Iranian Journal of Mathematical Sciences and Informatics
Vol. 10, No. 1 (2015), pp 81-93
DOI: 10.7508/ijmsi.2015.01.006

Gravitational Search Algorithm to Solve the K-of-N Lifetime Problem in Two-Tiered WSNs

Marjan Kuchaki Rafsanjani[a,*], Mohammad Bagher Dowlatshahi[a], Hossein Nezamabadi-Pour[b]

[a]Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.
[b]Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.

E-mail: kuchaki@uk.ac.ir
E-mail: mb.dowlatshahi@yahoo.com
E-mail: nezam@uk.ac.ir

Abstract. Wireless Sensor Networks (WSNs) are networks of autonomous nodes used for monitoring an environment. In designing WSNs, one of the main issues is limited energy source for each sensor node. Hence, offering ways to optimize energy consumption in WSNs which eventually increases the network lifetime is strongly felt. Gravitational Search Algorithm (GSA) is a novel stochastic population-based meta-heuristic that has been successfully designed for solving continuous optimization problems. GSA has a flexible and well-balanced mechanism to enhance intensification (intensively explore areas of the search space with high quality solutions) and diversification (move to unexplored areas of the search space when necessary) abilities. In this paper, we will propose a GSA-based method for near-optimal positioning of Base Station (BS) in heterogeneous two-tiered WSNs, where Application Nodes (ANs) may own different data transmission rates, initial energies and parameter values. Here, we treat with the problem of positioning of BS in heterogeneous two-tiered WSNs as a continuous optimization problem and show that

*Corresponding Author

Received 20 July 2013; Accepted 11 June 2014

proposed GSA can locates the BS node in an appropriate near-optimal position of heterogeneous WSNs. From the experimental results, it can be easily concluded that the proposed approach finds the better location when compared to the PSO algorithm and the exhaustive search.

Keywords: Wireless sensor network (WSN), Two-tiered WSNs, Base station location, Energy consumption, Network lifetime, Gravitational search algorithm (GSA).

2000 Mathematics subject classification: 68M10, 90B18.

1. Introduction

Wireless Sensor Networks (WSNs) are networks of distributed autonomous nodes that can sense their environment cooperatively. WSNs are used in diverse applications such as environment and habitat monitoring, structural health monitoring, healthcare, home automation, and traffic surveillance. These networks with their applications have created a small revolution in the evolution of information and hence those are an attractive field for computer science and engineering researchers [3]. In designing WSNs, one of the main issues is limited energy source for each sensor. Moreover, due to the large number of sensors in the network or lack of access to them, battery replacement for sensors is not practical. Hence, offering ways to optimize energy consumption, which eventually increases the network lifetime is strongly felt [3].

A two-tiered WSN consists of a number of Sensor Node (SN)/Application Node (AN) clusters and at least one Base Station (BS). A physical and logical view of two-tiered WSN is shown in Figure 1 and 2, respectively. In each cluster, there are many SNs and at least one AN. SNs are responsible for all sensing-related activities. Once triggered by an internal timer or an external event, an SN starts to capture and encode live information sent directly to an AN in the same cluster. SNs are small, low cost, and disposable, and can be densely deployed within a cluster. SNs do not communicate with other SNs in the same or other clusters, and usually are independently operated. ANs, on the other hand, have much more responsibilities than SNs. First, an AN receives raw data from all active SNs in the same cluster. It may also instruct SNs to be in sleep, idle, or active state, if some SNs are found to always generate uninterested or duplicated data, thereby allowing these SNs to be reactivated later when some existing active SNs run out of energy. Second, the AN creates an application-specific local view for the whole cluster by exploring correlations among the data sent from SNs. Excessive redundancy in raw data can be alleviated, and the fidelity of captured information should be enhanced. Third, the AN forwards the composite bit-stream toward a BS that generates a comprehensive global-view for the entire WSN. Optionally, ANs can be involved in inter-AN relaying, if such activities are applicable and favorable [12].

