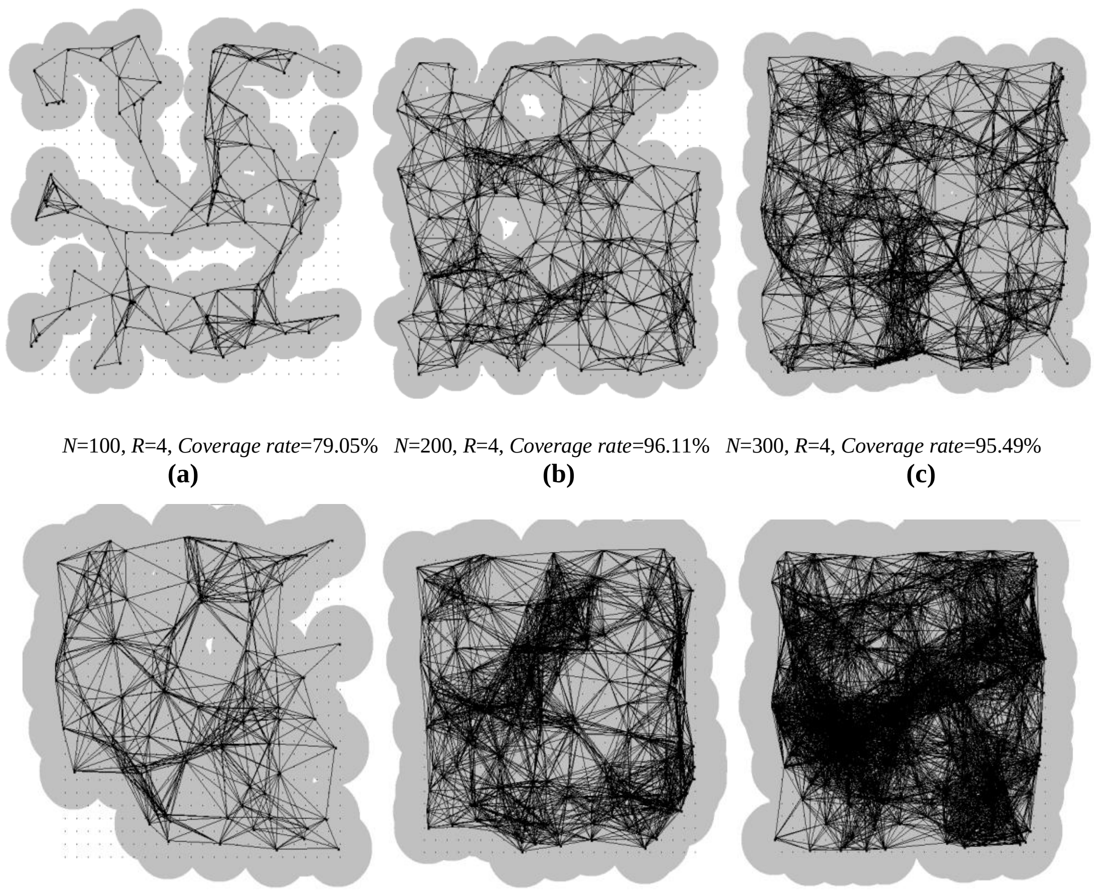
Figure 4. Verifying the Coverage Rate for Different Configuration of the Sensor Networks

Table 2. Verifying the Coverage Rate for Different Configuration of the Sensor Networks

The number of nodes (N)	*R*=4	*R*=6
50	61.92%	89.12%
60	62.40%	89.76%
70	63.20%	90.88%
80	75.36%	92.48%
90	77.60%	93.12%
100	79.04%	94.72%
110	80.16%	96.48%
120	83.52%	97.12%
130	85.12%	97.92%
140	89.92%	98.72%
150	92.96%	99.68%
160	93.28%	98.88%
170	93.92%	99.52%
180	94.40%	99.68%
190	94.72%	100.0%
200	96.00%	100.0%
210	96.48%	100.0%
220	97.12%	100.0%
230	97.44%	100.0%
240	97.28%	100.0%
250	99.20%	100.0%
260	98.88%	100.0%

270	98.72%	100.0%
280	97.92%	100.0%
290	98.56%	100.0%
300	99.52%	100.0%
310	99.20%	100.0%
320	99.68%	100.0%
340	99.04%	100.0%
350	99.36%	100.0%

As shown in Figure 6, Figure 7 and Table 3, due to the proposed protocol accuracy in selecting the active sensor set, it is able to provide the acceptable coverage rate in sparse deployment. As the result of increasing nodes density, the proposed protocol decreases the number of active nodes in the sensor networks. Thus, the proposed protocol decreases the energy consumption of the networks and prolongs the network lifetime (see Figure 8 and Table 4).



