IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 3-Summer 2014 (7-13)

An Improved Parallel Genetic Algorithm for Optimal Sensor Placement of Wireless Sensor Networks

Sara Kolangari
Department of Computer Engineering
Islamic Azad University
Science and Research Branch
Tehran, Iran
sara.kolangari@gmail.com

Mohammad Teshnehlab
Department of Control and Systems Engineering
K.N. Toosi University of Technology
Tehran, Iran
teshnehlab@eetd.kntu.ac.ir

Received: December 9, 2013 - Accepted: January 27, 2014

Abstract— **The wireless sensor network has recently become an intensive research focus due to its potential applications many years. Sensor placement is one of the most important issues in wireless sensor networks. An efficient placement scheme can enhance the quality of monitoring in wireless sensor networks by increasing the coverage rate of interested area. This paper presents an efficient method based on parallel genetic algorithms to solve a sensor placement optimization problem. We modify the general master-slave parallel genetic algorithm to improve the convergence rate of this optimization method. The results indicate the effectiveness of the proposed method in comparison with genetic algorithm, general parallel genetic algorithm, and some well-known evolutionary algorithms.**

Keywords- Sensor placement, Wireless sensor networks, and Parallel genetic algorithm

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a network of simple devices called sensor nodes that collaborate in order to monitor some conditions in the physical environment. These sensors are able to communicate with each other to collaboratively detect objects, collect information, and transmit messages [1]. The sensors fulfill two fundamental functions: sensing and communicating. The sensing can be of different types (seismic, acoustic, chemical, optical, etc.), and the communication is performed wirelessly [2]. Sensor networks have become an important technology, especially for environmental monitoring, military applications, disaster management, etc [3, 4]. Article [5] presents numerous applications of WSNs. Sensor nodes are designed for each application with specific facilities, but usually these sensor nodes are tiny and low-cost devices. Due to their limited size, sensors do not have very powerful processor and are limited in computational power and memory [5]. One of the important tasks of the sensors in a WSN, is monitoring the area completely. The success of the WSN is dependent on the sensors' positions in the area of interested. It is a challenge of the WSNs where is called placement problem. The placement of sensors must ensure that there are no undetected points (holes), i.e., every point in the area of interested should be monitored by at least one sensor [6]. The design space of the WSN layout optimization is highly non-linear [7]. The use of optimization techniques in the placement of WSN is very hot topic for researchers, and some placement algorithms have been developed for the placement problem in WSNs [8, 9]. The placement problem is an NP-hard problem, and it is difficult to obtain the precision solution by traditional methods [10]. Evolutionary algorithms (EAs) can be very helpful in the process of designing and planning the placement of sensor networks [11].

Genetic Algorithms (GAs) are the most important class of EAs that use some biologically inspired techniques to approximate optimal solutions for optimization problems. Several scholars have successfully implemented GAs in sensor network placement. Nabi et al. proposed general guidelines for using GAs in design space exploration and configuration of WSNs in [12]. They used a multi-objective version of GA for the optimal placement of sensors in a grid area with obstacles and preferences to minimize the number of sensors. Romoozi et al., in [13], illustrates A Positioning Method in Wireless Sensor Networks Using Genetic Algorithms. They investigate intelligent techniques for node positioning to reduce energy consumption with coverage preserved in wireless sensor network. In [7], a multi-objective GA is used to optimize the layout of the WSNs. Also, that paper considered the total sensor coverage and the lifetime of the network. As mentioned above, current works on WSN mainly focuses to use the simple GA or multi-objective GA. This paper uses Parallel Genetic Algorithm (PGA) to optimize the sensor placement of WSNs. Also, the new PGA based method is presented by modifying the general master-slave PGA which is named "Improved PGA" (IPGA). Then, the main contributions of this paper are present a sensor placement method via mentioned IPGA for the first time. This proposed method aims to generate much more variety of solutions in order to achieve better results in optimization process. We consider the maximizing of coverage rate as main objective.

The rest of this paper is organized as follow: section 2 reviews parallel GA, section 3 contains the modeling, section 4 presents our simulation results and the final section is the conclusion.