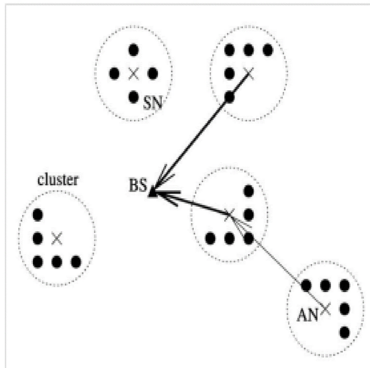

Figure 1. A physical view of two-tiered architecture of Wireless Sensor Networks [12].

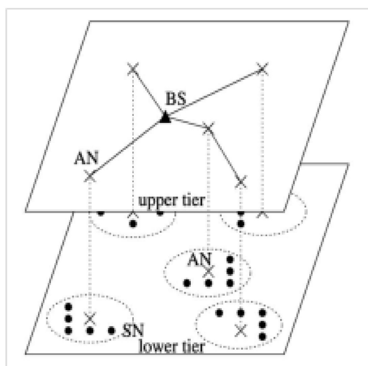

Figure 2. A logical view of two-tiered architecture of Wireless Sensor Networks [12].

In this paper, we will solve *K-of-N* lifetime problem in heterogeneous two-tiered WSNs. The definition of *K-of-N* lifetime problem in heterogeneous two-tiered WSNs which is shown by L_N^K is as follow: suppose given N ANs, where each ANs may own different data transmission rates, initial energies and parameter values, and the network survives as long as there are at least K ANs alive ($1 \leq k \leq N$), or the network fails when *N-K+1* ANs run out of energy, i.e. $L_N^K = min_{N-K+1}\{l_i\}$. Notice that even if some ANs fail, their responsibilities can be taken by nearby ANs, so that the WSN still has the capability to carry on its mission. Normally, K is close to N; otherwise, the deployment of ANs has too much redundancy. To solve *K-of-N* lifetime problem in heterogeneous two-tiered WSNs, we must find a location for BS in two-tiered WSNs so that network lifetime increases.

The remaining parts of this paper are organized as follows: Some related works about finding the location of BS in the two-tiered WSNs is reviewed in Section 2. The GSA is introduced in Section 3. A GSA-based method to find near-optimal location for BS in a two-tiered WSN is proposed in Section 4. Experimental results for demonstrating the performance of the proposed algorithm are described in Section 5. Finally, conclusions are stated in Section 6.

2. Related Work

In the past, many approaches were proposed to efficiently utilize energy in wireless networks. For example, appropriate transmission ways were designed to save energy for multi-hop communication in ad-hoc networks [5,6,8,10,14,17, 19]. Good algorithms for allocation of BSs and SNs were also proposed to reduce power consumption [9, 11, 13, 14, 16].

Pan *et al.* [12] proposed an algorithm to find the optimal locations of BSs in two-tiered wireless homogeneous sensor networks. Let d be the Euclidean distance from an AN to a BS, and r be the data transmission rate. Pan et al. adopted the following formula to calculate the energy consumption per unit time:

$$p(r,d) = r(\alpha_1 + \alpha_2 d^n), \tag{2.1}$$

where α_1 is a distance-independent parameter, α_2 is a distance-dependent parameter, and n is the Euclidean dimension. The energy consumption thus relates to Euclidean distances and data transmission rates. Pan et al. assumed each AN has the same α_1, α_2 and initial energy. For homogenous ANs, they showed that the center of the minimal circle covering of the whole ANs was the optimal BS location (with the maximum lifetime).

Also, Pan *et al.* extended their approach to find the optimal BS location for ANs with different transmission rates by using stacked planes [12]. But if the ANs have different data transmission rates, initial energies and parameter values, their approach can't work.

Hong *et al.* presented solving the *K-of-N* Lifetime Problem in two-tiered WSNs by Particle Swarm Optimization (PSO). Their proposed approach can find near-optimal BS locations in heterogeneous sensor networks, where ANs may own different data transmission rates, initial energies and parameter values [7].

