Cellular Learning Automata based Scheduling Method for Wireless Sensor Networks

M. Jahanshahi[1] M. R. Meybodi[2] M. Dehghan[2]

[1]Department of Computer Engineering, Islamic Azad University, Science and Research branch, Tehran, Iran

[2]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

*mjahanshahi@sr.iau.ac.*ir,*mmeybodi@aut.ac.ir,dehghan@aut.ac.ir*

Abstract

In wireless sensor network often micro-battery with very limited power provides the energy of sensor nodes. Since sensors are usually utilized in remote or hostile environments, recharging or replacing the battery of the sensors is something quite undesirable or even impossible. Thus long system lifetime is a must. Sleep scheduling is a mechanism in wireless sensor network to save energy. *In this paper, we propose an energy-efficient distributed scheduling method considering mobile target tracking also called dynamic target coverage. The algorithm is based on cellular learning automata. In this algorithm, each node is equipped with a learning automaton which will learn (schedule) the proper on and off times of that node based on the movement nature of a single moving target. To evaluate the proposed method it is tested under straight with constant velocity movement model of target. The results of experimentations have shown that the proposed scheduling algorithm outperforms two existing dynamic target coverage scheduling methods.*

1. Introduction

Sensors are scattered in a hostile environment in order to perform some tasks such as data gathering or monitoring. How well a sensor network can collect sensory data depends on its sensing coverage. Coverage problem deals with the ability of the network to cover a certain area or some certain events. The coverage problems in sensor networks can be classified into area coverage, barrier coverage, and point (target) coverage[1,14,20,21,31]. The main objective of area coverage which is the most studied coverage problem is to cover (monitor) an area. Barrier coverage can be considered as the coverage with the goal of minimizing the probability of undetected penetration through the barrier (sensor network) and the objective of point (target) coverage problem is to cover a set of stationary or moving points in the area of sensor network using as few sensor nodes as possible. Target coverage methods can be classified into two categories named static point coverage [20] and dynamic point coverage. In static target coverage, target is not moving where as in dynamic point coverage the target is capable of moving.

The problem of point coverage can be addressed in many different ways. One approach is to design a deployment strategy which can best address the criterion of minimum number of required nodes [11-13,16-18]. This solution is static in its nature, since has nothing to do with changes which are commonly occurred in sensor networks due to high probability of sensor failures throughout the lifetime of the network. Another more dynamic solution to the point coverage problem is the rearrangement of nodes assuming movement ability for them [22-25,27]. In this approach, nodes are first deployed randomly around the target points. After this initial random positioning, each node tries to find its best position according to the position of target points and other nodes around it. It is straight forward that using this approach, dynamic changes in the topology of the network or position of the targets can be adaptively addressed. One major problem with this solution is the high overhead of controlling packets which are required to control the position of each node according to the position of its neighbors. This can lead to fast energy exhaustion of nodes, and hence shortening the lifetime of the network. Many researchers try to overcome the point coverage problem by designing a suitable scheduling strategy for making nodes *on* and *off* in such a way that in each time slice, only nodes which can sense the target points in that period are *on* [14,21,1,31]. In this method, many nodes are scattered randomly throughout the filed, and the criterion of minimum number of required nodes addressed by the scheduling mechanism which tries to minimize the number of nodes being in *on* state according to their ability to sense the target points. In terms of dynamicity, this solution can deal with changes occurred in the topology of the network and position of target points using a dynamic scheduling strategy. In terms of overhead and energy consumption, this method best fit the sensor networks, since in each time slice, only nodes which can sense the target points in that period are *on*, and rest of the nodes are in *off* state, reserving their energy. If the sensors which are away from mobile target are active, they waste their power without being useful. Hence it is generally difficult to optimize energy efficiency and simultaneously track target without missing [37]. Centralized scheduling algorithms are suitable only for stationary targets or moving targets with known and static movement paths. For targets with unknown and dynamically changing movement paths, decentralized and more flexible scheduling algorithms are required.