II. Parallel Genetic Algorithms

The basic idea behind most parallel programs is to divide a large problem into smaller tasks [14]. Tasks are solved simultaneously on multiple processors. The parallel architecture of GA called parallel genetic algorithm, is commonly used to improve the convergence speed of GA [14]. One of the specifications of PGAs is increasing the diversity of candidate solutions. This characteristic causes to improve the convergence rate of the algorithm.

There are three main types of PGAs: (1) global single-population master-slave, (2) single-population fine-grained, and multiple-population coarse-grained GAs [15]. In a master-slave PGA there is a single population (just as in a simple GA), but the evaluation of fitness is distributed among several processor (see Fig. 1 (a)). Fine-grain PGAs are suited for massively parallel computers and consist of one spatially-structured population (see Fig. 1 (b)). Multiple-population PGAs are more sophisticated, as they consist on several subpopulations which exchange individuals. In a multiple-population PGA each process is a simple GA, and there is (infrequent) communication between the populations (see Fig. 1 (c)) [15].

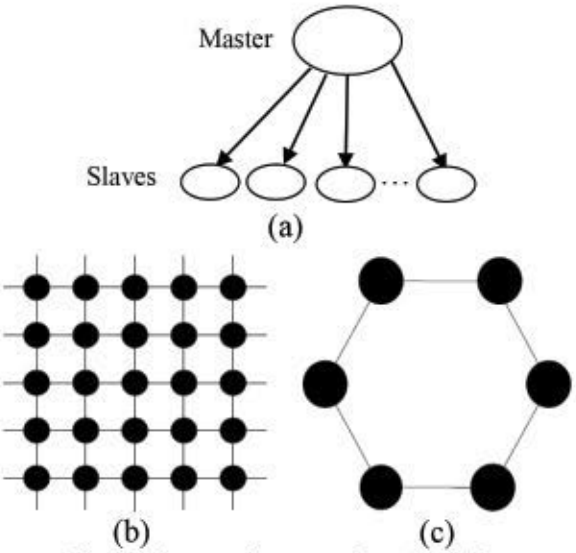


Fig. 1. Three main types of parallel GAs
(a) A schematic of a master-slave parallel GA,
(b) A schematic of a fine-grained parallel GA,
(c) A schematic of a multiple-population parallel GA [15]

III. Modeling

In this section, assumptions and coverage calculation for the sensor network are illustrated. After that, Improved PGA (IPGA) and encoding the WSN space to PGA space are presented. Note that we use the optimization algorithm to maximize the coverage rate of the network.

A. *Assumptions and coverage calculation*

In our work, the sensing field is considered to be a flat square grid, detection radius of sensors is identical (r), and all sensor nodes are static.

As mentioned above, the sensor field is a two-dimensional grid. Coverage rate of the WSN is calculated by Eq. (1) [16].

$$CR = \frac{\bigcup C_i}{A} \quad , i \in S \tag{1}$$

Where C_i is the coverage of a sensor i, S is the set of the nodes, and A is the total size of the area of interest.

There are two sensor detection models for finding the effective coverage of WSNs: the binary detection model and the probabilistic detection model. In this paper, we used the binary detection model which assumes that there is no uncertainty [17]. Assuming that, there are k sensors in the random placement stage, each sensor has the same detection range r, sensor i is positioned at (x_i, y_i). The calculation of C_i is presented in Eq. (2).

$$C_i = \begin{cases} 1 & \text{if } \sqrt{(x-x_i)^2+(y-y_i)^2} \le r \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For any point P at (x, y), in the area, Euclidean distance between the location of sensor i and P is calculated in Eq. (2).

As mentioned above, C_i is the generated coverage by sensor i in the surrounding environment. Note that, in Eq. (1), the reason why a union operation ($\bigcup$) is used rather than a plus operation, is that every grid point is calculated only once. To prevent the overlap, the following steps must be performed. While a sensor is placed in a grid point, every points of area which are detected by this sensor are listed in an array. After that, for next sensor placement, the points which are listed in the array are not calculated again.