3. Gravitational Search Algorithm

In physics, gravitation is the tendency of agents with object to accelerate towards each other [4]. In the Newtonian gravitational law, each object attracts every other object by a gravitational force. For example, consider a 2-dimensional space which includes objects O_1, O_2, O_3, and O_4. As seen in Figure 3, $F_{1j}(j \in \{2,3,4\})$ is the force acting on O_1 from $O_j(j \in \{2,3,4\})$, and F_1 is the overall force that acts on O_1 from all other objects and generates acceleration a_1 based on Newton's second law .

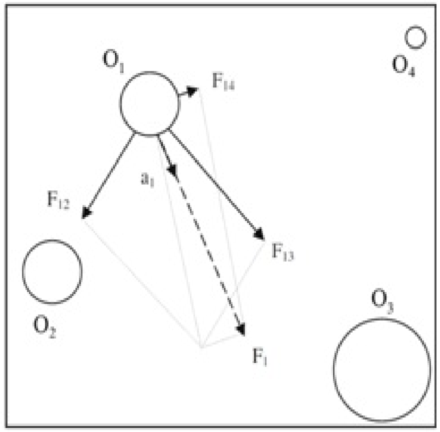

Figure 3. Every object accelerates in the direction of the resultant force that acts on it from the other objects [15].

Gravitational Search Algorithm (GSA) is one of the newest stochastic population based meta-heuristics that has been inspired by Newtonian laws of gravity and motion. In the basic model of the GSA which originally has been designed to solve continuous optimization problem, a set of agents, called objects, are introduced in the n-dimensional search space of the problem to find the optimum solution by simulation of Newtonian laws of gravity and motion. In GSA, the position of each agent demonstrates a candidate solution to the problem, and hence is represented by the vector X_i in the search space of the problem. Agents with a higher performance get a greater gravitational mass, because a heavy object has a large effective attraction radius and hence a great intensity of the attraction. During the lifetime of GSA, each agent successively adjusts its position X_i toward the positions of K_{GSA} best agents of population using gravitational law and laws of motion.

To describe the more details of GSA, consider a system with s agents (swarm size) in which the position of the i-th agent is defined as follows:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n);\ \ i = 1, 2, ..., s, \tag{3.1}$$

where x_i^d presents the position of the i-th agent in the d-th dimension where n is dimension of the search space. Based on [15], gravitational mass of each

Newton's second law states that when a force is applied to an object, the acceleration of this object depends only on the force and gravitational mass of this object. For example, suppose O_i is an object, F_i is a force which acts on O_i, and a_i is the acceleration of O_i. The value of a_i will be obtained as follows: $a_i = \frac{F_i}{gravitational\ mass\ of\ O_i}$

agent is calculated after computing current population's fitness as follows:

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \tag{3.2}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{s} q_j(t)}, \tag{3.3}$$

where $M_i(t)$ and $fit_i(t)$ represent the gravitational mass and the fitness value of the agent i at time t, respectively, and *worst(t)* and *best(t)* are defined as follows for a minimization problem:

$$best(t) = min_{j\in\{1,\ldots,s\}} fit_j(t), \tag{3.4}$$

$$worst(t) = max_{j\in\{1,\ldots,s\}} fit_j(t), \tag{3.5}$$

To compute acceleration of an agent, total forces from a set of K_{GSA} heavier agents (*Kbest* set) that apply on it should be considered based on the law of gravity using Eq. (3.6), which is followed by calculation of agent acceleration using the law of motion by Eq. (3.7):

$$F_i^d(t) = \sum_{j\in Kbest, j\neq i} rand_j G(t) \frac{M_j(t)M_i(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t)), \tag{3.6}$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j\in Kbest, j\neq i} rand_j G(t) \frac{M_j(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t)), \tag{3.7}$$

where:

- $rand_j$ is a uniformly distributed random number in the interval [0,1],
- ε is a very small value used in order to escape from division by zero error whenever the Euclidean distance between two agents i and j is equal to zero,
- $R_{ij}(t)$ is the Euclidean distance between two agents i and j, defined as $\|X_i(t), X_j(t)\|_2$,
- *Kbest* is the set of first K_{GSA} agents with the best fitness value and biggest gravitational mass, which K_{GSA} is a function of time, initialized to $K_{initial}$ value at the beginning and the its value is decreased with time, and
- *G(t)* is the gravitational constant that will take an initial value, $G_{initial}$, and it will be reduced with time toward end value, G_{end}, by Eq. (3.8):

$$G(t) = G(G_{initial}, G_{end}, t). \tag{3.8}$$

Afterwards, next velocity of an agent is calculated as a fraction of its current velocity added to its acceleration by Eq. (3.9). Then, its next position can be calculated using Eq. (3.10):

$$v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t), \tag{3.9}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{3.10}$$

where $rand_i$ is a uniformly distributed random number in the interval [0,1]. The pseudo code of the original GSA is shown in the algorithm (1).

Algorithm (1): Template of Gravitational Search Algorithm.

Randomly generate initial population;
Randomly generate initial velocity;
Evaluate the fitness for each agent;
While stopping criteria is not satisfied **Do**
  Update G, K_{GSA}, and $Kbest$;
  Calculate the acceleration of each agent by Eq. (3.7);
  Calculate the velocity of each agent by Eq. (3.9);
  Update the position of each agent by Eq. (3.10);
  Evaluate the fitness for each agent;
Endwhile
Output: Best solution found.

In GSA, parameters K_{GSA} and G are two main components to balance its intensification (intensively explore areas of the search space with high quality solutions) and diversification (move to unexplored areas of the search space when necessary). It is obvious that each meta-heuristic algorithm, in order to avoid trapping in a local optimum, must use the diversification at the beginning iterations. In GSA, this point is accomplished by assignment of high values to parameters K_{GSA} and G at the beginning. That is, the value of $K_{initial}$ and $G_{initial}$ must be high. It is obvious that the high value for parameter K_{GSA} allows that an agent moves in the search space based on the position of more agents and consequently the diversification of the algorithm is increased. Also, a high value for parameter G increases the mobility of each agent in the search space and hence the diversification of the algorithm is increased. With high value for parameters K_{GSA} and G, we can hope that the good regions of solution space are recognized in premier iterations. Hence, by laps of iterations, the diversification of GSA must fade out and the intensification of it must fade in. This issue is accomplished by reducing the value of parameters K_{GSA} and G by laps of iterations. It is obvious that the low value for parameter K_{GSA} causes that an agent moves in search space based on the position of few agents and consequently the intensification of the algorithm is increased. Also, the low value for parameter G decreases the mobility of each agent in the search space and hence the intensification of the algorithm is increased. Therefore, we can hope that the good regions of the search space are exploited in the ultimate iterations [1, 18].

4. The Proposed Approach: Using the GSA Meta-Heuristic to Find Near-Optimal BS Location

The ANs produced by different manufacturers may own different data transmission rates, initial energies and parameter values. When different kinds of ANs exist in a WSN, it is hard to find the optimal BS location. In this section, a heuristic algorithm based on GSA to find near-optimal location for BS in two-tiered is proposed. In the proposed approach, an initial population of agents is first randomly generated, so that each agent representing the coordinate of a possible BS location. Each agent is also allocated an initial velocity for changing its state.

Let $e_j(0)$ be the initial energy, r_j be the data transmission rate, a_{j1} be the distance-independent parameter, and a_{j2} be the distance-dependent parameter of the j-th AN. The lifetime of an application node AN_j for the i-th agent which is stated by $l_{ij}(t)$ is calculated by the following formula:

$$l_{ij}(t) = \frac{e_j(0)}{r_j(a_{j1} + a_{j2}d_{ij}^n)}, \tag{4.1}$$

where d_{ij}^n is the n-order Euclidian distance from the j-th AN to the i-th agent [13]. In the *K-of-N* lifetime problem, given N ANs, where each ANs may own different data transmission rates, initial energies and parameter values, and the network survives as long as there are at least K ANs alive ($1 \leq K \leq N$), or the network fails when *N-K+1* ANs run out of energy, i.e. $L_N^K = min_{N-K+1}\{l_i\}$. Hence, the fitness function for evaluating each agent can be considered as below:

$$fit_i(t) = \underset{j=1,\dots,m}{min}_{N-K+1}\{l_{ij}(t)\}. \tag{4.2}$$

That is, the fitness of the i-th agent is its (*N-K+1*)-th minimal lifetime among all the ANs. A larger fitness value denotes a better solution quality for the *K-of-N* lifetime problem, meaning the corresponding BS location is better. To achieve near-optimal location for BS, all agents continuously move in the search space of the problem. When the termination conditions are achieved, the best location in the population will be output as the location of the BS. Notice that the termination conditions may be predefined execution time, a fixed number of generations or when the agents have converged to a certain threshold. In the proposed algorithm, termination condition of algorithms is a fixed number of generations. The template of the proposed algorithm is shown in the algorithm (2).

Algorithm (2): Template of proposed algorithm.

Randomly generate a group of s agents, each representing a possible BS location;
Randomly generate initial velocity for each agent;
Randomly generate an initial velocity for each agent;
Evaluate the fitness for each agent;
Set value of K_{GSA} with s (s is the number of population);
$t = 0$;
While stopping criteria is not satisfied **Do**
  Calculate the lifetime $l_{ij}(t)$ of the j-th AN for the i-th agent in step t by Eq.(4.1);
  Calculate the (*N-K+1*)-th minimal lifetime among all the ANs for the i-th agent as its fitness value, $fit_i(t)$, by Eq.(4.2);
  Calculate K_{GSA} and identify *Kbest* set of K_{GSA} best agents;
  Calculate *G(t)*, *best(t)*, *worst(t)*, $M_i(t)$, $F_i(t)$, $a_i(t)$ and $v_i(t+1)$;
  Update the position of each agent, $x_i(t+1)$, by Eq. (3.10);
  $t = t + 1$;
Endwhile
Output: Best solution found.

To better understand how the proposed GSA can be used to find BS location for the *K-of-N* lifetime problem, a small example in a two-dimensional space is given as follows. Suppose there are five ANs in our example and their initial parameters are shown in Table 1. For simplicity, all a_{j1}'s are set at 0 and all a_{j2}'s at 1. Also assume the allowed number K of alive ANs is 3.

Table 1. The initial value of parameters of ANs in our example

AN NO.	Location	Rate	Power
1	(2,3)	4	8000
2	(8,4)	3	10000
3	(3,7)	4	6000
4	(10,8)	5	10000
5	(5,4)	3	8000

Assume 4 agents are used as initial swarm and are randomly located at (2, 7), (6, 1), (5, 9), and (1, 4). For simplicity, initial velocity of each agent is equaled to zero. Then, the lifetime of each AN for an agent is calculated by Eq. (12). Table 2 shows the lifetime of all ANs for all agents. For example, the lifetime of first AN for first agent is calculated as follows:

$$l_{11} = \frac{8000}{4(0+1((2-2)^2+(7-3)^2))} = 125.$$

Table 2. The lifetime of all ANs for all agents.

Agent	**AN**				
	1(2,3)	2(8,4)	3(3,7)	4(10,8)	5(5,4)
1(2,7)	125	74.07	1500	30.77	148.15
2(6, 1)	100	256.41	33.33	30.77	266.67
3(5, 9)	44.44	98.04	187.50	76.92	106.67
4(1, 4)	1000	68.03	115.38	20.62	166.67

As mentioned above, the $(N\text{-}K\text{+}1)$-th minimal lifetime among all the ANs for each agent is computed as its fitness value. For example, (5-3+1)-th minimal lifetime ANs for each agent are 125, 100, 98.04, and 115.38, respectively. Therefore, we have $fit_1 = 125$, $fit_2 = 100$, $fit_3 = 98.04$, and $fit_4 = 115.38$.

