2010 International Conference on Computational Intelligence and Communication Networks

Modified Discrete Binary PSO based Sensor Placement in WSN Networks

Shirin Khezri [1] , Karim Faez [2] , Amjad Osmani [3]

[1,2,3] *Department of Computer Engineering and Information Technology*
[1] Islamic Azad University, Qazvin Branch, Iran
[2] Amirkabir University of Technology, Tehran, Iran
[3] Islamic Azad University, Saghez Branch, Iran
s.khezri @ qiau.ac.ir [1] , kfaez@aut.ac.ir [2], a.osmani @ {iausaghez.ac.ir, qiau.ac.ir} [3]

Abstract

In this article we applied modified binary particle swarm optimization algorithm for solving the sensor placement in distributed sensor networks. PSO is a real value algorithm, and the discrete PSO is proposed to be adapted to discrete binary space. In the distributed sensor networks, the sensor placement is NP-complete for arbitrary sensor fields and it is one of the most important issues in the research fields, so the proposed algorithms are going to solve this problem by considering two factors: one is the complete coverage and the second one is the minimum costs. The proposed method on sensors surrounding is examined in different area. The results not only confirmed the successes of using new method in sensor replacement, but also they showed that the new method performs more efficiently compared to the Simulated Annealing Algorithm.

Key words: Modified binary PSO, Distributed Sensor Network, Sensor Placement

1- Introduction

In the distributed sensor networks, the issue of sensor placement is of paramount importance in researches. A sensor network can arrange in two ways, one as a random placement and the second as a grid-based placement. Once the surrounding is unknown the random placement is the only option and the sensors may disintegrated everywhere but when the features of the network were known before, then the sensor placement could be done with great scrutiny and we could guarantee the quality of providing services along with satisfying the limitations. The strategy of sensor placement depends on the application of the distributed sensor network (DNS). In this article the focus is on the gird-based placement. And we applied the modified binary PSO algorithm for solving these NP-complete problems [6].

Considering the existence of many networks with high velocity and computational capabilities of these sensor networks, we can say that they have different applications for example in aviation, military, medical, robot, air forecasting, security and anti terrorism applications and also we can use them in very important infrastructures like power plants ,environmental and natural resource monitoring, and military applications like communication systems, commanding, reconnaissance patrols, looking –out etc [7][8].

In [9] and [10], they present a resource-bounded optimization framework for sensor resource management under the constraints of sufficient grid coverage of the sensor field. In [11], they formulate the sensor placement problem in terms of cost minimization under coverage constraints. In [12] Node placement in heterogeneous WSN is formulated using a generalized node placement optimization problem to minimize the network cost with lifetime constraint, and connectivity. In [13] they formulate and solve the sensor placement problem for efficient target localization in a sensor network, they develop a

978-0-7695-4254-6/10 $26.00 © 2010 IEEE
DOI 10.1109/CICN.2010.49

mathematical framework for the localization of the missile using multiple sensors based on Cramer-Rao Lower Bound (CRLB) analysis. In [14] they present the practical problem of optimally placing the multiple PTZ cameras to ensure maximum coverage of user defined priority areas with optimum values of parameters like pan, tilt, zoom and the locations of the cameras. Moreover in [15] a heuristic algorithm is proposed based on Simulation Annealing Algorithm to solve this problem considering the coverage and cost limitations.
The rest of the paper is organized as follows: in section 2, we state the sensor placement problem and then present the mathematical model in section 3. Section 4, proposes an algorithm. The performance evaluations are in Section 5. Section 6 concludes the paper.

2- Defining the problem

The sensor network based on gird-based could be considered as a two or three dimensional network [9]. A set of sensors are settled on different points of gird points in order to monitor the sensor field. In this section we defined a power vector for each point of the field to show whether these sensors could cover that point on the field or not, for which the number of components are as many as the number of sensors available.
Now if the Euclidean distance between each grid point and the corresponding sensor is less than the coverage radius of the sensor ($d<r$), so the coverage is assumed to be full (1), and it becomes a parallel component of that sensor on the power vector, Otherwise, the coverage is ineffective and the parallel components equaled to (0). If each point on the gird point in a sensor field can be covered by at least one sensor so that the sum of the vector components of that field equals to one, the field is called completely covered. In Fig.1, a complete and discriminated sensor field of 4*4 with radius =1 is illustrated, that a target can be detected at any place in the field. In figure 1, for example the power vector for point 7 equals to (0, 1, 0, 0) which is calculated based on the sensors of 2, 8, 9 and 15. When a target appears at the grid point 7, the backend will receive reports from sensor 8 [15]. In a completely covered sensor field, when each grid point is identified by a unique power vector, the sensor field is said to be completely discriminated, as shown in Fig.1.