B. *Encoding the WSN space to PGA space*

For sensor placement in the WSN area, on the one hand, we have a given WSN with a certain specification and on the other hand, we have an optimization method (here, parallel GA) with a specific selection method that has itself some controllable parameters [12]. The area of interest considered in this paper is composed of a two-dimensional grid field. This means the area is a square region. There are l rows and l columns with total $m = l*l$ grids in the field. Fig. 2 shows our grid sensing area. A sensor S_i ($1 \le i \le k$ and k is number of nodes) can be placed at a grid which is represented by a black point in this figure.

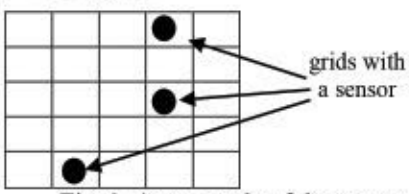


Fig. 2. An example of the sensor field

The sensor nodes can be deployed by various forms in the WSN which are called WSN configurations. For encoding the WSN space to the PGA space, we indicate each special configuration of sensors by one chromosome of GA. The chromosomes in GA are candidate solutions of the problem. In our model each chromosome (solution) represents by one array which k items. k is the number of sensor nodes in area. Fig. 3 shows a solution array. Each solution represents an array which has $2k$ items. Items of the solution array are (x, y) positions of the sensors in the network [17].

1	2	3	4	…	2k-1	2k
x_1	y_1	x_2	y_2	…	x_k	y_k

Fig. 3. Solution array

C. *Improved PGA for sensor placement*

As mentioned above, global parallel GAs usually are implemented as master-slave programs, where the master stores the population and the slaves evaluate the fitness. The most common operation that is parallelized, is the evaluation of the individuals, because the fitness of an individual is independent from the rest of population, and there is no need communicate during this phase. The evaluation of individuals is parallelized by assigning a fraction of the population to each of the processors. Communication occurs only as each slave receives its subset of individuals to evaluate and when the slaves return the fitness values [15].

In this paper, we modify the mentioned general master-slave PGA. This change is to perform two genetic operations (i.e. crossover and mutation) in the slaves. This modifying causes to increase the diversity in the solutions. Due to this, the convergence rate of PGA improves. We call this PGA, Improved PGA (IPGA). The system of general PGA and proposed IPGA and comparison of these PGAs are shown in Fig. 4 and Fig. 5.

In Fig. 4 and Fig. 5, the number of slaves is assumed two. We can see four configurations of WSN area in both Fig. 4 and Fig. 5. Each configuration is represented by a solution array or a chromosome (see Section 3.2). In these figures, each chromosome contains P_i^j, which is the position of i th sensor in a configuration of the WSN (j is the label of the chromosome). Consider the curved arrows. These show the communication between the master and the slaves. The chromosomes are sent to slave processors to be evaluated and return to the master at the end of every generation. In general PGA (Fig. 4) the "master" processor would be responsible for explicitly sending the individuals to the other processors (the "slaves") for evaluation, collecting the results, and applying the genetic operators to produce the next generation, while in IPGA (Fig. 5), the slaves evaluate the fitness of the individuals and also execute the genetic operations (mutation and crossover). Performing the mutation and crossover operations in the slaves in IPGA causes to improve the diversity of population in each step of the algorithm.

Note that in IPGA we require the cost for performing the mutation and crossover operations in each slave. However, the placement problem in the WSNs is an off-line problem. It means that the excess cost can be ignored.

The Steps of the IPGA for placement problem are as follow.

1. Initialize the parameters: detection radius r, size of interested area A, number of sensors k, maximum number of generations *MaxItt*, and GA initial parameters (mutation and crossover probability, population size).
2. Deploy k sensors randomly in the area and produce a number of WSN configurations.
3. Encode the configurations to the GA solutions (chromosomes).
4. $itt = 0$.
5. Repeat.
6. Send a fraction of population to each of the slaves.
7. Execute the crossover and mutation operators and produce new solutions.
8. Evaluate the fitness value for each solution (chromosome).

9. Collect the results (new population) from slaves by master.
10. Apply a genetic selection operator on new population in the master.
11. Memorize the best solution achieved thus far.
12. $itt = itt + 1$.
13. Until $itt = MaxItt$
14. Decode the final best solution of IPGA to the WSN configuration space.