To our best knowledge, in all works done in this area, some sort of notification messages are exchanged between nodes currently in *on* state, monitoring the target points, and their neighbor nodes which are currently in *off* state, but with high probability should become *on*, since target points are moving towards them. This has two drawbacks; one is the overhead of these notification messages and the other is that nodes in off state should have the ability to receive these messages, and hence they cannot power off their receiving antenna. This means that these nodes just switch off their processing unit, but their communication units are in idle state. According to [28], energy consumption of a sensor node in receiving and idle states is nearly equal to the energy consumed during transmission state.

Some dynamic point coverage methods, schedule the sleep pattern based on only the distance of a node from the target's instant position: the closer the node is from the target, the lighter its sleep level should be. Such a sleep scheduling algorithm is usually called "circle-based scheme" or "legacy scheme" [14,21]. PECAS (Probing Environment and Collaborating Adaptive Sleeping) is an example of a "circle-based scheme" method [14]. In this method a sensor enters the tracking mode and remains active when it senses a target during wake-up period. The network operations have two states. One is the surveillance state during which there are no events of interest in the field, but the sensors are ready to detect any possible occurrences. The other state is the tracking state during which the network reacts in response to any moving targets, and the sensors are collaborating in measuring the target's path and speed. PECAS uses a sleep-awake protocol which provides better quality of surveillance while reducing the power consumption. According to sleep-awake protocol a working node will go to *sleep* mode after a specified period of time. It also advertises the remaining working time in its reply messages to its neighbors' probe messages. In this way, a working neighbor who decides to enter the *sleep* mode can schedule itself to wake up before this node goes to *sleep*, thus preventing the occurrence of blind spots. With these schemes, all the nodes in a circle will take the same sleep pattern regardless to moving directions which are undesired and different from our work.

Before describing the describing the work proposed in this paper we explain the modes that a sensor node may be in. A sensor node may be in four different modes: **on duty**, **sensing unit on duty**, **transceiver on duty** and **off duty** [29]. In **on-duty** mode which is not an energy-saving mode, all the components in the sensor are turned on. The sensor is able to collect sensory data, send/receive messages, process data and messages, and do other types of computation. In **sensing unit on-duty (SU-on-duty)** mode at least one sensing unit and the processor are turned on, but the transceiver is turned off. In this mode, the sensor is capable of sensing and processing sensory data, but not transmitting or receiving messages. In **transceiver on-duty (TR-on-duty)** the transceiver and the processor are turned on, but all the sensing units are turned off. In this mode, the sensor is capable of transmitting, receiving and processing messages, but not sensing. Sensors save more energy in the SU-on-duty mode than in the TR-on-duty mode, because communication usually consumes more energy than sensing. In **off-duty** mode: the sensor's processor is turned off, but a timer or some other triggering mechanism may be running to wake up the sensor. Some sensors have 3 off-duty modes called Monitor, Observe, and Deep Sleep, each with a different wakeup mechanism. The processor is turned off in all three modes, so the sensor cannot process any sensory data or messages. However, in the Monitor mode, both the sensing unit and the transceiver are left on to receive wakeup signals. In the Observe mode, only the sensing unit is on. Note that the Observe mode is different from the SU-on-duty mode as the processor is turned on in the latter. In the Deep Sleep mode, neither the sensing unit nor the transceiver is turned on, so the sensor relies on a preset internal timer to wake itself up. Some researchers also designed a fourth sleep mode called the **Listen** or **Idle** mode. In this mode, only the transceiver is turned on to receive wakeup signals. That is the processor and sensing unit are off.

In this paper we propose a non-circle based dynamic target coverage scheduling method in which each node uses a learning automaton to learn (schedules) its proper *On-Duty* and *Deep Sleep* times based on the nature of movements of target points. The method is distributed and uses the local information.