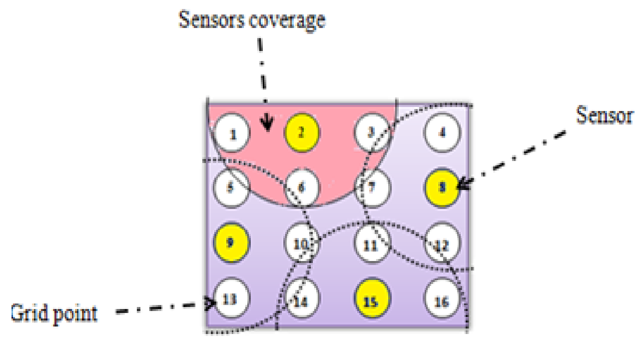

Fig.1- A complete covered and discriminated sensor field with radius =1

3- The proposed algorithm

The PSO algorithm is an optimization technique based on statistical rules which were proposed by Eberhart and Kennedy in 1995 and the proposed algorithm inspired by the social behavior of birds and fishes in searching for food [1]. Suppose a group of birds are searching for food in a place randomly and food is available in one part of searching area and the birds have no information about the place where the food is available and they only know their distance to the food source. The adopted strategy by birds is that they follow the bird which has minimum distance to the food source.
In PSO algorithm, each answer to the problem is considered as a bird in the search space which is called a particle. Each particle has its own fitness determined by the fitness function. A bird which is close to food source has a better fitness. This algorithm has a continuous nature and it proved its performance in different applications [2][3][4]. There are many subjects which have

discrete nature and because there are many problems which have a discrete nature and also because many of both discrete and continuous problems can be solved in a discrete space so there is a need to use the binary PSO algorithm. A copy of binary algorithm was proposed by its designers in 1997 which unfortunately lacks enough convergence [5].

PSO is initialized with a group of random particles (solutions) and then searches for optimal solutions by updating the position of particles. Each particle is distinguished by an N_d-dimensional vector (the number of points of sensor field) along with two value of X_{id} and V_{id} Where V_{id}= (V_{id1}...V_{idn}) is the velocity of particle *id*, X_{id}= (X_{id1}...X_{idn}) is the current position (solution).

In binary PSO model, the position of each particle is defined by two values of (0 and 1), it means that each particle moves in a limited space of zero and one and V_{id} explains the probability of defining the value of one for X_{id}. In each iteration, the velocity and the position of each particle *id* is updated using two quantities: the Personal Best solution obtained by particle *id* ($p_$best) and the Global Best solution ($g_$best) obtained by the group of particles. After finding these two quantities, particle *id* updates its velocity according to equation (1). Since V_{id} in binary algorithm is defined as probability function so it should be limited in the range (0, 1). So we can update the position of a particle by applying equation (2). In equation (2) the new position component to be exchanged with a probability value which is obtained by applying a modified sigmoid transformation $S'(v_{id})$ to the velocity component according to equations (3). The v_{id} in high value indicates that particle's position is unfit, so it causes the value of x_{id} to be changed from 0 to 1 or vice versa, and a low value for x_{id} decreases the probability of changes in the value of x_{id}. Finally if the value of v_{id} is zero, the value of x_{id} is unchanged according to equations (2).

$$V_{id}^{d}(t+1) = w(t) * V_{id}^{d}(t) + c_1 * rand\left(p\_best_{id}^{d} - x_{id}^{d}\right) + c_2 * rand\left(g\_best_{id}^{d} - x_{id}^{d}\right), \quad d \in 1, \dots, N_d \tag{1}$$

In equation (1) *Rand* is a random number in the range (0, 1), c_1 and c_2 are learning coefficients. Usually *c1* is equal to *c2,* and they are in the range (1, 2); the inertia factor *w* usually is a number in the range (0, 1). Also the final value for velocity of each particle is limited to a span to avoid the divergence of each algorithm: $v_{id} \in [-v_{max}, v_{max}]$. And finally the only condition to end an algorithm we need algorithm convergence or finishing it after several repetitions.

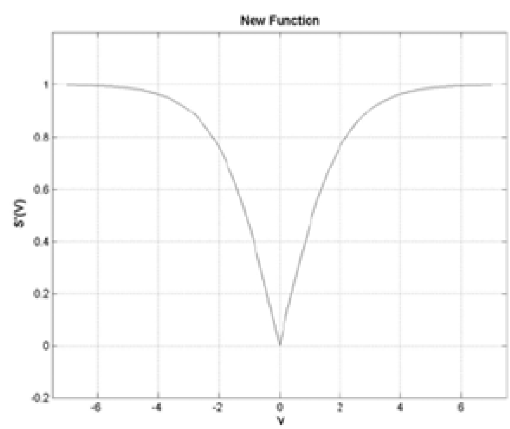

Fig.2- Hyperbolic function $S'(v_{id})$

Fitness function is sum of ones in particle here, which indicate cost of sensors that used for coverage the field sensor completely.