5. Experimental Results

In this section, the experiments were made to show the performance of the proposed approach on finding the near-optimal location of BS in two-tiered WSNs. All of them were implemented in C language and were performed on an Intel PC with a 2.0GHz processor and 1GB main memory and the Microsoft Window XP operating system. The simulation was done in a two-dimensional real-number space of 1000m*1000m. The data transmission rate was limited within 1 to 10 and the range of initial energy was limited between 100000000 to 999999999.

In simulation, the number of ANs is equal to 50. Some data of all ANs such as: its own location, data transmission rate and initial energy, were randomly generated based on the above assumptions. Also, the distance-independent parameter (a_{j1}) for each ANs are set at zero, the distance-dependent parameter (a_{j2}) for each ANs was set at one, and the allowed number of alive ANs (i.e. K) was set at 40. As said, K must close to N; otherwise, the deployment of ANs has too much redundancy.

Moreover, for tuning the parameters of GSA, the number of agents is equal to 15 (i.e. $s = 15$), the number of generations equal to 50, $K_{initial}$ is equal to s

Table 3. The lifetime comparison of the proposed approach, the PSO algorithm and the exhaustive grid search

Method	Lifetime
The proposed approach	212.4413
The PSO algorithm [7]	212.4404
The exhaustive grid search (grid size = 1) [7]	212.0781
The exhaustive grid search (grid size = 0.1) [7]	212.4158
The exhaustive grid search (grid size = 0.01) [7]	212.4379

(i.e. the number of agents), $G_{initial}$ is equal to 5, and G_{end} is equal to 1. Also, we will use the linear function to reduce the value of parameters G and K_{GSA} with time.

The found life time for 40-of-50 life time problem by proposed algorithm is shown in Table 3. the lifetime of the proposed approach is obtained as follows: first 10 WSNs have created randomly and run 50 times the proposed approach on each WSN (i.e. the proposed approach runs 50*10=500 times). We assume network lifetime is equal to average of obtained lifetime from running the 500 times the proposed approach. In Table 3, the lifetime of a PSO algorithm and an exhaustive search with different grid sizes are also shown. One can see from Table 3 that the lifetime obtained by our proposed approach is not worse than PSO algorithm and exhaustive grid search within a certain precision. The lifetime by the PSO algorithm was 212.4404 and also by the exhaustive search for the grid size was set at 1, 0.1 and 0.01 was 212.0781, 212.4158 and 212.4379, respectively. Therefore, our proposed approach can find better BS location than the PSO algorithm and the exhaustive search when grid size equal 1, 0.1 and 0.01.

In Table 4, the execution time by the three mentioned approaches is shown. It can be seen from Table 4 that the execution time of the exhaustive grid search increased along with the decrease of grid sizes. It was very consistent with the processing way of the exhaustive grid search. Besides, the exhaustive grid search spent much more execution time than our proposed approach and PSO algorithm, especially when the grid size was small. The advantage of the proposed approach and PSO algorithm for solving the problem lies in that it can reduce the computation time and keep the same good quality. However,

Table 4. Comparison of execution time by the three approaches

Method	Time (sec.)
The proposed approach	0.17
The PSO algorithm [7]	0.06
The exhaustive grid search (grid size = 1) [7]	36.563
The exhaustive grid search (grid size = 0.1) [7]	2480.718
The exhaustive grid search (grid size = 0.01) [7]	170871.8558

computation time of our approach is somewhat more than the PSO algorithm because the GSA meta-heuristic has more computational steps than PSO algorithm in each iteration [15].

6. Conclusion

In Wireless Sensor Networks, minimizing power consumption to prolong network lifetime is very important. In this paper, a two-tiered Wireless Sensor Networks has been considered and an algorithm based on Gravitational Search Algorithm (GSA) has been proposed for general power-consumption constraints. The proposed approach can find near-optimal BS location in heterogeneous sensor networks, where ANs may own different data transmission rates, initial energies and parameter values. It is very easy to model such a problem by the proposed algorithm based on GSA. Experiments show the performance of the proposed approach. From the experimental results, it can be easily concluded that the proposed approach finds the better location when compared to the PSO algorithm and the exhaustive search (with investigated grid size, i.e. 1, 0.1 and 0.01) and also converges very fast when compared to the exhaustive search (with investigated grid sizes, i.e. 1, 0.1 and 0.01).