In steps 7 and 10, the main GA operators are mentioned. The crossover is performed between two individuals (or chromosomes). Every solution mates with another to produce two Children. The crossover point is chosen randomly. Another GA operator is the mutation. By mutation solutions are randomly altered. In mutation, the solution may change entirely from the previous solution. The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. Selection is also one of the main GA operators. Selection is the stage of a GA in which individuals are chosen from a population for later breeding (recombination or crossover). More details about GA and its operators can be found in [19].

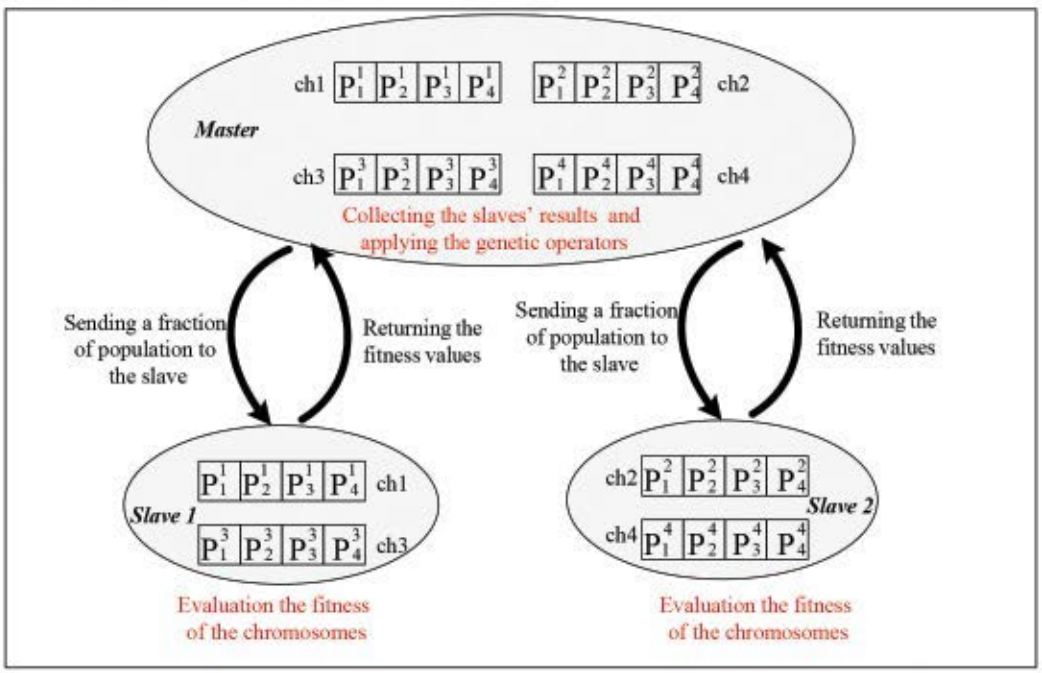


Fig. 4. The illustration of general master-slave PGA for sensor placement

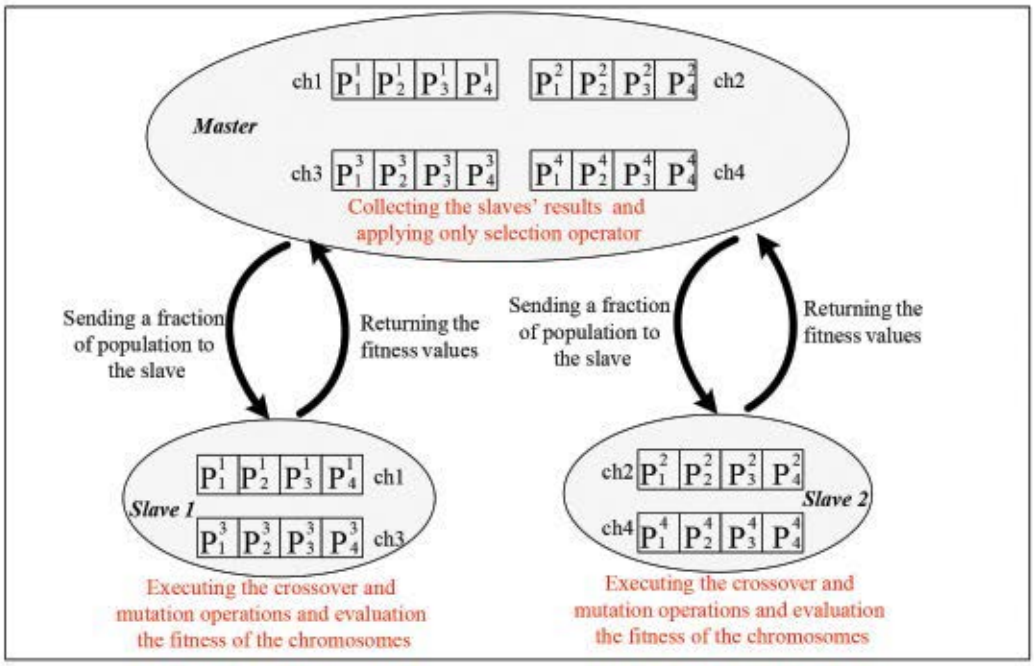