$$\begin{aligned} &if\ rand < S'\left(v_{id}(t+1)\right)\ then \\ &\quad x_{id}(t+1) = exchange\left(x_{id}(t)\right) \\ &\quad else\ \ x_{id}(t+1) = x_{id}(t) \end{aligned} \tag{2}$$

$$Sigmoid(v_{id}) = \frac{1}{1 + e^{-v_{id}}}$$

$$S'(v_{id}) = 2 \times |(Sigmoid(v_{id}) - 0.5| \tag{3}$$

Power Vector for each grid point formula is:

$$pv_i = (pv_{i1}, pv_{i2}, \dots, pv_{ik})$$

$$= \begin{cases} \mathbf{1}, pv_{ik} \text{ is } 1 \text{ if the target at location } i \text{ can be detected by the sensor at location } k \\ \mathbf{0}, \text{otherwise} \end{cases} \quad (4)$$

For each particle $id \in 1, \dots, s$ *do*
Initialize v_{id} (set v_{id} to 0.5)
Repeat
Calculate particle position according equation (2)
Until *coverage* is satisfied according equation (4)
Set $p\_best_{id} = x_{id}$
End for
Repeat
For each particle $id \in 1, \dots, s$ *do*
Evaluate the fitness of particle $id, f(x_{id})$, according sum of one in Particle id (x_{id})
Update p_best_{id} using $p\_best_{id}(t+1)$
$$= \begin{cases} p\_best_{id}(t) & \text{if } f(x_{id}(t+1)) \geq f(p\_best_{id}(t)) \\ x_{id}(t+1) & \text{if } f(x_{id}(t+1)) < f(p\_best_{id}(t)) \end{cases}$$
Update g_best_{id} using $p\_best_{id} \in \{p\_best_0, p\_best_1, \dots, p\_best_s\} = min\{f(p\_best_0(t)), \dots, f(p\_best_s(t))\}$
For each dimension $j \in 1, \dots, N_d$ do
Apply velocity update using equation (1)
End loop
Repeat
Apply position update according equation (2)
Until *coverage* is satisfied according equation (4)
End loop
Until some convergence criteria is satisfied

Fig.3- MDPSO for Sensor Placement

4- Simulation results

In this section we presented the results of algorithm simulation. Here we assumed the number of population as 30, $c_1 = c_2 = 2$, v_{max}=6.0. The value of w from 0.9 to 0.2 is considered as decreasing values. The proposed method is examined on sensor field with different area. The results confirmed the superiority of the proposed algorithm to the Simulated Annealing algorithm [15] considering the convergence factor.

Table1- Comparison between SA algorithms and the proposed algorithm for some target area values

Area	#Sensors	
	SA [15]	MDPSO
4*3	6	4
4*4	7	4
6*3	8	6
6*4	10	8
7*3	9	7
8*3	10	9
9*3	11	10

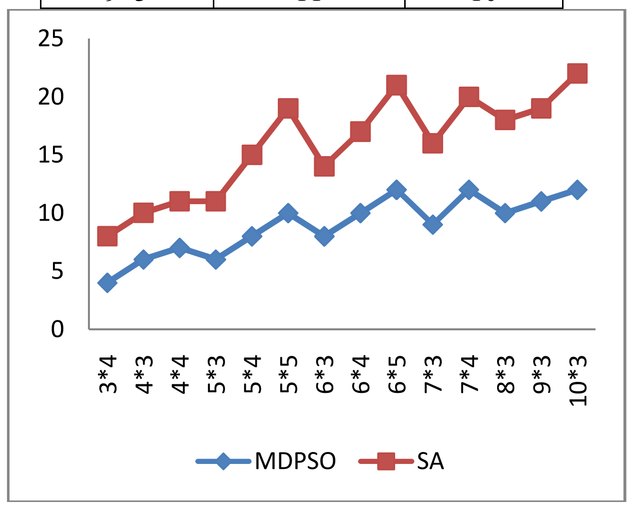

Fig.4- Sensor density (in #Sensors) vs. target area parameter

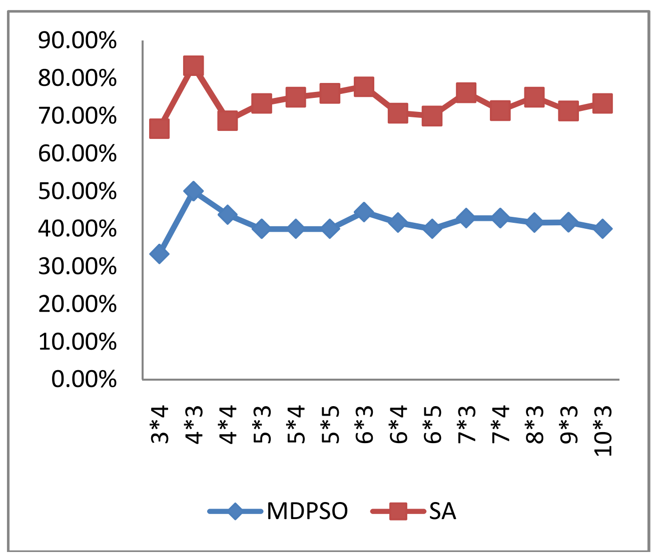

Fig.5- Sensor density (in percent) vs. target area parameter

Sensor density formula is:

$$Sensor\ density(\%) = \left(\sum_{k=1}^{m} \frac{y_k}{n}\right) \times 100\%$$

Where:

$$y_k = \begin{cases} 1\ , if\ a\ sensor\ is\ allocated\ at\ location\ k \\ 0\ , otherwise \end{cases}$$

$and\ n\ \ is\ the\ number\ of\ grids\ in\ sensor\ field.$

5- Conclusion

This paper considers the sensor placement problem for locating targets under constraints (complete coverage of sensor network with minimum costs). Firstly, we defined this NP-complete problem as a combinatorial optimization model then the modified binary PSO algorithm expanded for solving the problem. The results showed that compared to simulated annealing algorithm, the proposed algorithm is able to detect more effectively the optimization solution in a limited time and costs, which provides placement of sensors to increase the coverage on the sensor field also improves the chance of escaping of local optimal. In addition the proposed algorithm is more useful, scalable and durable.

6- References

[1] Kennedy, J. and Eberhart, R. C., "*Particle Swarm Optimization*", Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, , pp. 1942–1948, 1995.

[2] Firip, H. A. and Goodman, E., "*Swarmed feature Selection*", in IEEE Proc. of the 33rd Applied Imagery Pattern Recognition Workshop, AIPR'04, 2004.

[3] Lip, H. B., Tang, Y. Y., Meng, J. and Jp, Y., "*Neural networks learning using vbest model particle swarm optimization*", the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, 2004.

[4] Al- Kazemi, B. and Mohan, C. K., "*Training feed forward neural networks using multi-phase particle swarm optimization*", the 9th International conference on Neural Information, Vol. 5, 2002.

[5] Kennedy, J., and Eberhart, R. C., "*A Discrete Binary Version of The Particle Swarm Algorithm*", Proceedings of Conference on Systems, Man, and Cybernetics, IEEE Service Center, Piscataway, NJ, pp. 4104-4108, 1997.

[6] Rostami, M., Nezam Abadi, H., "*Modified Binary PSO*", ICEE 14th International Conference on Electrical Engineering, Iran 2006.

[7] Akyildiz, I.F., Sankarasubramaniam, W. Su, Cayirci, E., "*Wireless Sensor Networks: a survey*", Computer Networks, Published by Elsevier Science, Vol. 38, No. 4, pp. 393-422, 2002.

[8] Chong, C-Y., Kumar, S.P., "*Sensor Networks: Evolution, Opportunities, and Challenges*", Proceedings of the IEEE, Vol. 91, No. 8, pp. 1247–1256, 2003.

[9] Dhillon, S.S., Chakrabarty, K., and Iyengar, S.S., "*Sensor Placement for Grid Coverage under Imprecise Detections*," Proc. of the Fifth International Conference on Information Fusion, Vol. 2, No.3, pp. 1581-1587, July 2002.

[10] Dhillon, S.S. and Chakrabarty, K., "*Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks*," Proc. of lEEE *WCNC*, Vol. 3, pp. 1609-1614, March 2003.

[11] Chakrabarty, K., Iyengar, S.S., Qi, H., and Cho, E., *"Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks,"* IEEE Trans. on Computers, Vol. 51, No. 12, Dec. 2002.

[12] Sasikumar, P., Vasudevan, S. K., Vivek, C., Subashri, V., "*Heuristic Approaches with Probabilistic Management for Node Placement in Wireless Sensor Networks",* International Journal of Recent Trends in Engineering, Vol. 2, No. 4, November 2009.

[13] Rajagopalan, R., Niu, R., Mohan, Ch. K., Varshney, P. K., Drozd, A.L., "*Sensor placement algorithms for target localization in sensor networks",* IEEE Radar Conference, pp. 1-6, 2008.

[14] Indu, S., Chaudhury, S., Mittal, N. R., Bhattacharyya, A., "*Optimal Sensor Placement for Surveillance of Large Spaces",* Distributed Smart Cameras, Third International Conference on Digital Object Identifier, pp. 1-8, ICDSC 2009.

[15] Lin, Frank Y. S. and Chiu, P. L., *"A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks*", IEEE Communication letters, Vol. 9, No. 1, January 2005.