The proposed method like the methods reported in [1] and [31] is based on S-MAC protocol [32] of wireless sensor networks. In S-MAC protocol, each operating period which is called a Round is divided into two phases called phase (A) and (B). In phase (A), the learning automaton or ant agent resident in each node helps the node to determine its next mode. In phase (B), switching between modes occurs. To evaluate the proposed method it is tested under straight with constant velocity movement model of target. The results of experimentations have shown that the proposed scheduling algorithm outperforms two existing non circle based dynamic target coverage scheduling methods reported in [1][31].

The rest of this paper is organized as follows. In section 2 related works are presented. Cellular learning automata will be briefly discussed in section 3. In section 4 the proposed method is presented. Simulation results are given in section 5. Section 6 is the conclusion.

2. Related works

In this paper we are interested in non-circle based dynamic point coverage scheduling problem in which a single target point is moving throughout the area of the sensor network and should be detected by nodes which are close enough and have the ability to sense its movement. Below two non-circle based dynamic target coverage scheduling methods with which the proposed methods in this paper are compared will be explained.

A non-circle based dynamic target coverage scheduling method which is based on swarm optimization is reported in [1]. In this method each sensor node in the sensor network is mapped to an ant in ant colony system and node communication information is modeled by the current pheromone. The sleep scheduling problem of the sensor network is modeled as a swarm intelligence optimization problem. Hereafter this method is referred to SWIB (SWarm Intelligent Based method).

Another non-circle based dynamic target coverage scheduling method is reported in [31]. In this method only two modes *On* and *Off* are considered for every sensor node. *On* and *Off* modes are in fact the same as *Deep Sleep* and *On-Duty* modes respectively. In this method each node *i* in the network is equipped with a learning automaton. This learning automaton helps the node in determining it's suitable *Off* state period. Hereafter this method is referred to LAB (Learning Automata Based method).

3. Cellular Learning Automata

In this section cellular learning automata will be briefly reviewed.

Cellular Learning Automata: Cellular automata [6-10] are mathematical models for systems consisting of large numbers of simple identical components with local interactions. The simple components act together to produce complicated patterns of behavior. CA perform complex computation with high degree of efficiency and robustness. CA are non-linear dynamical systems in which space and time are discrete. A CA consists of a finite dimensional lattice of cells whose states are restricted to a finite set of integers. The state of each cell at any time instant is determined by a rule from states of neighboring cells at the previous time instant. Given a finite set and a finite dimension *d,* CA can be considered as a d-dimensional lattice. A *cellular learning automata* (CLA) is a CA in which an LA (or multiple LA) is assigned to its every cell. The LA residing in a particular cell determines its state on the basis of its action probability vector. Like CA, there is a rule that CLA operate under it. The rule of CLA and state of neighboring cells of any particular cell determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring cells of any particular cell constitute the environment for that cell. Because the neighboring cells produce the reinforcement signal to the LA residing in that cell. The operation of CLA could be described as follows: At the first step, the internal state of every cell is specified. The internal state of every cell is determined on the basis of action probability vectors of learning automata residing in that cell. The initial value of the action probability vectors may be chosen on the basis of past experience or at random. In the second step, the rule of CLA determines the reinforcement signal for each LA. Finally, each LA updates its action probability vector on the basis of supplied reinforcement signal and the action chosen by the cell. This process continues until the desired state reached. The CLA can be classified into *synchronous* and *asynchronous* CLA. In synchronous CLA, all cells are synchronized with a global clock and activated at the same time.

A CLA consists of following major components: neighboring cells, local rule, and a set of LA. The operation of CLA can be described as follows: Without loss of generality assume that an automaton is assigned to every cell. At instant *n,* the LA *A* associated to a particular cell *u* selects one of its actions, say a(*n)* based on its action probability vector. The reinforcement signal *f(n)* is produced by a local rule *R* from the state of neighborhood cells. Every LA updates its action probability vector based on the reinforcement signal and selected action. This process continues until the average received penalty is minimized.