N=100, *R*=4, *Coverage rate*=79.05% **(a)** *N*=200, *R*=4, *Coverage rate*=96.11% **(b)** *N*=300, *R*=4, *Coverage rate*=95.49% **(c)**

N=100, *R*=6, *Coverage rate*=94.62% **(d)** *N*=200, *R*=6, *Coverage rate*=100% **(e)** *N*=300, *R*=6, *Coverage rate*=100% **(f)**

Figure 5. The Coverage Rate for the Sensor Network (all Sensor Nodes are in Active Mode)

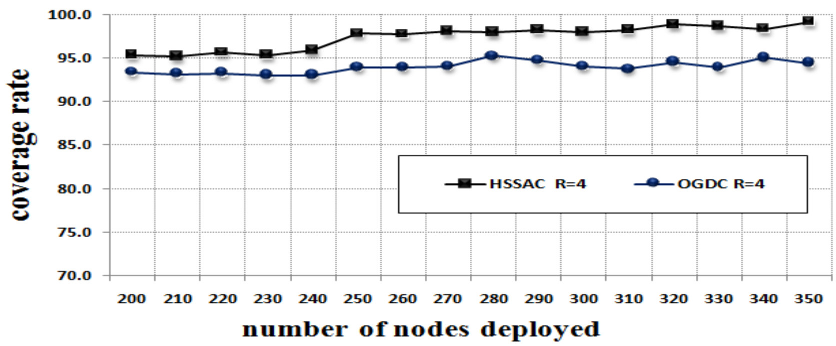


Figure 6. The Coverage Rate of the Active Sensor Set in Different Configurations (R=4)

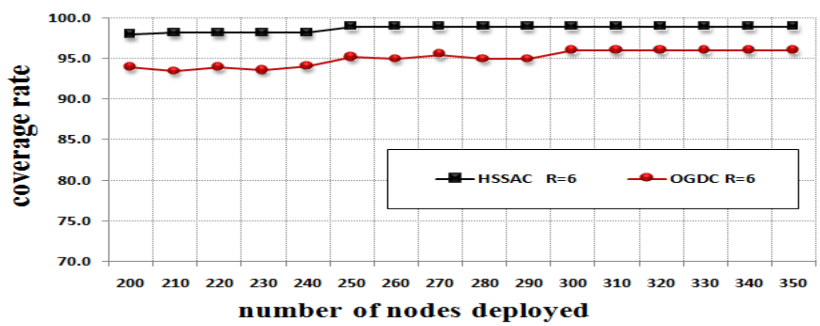


Figure 7. The Coverage Rate of the Active Sensor Set in Different Configurations (R=6)

Table 3. The Coverage Rate of the Active Sensor Set in Different Configurations

The number of nodes (N)	R=4		R=6	
	HSSAC	OGDC	HSSAC	OGDC
200	95.41%	93.41%	98.00%	94.19%
210	95.23%	93.22%	98.21%	93.51%
220	95.74%	93.36%	98.23%	94.23%
230	95.42%	93.15%	98.31%	93.62%
240	95.91%	93.92%	98.65%	94.19%
250	97.89%	94.00%	99.07%	95.23%
260	98.87%	94.17%	99.00%	95.62%
270	98.17%	94.52%	99.16%	95.18%
280	98.00%	95.32%	99.32%	95.43%
290	98.03%	94.19%	99.26%	95.12%
300	98.94%	93.81%	99.31%	96.32%
310	98.21%	94.53%	99.18%	96.54%
320	98.56%	94.51%	99.17%	95.98%
340	98.80%	95.12%	99.42%	96.12%
350	99.11%	94.87%	99.56%	96.18%

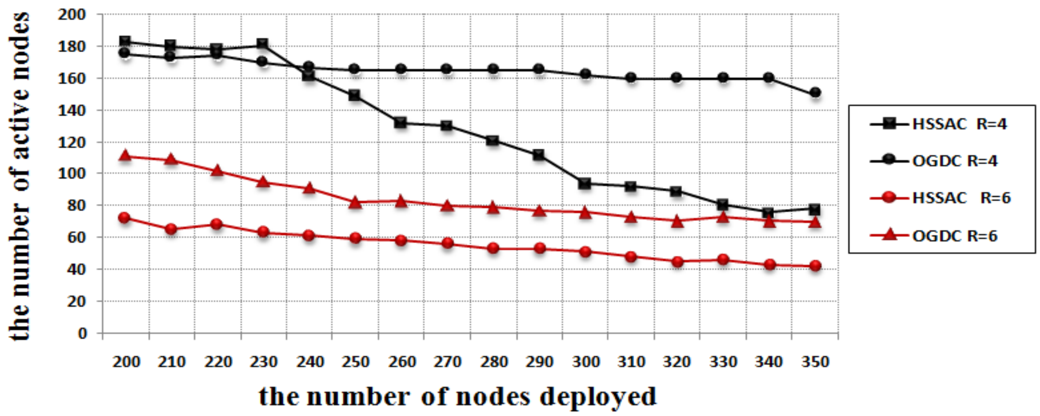


Figure 8. The Number of Active Nodes in Different Configurations

Table 4. The Number of Active Nodes in Different Configurations

The number of nodes (N)	R=4		R=6	
	HSSAC	OGDC	HSSAC	OGDC
200	183	175	72	111
210	180	173	65	109
220	178	174	68	102
230	181	170	63	95
240	161	167	61	91
250	149	165	59	82
260	132	164	58	83
270	130	165	56	80
280	121	165	53	79
290	112	163	52	77
300	94	162	50	76
310	92	160	48	73
320	89	161	45	71
330	81	160	46	73
340	76	159	43	71
350	78	151	41	70