Acknowledgments

We would like to thank the referees for their constructive comments and suggestions.

References

1. G. Aslania, S. H. Momeni-Masuleh, A. Malekb, F. Ghorbanib, Bank efficiency evaluation using a neural network-DEA method, *Iranian Journal of Mathematical Sciences and Informatics*, **4**(2), (2009), 33-48.

2. J. Chou, D. Petrovis, K. Ramchandran, *A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks*, The 22nd IEEE Conference on Computer Communications (INFOCOM), p. 1054-1062, San Francisco, USA, March, 2003.
3. D. Estrin et al., Embedded, Everywhere: *A Research Agenda for Networked Systems of Embedded Computers*, Nat Research Council Report, 2001.
4. D. Halliday, R. Resnick, J. Walker, *Fundamentals of Physics*, John Wiley and Sons, 1993.
5. W. Heinzelman, A. Chandrakasan, H. Balakrishnan, *Energy-efficient communication protocols for wireless microsensor networks*, The Hawaiian International Conference on Systems Science, 2000.
6. W. Heinzelman, J. Kulik, H. Balakrishnan, *Adaptive protocols for information dissemination in wireless sensor networks*, The 5th ACM International Conference on Mobile Computing and Networking, p. 174-185, 1999.
7. T. P. Hong, G. N. Shiu, Solving the K-of-N Lifetime Problem by PSO, *International Journal of Engineering, Science and Technology*, **1**(1), (2009), 136-147.
8. C. Intanagonwiwat, R. Govindan, D. Estrin, *Directed diffusion: a scalable and robust communication paradigm for sensor networks*, The ACM International Conference on Mobile Computing and Networking, 2000.
9. V. Kawadia, P. R. Kumar, *Power control and clustering in ad hoc networks*, The 22nd IEEE Conference on Computer Communications (INFOCOM), 2003.
10. S. Lee, W. Su, M. Gerla, Wireless ad hoc multicast routing with mobility prediction, *Mobile Networks and Applications*, **6**(4), (2001), 351-360.
11. N. Li, J. C. Hou, L. Sha, *Design and analysis of an mst-based topology control algorithm*, The 22nd IEEE Conference on Computer Communications (INFOCOM), 2003.
12. J. Pan, L. Cai, Y. T. Hou, Y. Shi, S. X. Shen, Optimal base-station locations in two-tiered wireless sensor networks, *IEEE Transactions on Mobile Computing*, **4**(5), (2005), 458-473.
13. J. Pan, Y. Hou, L. Cai, Y. Shi, X. Shen, *Topology control for wireless sensor networks*, The 9th ACM International Conference on Mobile Computing and Networking, p. 286-299, 2003.
14. R. Ramanathan, and R. Hain, *Topology control of multihop wireless networks using transmit power adjustment*, The 19th IEEE Conference on Computer Communications (INFOCOM), 2000.
15. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A Gravitational Search Algorithm, *Information Sciences*, **179**(13), (2009), 2232-2248.
16. V. Rodoplu, and T. H. Meng, Minimum energy mobile wireless networks, *IEEE Journal on Selected Areas in Communications*, **17**(8), (1999), 1333-1344.
17. D. Tian, N. Georganas, *Energy efficient Routing with guaranteed delivery in wireless sensor networks*, The IEEE Wireless Communication & Networking Conference, 2003.
18. M. Yarahmadi, S. M. Karbassi, Design of robust controller by neuro-fuzzy system in a prescribed region via state feedback, *Iranian Journal of Mathematical Sciences and Informatics*, **4**(1), (2009), 1-16.
19. F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, *A two-tier data dissemination model for large scale wireless sensor networks*, The ACM International Conference on Mobile Computing and Networking, 2002.