Irregular Cellular Learning Automata: An Irregular cellular learning automata (ICLA) is a cellular learning automata (CLA) in which the restriction of rectangular grid structure in traditional CLA is removed. This generalization is expected because there are applications such as wireless sensor networks, immune network systems, graph related applications, etc. that cannot be adequately modeled with rectangular grids. An ICLA is defined as an undirected graph in which, each vertex represents a cell which is equipped with a learning automaton. The learning automaton residing in a particular cell determines its state (action) on the basis of its action probability vector. Like CLA, there is a rule that the ICLA operate under. The rule of the CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell. The neighboring LAs of any particular LA constitute the local environment of that cell. The local environment of a cell is non-stationary because the action probability vectors of the neighboring LAs vary during the evolution of the ICLA [30].

4. The Proposed Methods

In the remaining part of this paper a non circle based dynamic target coverage scheduling method for a sensor network is presented. The proposed method is based on cellular learning automata. In this method which we call it DTTCLA (Dynamic Target Tracking based on Cellular Learning Automata) each sensor node is equipped with a learning automaton and assume that each sensor has four modes called *SU-On-Duty*, *TR-On-Duty*, *Deep Sleep* and *On-Duty*. The learning automaton in each sensor node s_i of sensor network, referred to as LA_i, has two actions a_0 and a_1. Action a_1 is *Deep Sleep* and action a_0 is *On-Duty*. The probability of selecting each of these actions is initially set to 0.5. The set of learning automata in the sensor nodes forms an irregular cellular learning automaton (ICLA). That is two LA_i and LA_j in ICLA are adjacent to each other if nodes s_i and s_j in the sensor network are close enough to hear each other's signal. The operation of the proposed algorithm is divided into rounds. A round of the proposed method which runs on each sensor node is explained as follows. All the sensor nodes are initially in mode *SU-On-Duty*, that is it senses its covered area. In this mode the *learning automaton* LA_i *of each sensor node* s_i is activated and chooses an action. Then the sensor goes to mode *TR-On-Duty* during which it computes its awake neighbors. Depending on the result of sensing the coverage area, the action chosen by the learning automaton during *SU-On-Duty* mode and the number of awake neighbors which is computed in mode *TR-On-Duty* one of the following takes place.

1-If target is in the sensing area of the sensor, its resident learning automaton have chosen the *On-Duty* action and the number of awake neighbors is less than *n*, then the sensor switch its mode to *SU-On-Duty* again and reward the action chosen by its learning automaton according to L_{R-P} learning algorithm and goes to *On-Duty* mode.

2- If target is not in the sensing area of sensor and its resident learning automaton have chosen the *On-Duty* action, then the sensor switch its mode to *SU-On-Duty* again and penalize the action chosen by its learning automaton according to L_{R-P} learning algorithm and goes to *On-Duty* mode.

3- If target is in not the sensing area of sensor and its resident learning automaton have chosen the *Deep Sleep* action, then the sensor switch its mode to *SU-On-Duty* again and reward the action chosen by its learning automaton according to L_{R-P} learning algorithm and goes to *Deep Sleep mode.*

4- If target is in the sensing coverage of sensor, its resident learning automaton have chosen the *Deep Sleep* action and the number of awaken neighbors is less than *n*, then the sensor switch its mode to *SU-On-Duty* again and penalize the action chosen by its learning automaton according to L_{R-P} learning algorithm and goes to *Deep Sleep* mode.

5. Simulation Results

In order to evaluate the proposed method we simulate them using Vergil simulator [34-36]. Simulations are conducted for a network with 15 sensor nodes which are uniformly scattered in an area of 100*100 square meters. We also define two radio ranges named *Range* and *SureRange.* The former is set to 22 m and the later is equal to 20 m. If destination of the packet is in *SureRange* distance, then the packet with probability of 1 will be delivered. But if destination is out of *SureRange* and in the *Rang,* then the packet with probability of (*1-p)* will be lost. *p* is computed using equation 3 given below[34-36].

$$p = \frac{(PI * SureRange^2)}{(PI * Range^2)} \qquad (3)$$

The tracking field is flat and thus a 2D mesh model is used to describe the network, and energy consumption ratio is similar to CISCO Aironet 350 card which is shown in table 1[33]. Each sensor node uses two wireless channels; one for sensing the environment and one for collaboration between nodes. For all simulations for DTTCLA method, *n*, the number of awake neighbors, is set to 3.