6. Conclusion

In this paper, we have proposed HSSAC protocol to maintain sensing coverage by keeping a small number of active sensor nodes and a small amount of energy consumption in a wireless sensor network. HSSAC use a cluster-based coverage control scheme and determine the active sensor set using Harmony Search algorithm. Due to the proposed protocol accuracy in selecting the active sensor set, it is able to provide the acceptable coverage rate in sparse deployment. As the result of increasing nodes density, the proposed protocol decreases the number of active nodes in the sensor networks. Thus, the proposed protocol decreases the energy consumption of the networks and prolongs the network lifetime. We have simulated our protocol and simulation results show high efficiency of the proposed protocol.

References

[1] Y. Li, C. Ai, Z. Cai and R. Beyah, "Sensor scheduling for p-percent coverage in wireless sensor networks", in: Springer Science Business Media, LLC, **(2009)**.
[2] J. Jia, J. Chena, G. Changa, Y. Wena and J. Songa, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius", Elsevier, Computers and Mathematics with Applications, vol. 57, **(2009)**, pp. 1767-1775.
[3] X. Zhang, X. Ding, S. Lu and G. Chen, "Principles for Energy-Efficient Topology Control in Wireless Sensor Networks", WiCom '09. 5th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, **(2009)** September 24-26.
[4] M. Esnaashari and M. R. Meybodi, "A learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks", Computer Networks journal (Elsevier), **(2010)**.
[5] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks", Proc. of IEEE International Conference on Communications, ICC, Helsinki, Finland, **(2001)**, pp. 472-476.
[6] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks", Proc. of the First ACM Intel workshop on Wireless Sensor Networks and Applications, New York, USA, **(2002)**, pp. 32-41.
[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks", Proc. of the First international conference on Embedded networked sensor systems, Los Angeles, California, USA, **(2003)**, pp. 28-39.
[8] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks", Ad-hoc and Sensor Wireless Networks, vol. 1, **(2005)**, pp. 89-124.
[9] S. Yang, F. Dai, M. Cardei, J. Wu and F. Patterson, "On connected multiple point coverage in wireless sensor networks", Wireless Information Networks, vol. 13, **(2006)**, pp. 289-301.
[10] F. Ye, G. Zhong, J. Cheng, S. Lu and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks", Proc. of the 23rd International Conference on Distributed Computing Systems, ICDCS, Providence, USA, **(2003)**, pp. 28-37.
[11] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search", in: Elsevier, Simulation, vol. 76, no. 2, **(2001)**, pp. 60-68.
[12] B. Alatas, "Chaotic harmony search algorithms", Elsevier, Applied Mathematics and Computation, vol. 216, **(2010)**, pp. 2687-2699.
[13] M. G. H. Omran and M. Mahdavi, "Global-best harmony search", Elsevier, Applied Mathematics and Computation, vol. 198, **(2008)**, pp. 643-656.
[14] M. Mahdavi, M. Fesanghary and E. Damangir, "An improved harmony search algorithm for solving optimization problems", Elsevier, Applied Mathematics and Computation, vol. 188, **(2007)**, pp. 1567-1579.
[15] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less low-cost outdoor localization for very small devices", IEEE Personal Communications, vol. 7, **(2000)**, pp. 28-34.
[16] J. Jia, J. Chen, Y. Wen and G. Chang, "An extensible core-control routing protocol in large scale ad-hoc networks", Proc. of the 6th International Conference on ITS Telecommunications, Chengdu, China, **(2006)**, pp. 955-958.
[17] M. Lu, J. Wu, M. Cardei and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor networks", Proc. of the International Conference on Computer Networks and Mobile Computing, ICCNMC, Zhangjiajie, China, **(2005)**, pp. 43-52.
[18] The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/.

Authors

Hagar Noori received the B.S. degree in Computer Engineering from the Islamic Azad University and the M.S. degrees in Computer Engineering from Arak Science and Research Branch in Iran, in 2007 and 2012, respectively. Her research interests include wireless sensor and Ad-hoc networks.

Reza Rafeh is a faculty member in the Department of Computer Engineering at the Faculty of Engineering in Arak University (Iran). He got his BSc and MSc in Software Engineering from Sharif University of Technology (Iran). He received his PhD in computer science from Monash University (Australia). His interesting areas are compiler design, constraint programming and theorem proving. He is the chief editor of Software Engineering Journal as well as an editorial board of Soft Computing Journal.