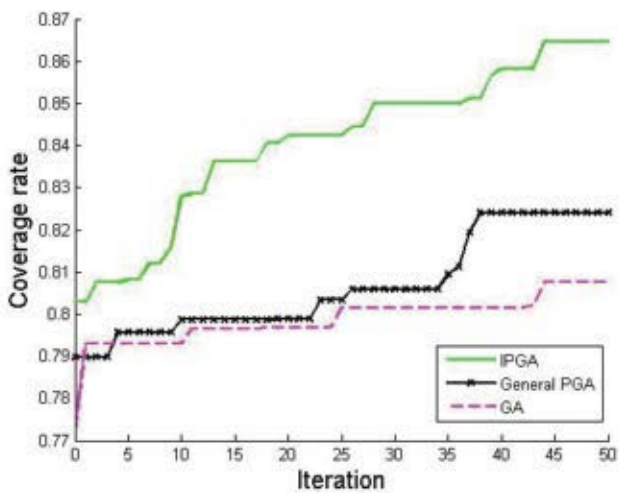
Fig. 5. The illustration of Improved PGA for sensor placement

IV. SIMULATION RESULTS

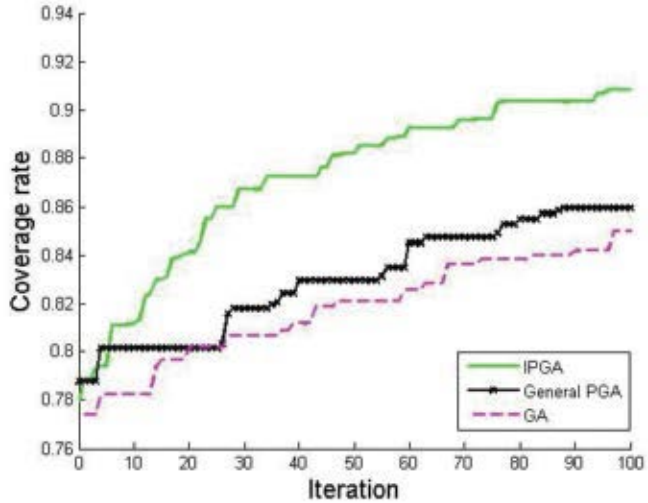
In this section, we study performance of IPGA to optimize the sensor placement by simulation. Also, the performance of the IPGA on placement problem is compared with the results of the GA and general PGA. Moreover, to compare the performance among proposed method (IPGA) and other evolutionary algorithms, we perform the same simulations based on Particle swarm optimization (PSO), Differential Evolutionary (DE), and Evolutionary Strategy (ES).

As mentioned above, in this work the WSN includes static sensors. Detection radius of each sensor (r) is 7 meters, size of the area which is a square region A is 100 m by 100 m (10000 square meters). This wireless sensor network including 100 sensor nodes.