Table 1. Amount of energy consumption in each mode of a sensor

Deep Sleep	Idle	On-Duty	TR_On_Duty
0.07	1	2	2.5

Experiment for straight with constant velocity movement model: This experiment is conducted to study the network utilization defined as the ratio of the number of useful *on* sensor nodes to total *on* sensor nodes, target missing rate, and energy consumption. For this experiment we assume that the target moves in a straight line with constant velocity. Figure 7 compares the network utilization for the proposed algorithm, SWIB and LAB. As it can be seen DTTCLA method with utilizations 0.54, outperforms the existing methods.

Despite the fact that node sleeping helps conserve energy, it may have a negative impact on the network performance. For example sensor sleeping may result in interesting events being missed by the network or may lead to a lower quality of data being sensed [10]. Figure 8 depicts target tracking missing rate of the proposed method and compare the results with the results obtained for methods SWIB and LAB. The results demonstrate that target missing rate for the proposed method is less than that of SWIB and LAB.

Figure 9 compares the methods in terms of energy consumption. After method SWIB, method DTTCLA comes. Although DTTCLA method comparing SWIB consumes more energy, it considering target tracking missing rate and utilization outperforms the SWIB method.

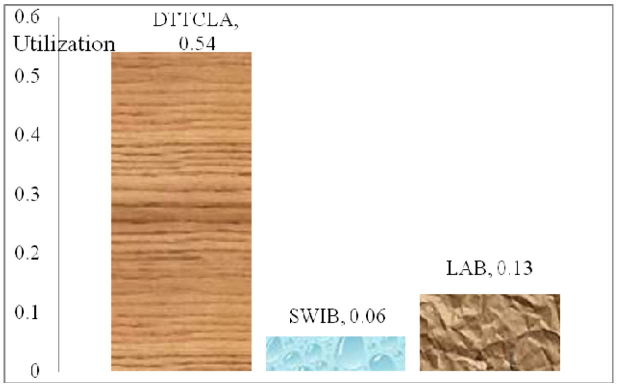

Fig 7: Utilization for different methods

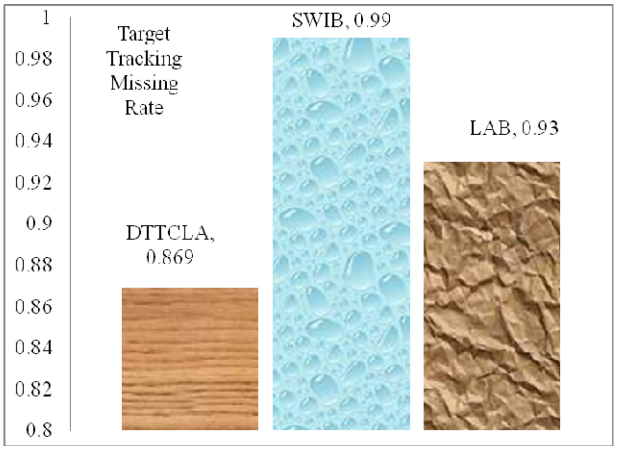

Fig 8: Target tracking missing rate for all methods

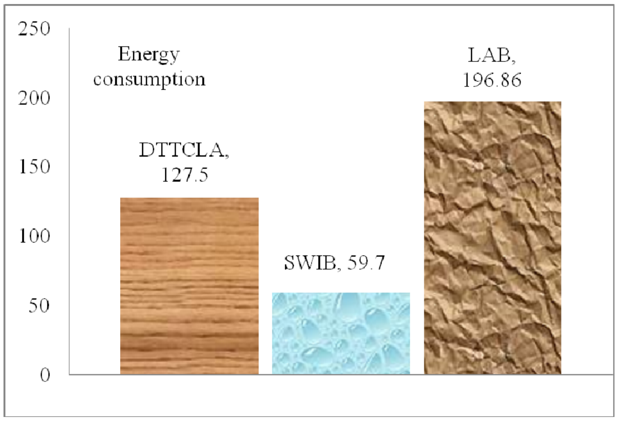

Fig 9: Energy consumption for different methods

In table 2 the methods are ranked according to different criteria. Rank 1 is given to the best method and 3 to the worst method. Table 3 gives the sum of rankings of each method. For example for method DTTCLA the sum of rankings is (1+2+1) = 4. According to the sum of ranking criteria the best method is DTTCLA and the proposed method outperforms methods reported works in [1][31].