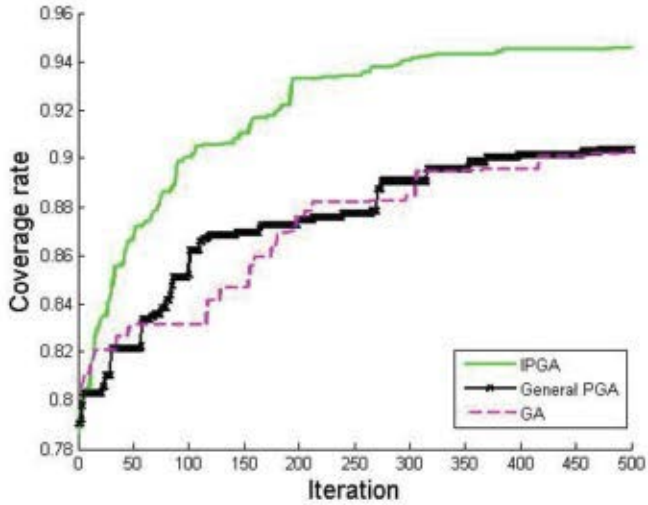
The GA control parameters are set as follow: the crossover probability is 1, the mutation probability is 0.01, and the population size is 12. To allow a fair comparison of running times, all the experiments were performed on a PC with a core 2 Dou processor 266 GHZ and 2.99 GB RAM. Our implementation was compiled using MATLAB R2008a running under windows XP SP3. No commercial GA tools or parallel GA tools where used in the experiments. Each problem has been tested 10 times with random initial populations, but only the best results are shown in the following example. To observe the development of the best solutions for the algorithms through the iterations, refer to Fig. 6. These plots are focused on the comparison among IPGA, PGA, and GA.



(A)



(B)



(C)

Fig. 6. The coverage rate in GA, PGA, and IPGA. Size of the area is 10000 square meters. Number of iterations is 50, 100 and 500 in a, b, and c.

Fig. 6 including the developing graphics of the populations through the different iterations for GA, PGA, and IPGA, demonstrate that the IPGA finds better placement of sensors than the other algorithms. In Fig. 6, the convergences of three algorithms are shown by coverage rate for the iterations: iteration number 50, iteration number 100, and iteration number 500. Comparing the curves in this figure shows the quality of IPGA results for sensor placement problem, especially in high iterations. The more vibrations in IPGA plots than other plots reveal the much more diversity in IPGA solutions. IPGA increases this diversity because of performing the genetic operation in each slave (see section 3.3). This diversity raises the convergence rate of the IPGA.

As mentioned above, we perform the same simulations based on PSO, DE, and ES. The results are recorded in Table 1. In this table, the numbers in columns indicate the coverage rate of area. These results have produced by IPGA and other EAs on different iterations.

As seen from Table 1, the IPGA is more successful than the PSO, DE, and ES for the placement problem of WSNs. In addition, the simulation results show that the coverage rate of area has improved through the increasing of iterations.

V. Conclusion

In this study, an Improved Parallel Genetic Algorithm (IPGA) was introduced to solve the sensor placement problem of WSNs. In fact some modifications were introduced on general master-slave parallel genetic algorithm for sensor placement. We investigate the performance of our method in various conditions. In the simulations coverage rate of interested area is considered. Experiment results illustrate the IPGA can achieve much better performance than general master-slave parallel genetic algorithm and genetic algorithm. Also, the IPGA has compared with PSO, DE, and ES. The results indicate that IPGA has higher performance than mentioned methods.

Table 1: Node placement results on different iterations by IPGA and other EAs

	GA	PGA	IPGA	PSO	DE	ES
iteration=50	0.8116	0.8197	**0.846**	0.7622	0.7917	0.8175
iteration=100	0.8135	0.8235	**0.8626**	0.7753	0.7895	0.8373
iteration=500	0.8923	0.8955	**0.9194**	0.7771	0.8164	0.8168

REFERENCES

[1] T. Vincent, C. King-Yip, L. King-Shan, Improving Localization in Wireless Sensor Networks with An Evolutionary Algorithm, Consumer Communications and Networking IEEE Conference 2006.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] V. Kumar, P. Khanna , S. Bisht, Adaptive PSO based Algorithm for Optimal WSN Deployment in 3 Dimensional Terrains, National Conference on Advancement of Technologies - Information System & Computer Networks , 2012.

[3] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, Connecting the physical world with pervasive networks. In IEEE Pervasive Computing, pages 59-69, 2002.

[4] S. Goyal, A Survey on Deployment Methods in Wireless Sensor Networks, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, 2013.

[5] R. Kulkarni, A. Forster, G. Venayagamoorthy, Computational Intelligence in Wireless Sensor Networks: A Survey, IEEE Communications Surveys and Tutorials, 2011, Vol. 13, No 1, 68-96.

[6] X. Yong, Y. Xin, A GA Approach to the Optimal Placement of Sensors in Wireless sensor Networks with Obstacles and Preferences, Consumer Communications and Networking Conference, 3rd IEEE, 2006.

[7] Da. B. Jourdan, O. L. de Weck, Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm, Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th (Volume:5)

[8] S.S. Dhillon, K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks", Wireless Communications and Networking, Vol. 3, pp. 1609-1614, 2003.

[9] G. Molina, E. Alba, "Wireless sensor network deployment using a memetic simulated annealing", International Symposium on Applications and the Internet, pp. 237-240, 2008.

[10] G. Nan M. Li, Evolutionary Based Approaches in Wireless Sensor Networks: A Survey, Fourth International Conference on Natural Computation, IEEE, 2008.

[11] R. Kulkarni, A. Forster, G. Venayagamoorthy, Computational Intelligence in Wireless Sensor Networks: A Survey, IEEE Communications Surveys and Tutorials, 2011, Vol. 13, No 1, 68-96.

[12] M. Nabi, M. Blagojevic, T. Basten, M. Geilen, T. Hendriks, Configuring Multi-Objective Evolutionary Algorithms for Design-Space Exploration of Wireless Sensor Networks, In 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks, pages 111-119. 2009.

[13] M. Romoozi, H. Ebrahimpour, A Positioning Method in Wireless Sensor Networks Using Genetic Algorithms, International Conference on Medical Physics and Biomedical Engineering,Volume 33, 2012, Pages 1042-1049.

[14] E. Alba, M. Tomassini, Parallelism and Evolutionary Algorithms, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 5, OCTOBER 2002.

[15] DS. Knysh, VM. Kureichik, Parallel genetic algorithms: a survey and problem state of the art, Journal of Computer and Systems Sciences, Springer, 2010.

[16] K. Chakrabarty, S.S. Iyengar, H. Qi, E, Cho, Grid coverage for surveillance and target location in distributed sensor networks. IEEE Trans. Comput., 51, 1448–1453, 2002.

[17] C. Ozturk, D. Karaboga, B. Gorkemli, Artificial Bee Colony Algorithm for Dynamic Deployment of Wireless Sensor Networks, International Symposium on Innovations in Intelligent Systems and Applications, 2010.

[18] A. p. Bhondekar, R. Vig, M.L. Singla, C. Ghanshyam, P. Kapur, Genetic Algorithm Based Node Placement Methodology For Wireless Sensor Networks, International MultiConference of Engineers and Computer Scientists, Hong Kong, 2009.

[19] L. Chambers, GENETIC ALGORITHMS, second edition, CHAPMAN & HALL/CRC , 2001, pp. 79–82.

Sara Kolangari was born in Arak, Iran, in 1984. She received the B.Sc. degree in computer engineering from Azad University, Arak, Iran, in 2006 and the M.Sc. degree in computer engineering, specializing in Artificial Intelligence, from Azad University, Science and Research Branch, Tehran, Iran, in 2013.She worked in a software engineering company as an application systems programmer and software testing expert for 6 years. Her current research interests include evolutionary algorithms, swarm intelligence, neural networks, pattern recognition, image processing, and wireless sensor networks.

Mohammad Teshnehlab received the B.Sc. degree in electrical engineering from Stony Brook University, Stony Brook, NY, in 1980, the M.Sc. degree in electrical engineering from Oita University, Oita, Japan, in 1991, and the Ph.D. degree in computational intelligence from Saga University, Saga, Japan, in 1994.He is currently an Full Professor of control systems with the Department of Electrical & Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran. He is the author or a coauthor of more than 130 papers that have appeared in various journals and conference proceedings. His main research interests include neural networks, fuzzy systems, evolutionary algorithms, swarm intelligence, fuzzy-neural networks, pattern recognition, metaheuristic algorithms and intelligent identification, prediction and control, and interval soft computing.

IJICTR

This Page intentionally left blank.