Table 2: Evaluation of the proposed method as compared with reported works in [1] and [35]

	Utilization	Power consumption	Missing Rate
DTTCLA	1	2	1
LAB	2	3	2
SWIB	3	1	3

Table 3: Quantify the quality of all methods.

Method name	DTTCLA	LAB	SWIB
Sum of rankings	4	7	7

6. Conclusion

In this paper, an energy-efficient distributed scheduling method for mobile target tracking was proposed. In the proposed method which is based on cellular learning automata, each sensor node uses information about its neighboring sensor nodes to adaptively determine the proper *on* and *off* times of the node based on the movement nature of a single moving target. To evaluate the proposed method it is tested under straight with constant velocity movement model of target. Computer simulation showed that the proposed scheduling algorithm outperforms two existing dynamic target coverage scheduling methods.

7. References

[1] W. Rui, L. Yan, Y. Gangqiang, L. Chaoxia and P. Quan, "Swarm Intelligence for the Self-Organization of Wireless Sensor Network", IEEE Congress on Evolutionary Computation, Canada, 2006.
[2] K. S. Narendra and M. L. Thathachar, “Learning automata: an introduction”, Englewood Cliffs, NJ: Prentice Hall, 1989.
[3] K. Najim and A. S. Poznyak, “Learning Automata: Theory and Application”, Tarrytown, NY: Elsevier Science Ltd., 1994.
[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next Century Challenges: Scalable Coordination in Sensor Networks”, MobiCom, pp. 263-270, 1999.
[5] G. J. Pottie and W. J. Kaiser, “Wireless Integrated Network Sensors”, ACM, vol. 43, no. 5, pp. 51-58, 2000.
[6] H. Beigy, and M. R. Meybodi, , "Open Synchronous Cellular Learning Automata", Journal of Computer Science and Engineering, vol. 1, no. 4 (b), pp. 39-51, 2003.
[7] R. Rastegar, M. Rahmati and M. R. Meybodi, "A Clustering Algorithm Using Cellular Learning Automata Based Evolutionary Algorithm", Adaptive and Natural Computing Algorithms, Springer Verlag, pp.144-150, 2005.
[8] R. Rastegar, M. R. Meybodi, and A. Hariri, "A New Fine Grained Evolutionary Algorithm based on Cellular Learning

Automata", International Journal of Hybrid Intelligent Systems, IOS Press, vol 3, no. 2, pp. 83-98, 2006.

[9] M. Bohlool and M. R. Meybodi, "Edge Detection Using Open and Asynchronous Cellular Learning Automata", 4th Iranian Conference on Machine Vision and Image Processing, Mashhad, Iran, 2007.

[10] S. Noferesti and M. R. Meybodi, "Simulation of Pesticides Percolation in the Soil Using Cellular Learning Automata", 12th Annual CSI Computer Conference of Iran, Shahid Beheshti University, Tehran, Iran, pp. 2247-2251, 2007

[11] S. S. Dhillon, K. Chakrabarty and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections", Proc. International Conference on Information Fusion, pp. 1581-1587, 2002.

[12]Y. Zou and K. Chakrabarty, "Uncertainty-aware and coverage-oriented deployment for sensor networks", Journal of Paralle and Distributed Computing, Vo. 64, No. 7, pp. 788-798, 2004.

[13]H. Chen, H. Wu and N-F. Tzeng, "Grid-based Approach for Working Node Selection in Wireless Sensor Networks", Proc. of IEEE International Conference on Communications, 2004.

[14] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks",10th Annual International Conference on Mobile Computing and Networking (Mobicom '04), pp. 129–143, 2004.

[15] D. Tian and N. D. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks", 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02), pp. 32–41, 2002.

[16] Y. Zou and K. Chakrabarty, "A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks", IEEE Transactions on Computers, Vol. 54, No. 8, pp. 978-991, 2005.

[17] F. Pedraza, A. García and A. L. Medaglia, "Efficient coverage algorithms for wireless sensor networks", Proc. of the 2006 Systems and Information Engineering Design Symposium, 2006.

[18]Zh. Dingxing, X. Ming, Ch. Yingwen and W. Shulin, "Probabilistic Coverage Configuration for Wireless Sensor Networks", 2nd International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, 2006.

[19] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, 2004.

[20] M. Cardei and D. Du, "Improving Wireless Sensor Network Lifetime through Power Aware Organization", ACM Wireless Networks, vol. 11 no.3, pp. 333-340, 2005.

[21] X. Wang et al., "Prediction-based Dynamic Energy Management in Wireless Sensor Networks", Sensors, 7, pp. 251-266, 2007.

[22] M. Schwager, J. McLurkin and D. Rus, "Distributed Coverage Control with Sensory Feedback for Networked Robots", Proc. of Robotics: Science and Systems, Philadelphia, 2006.

[23]M. A. Batalin and G. S. Sukhatme, "Multi-robot Dynamic Coverage of a Planar Bounded Environment", *CRES-03-011*, 2003.

[24]B. Jung, "Cooperative Target Tracking using Mobile Robots", *PhD dissertation proposal*, University of Southern California, 2004.

[25]B. Shucker and J. K. Bennett, "Target Tracking with Distributed Robotic Macrosensors", *Proc. of MILCOM 2005*. Atlantic City, New Jersey, 2005.

[26] C. F. Chiasserini and M. Garetto, "An Analytical Model for Wireless Sensor Networks with Sleeping Nodes", IEEE Transactions on Mobile Computing, vol. 5, No. 12, 2006.

[27]R. Olfati-Saber, "Distributed Tracking for Mobile Sensor Networks with Information- Driven Mobility", Proc. of the 2007 American Control Conference, New York, NY, pp. 4606–4612, 2007.

[28] V. Raghunathan, C. Schurgers, S. Park and M. B. Srivastava, "Energy-Aware Wireless sensor networks", *IEEE Singal Processing Magazine*, 2002.

[29] L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks", Mobile Networks and Applications, Vol. 11, Issue 5, pp. 723 - 740, 2006.

[30] M. Esnaashari and M. R. Meybodi, "A Cellular Learning Automata based Clustering Algorithm for Wireless Sensor Networks", Sensor Letters, *Vol. 6, No. 5, pp. 723-735, October 2008.*

[31] M. Esnaashari and M. R. Meybodi, "Dynamic Point Coverage in Wireless Sensor Networks: A Learning Automata Approach", Springer Verlag, Proceedings of 13th International CSI Computer Conference of Iran, Kish Island, Iran, 2008.

[32] Y. Wang, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", 21 Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings of INFOCOM, Vol 3, pp. 1567 – 1576, 2002.

[33] P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks", Wiley, ISBN 0470094532, 2005

[34]C. Brooks et al. , "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II)", EECS Department, University of California, Berkeley, UCB/EECS-2008-37, 2008. (http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-37.html)

[35]E. Cheong et al. ,"Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks, EECS Department, University of California, Berkeley, UCB/EECS-2006-15, 2006.(http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-15.html)

[36] C. Brooks et al. , "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)", EECS Department, University of California, Berkeley, UCB/EECS-2008-37,2008) http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-37.html)

[37] S. Chachra and M. Marefat, "Distributed Algorithms for Sleep Scheduling in Wireless Sensor Networks", IEEE International Conference on Robotics and Automation Orlando, Florida, 2006.