https://dx.doi.org/10.22161/ijaers.6.3.3 ISSN: 2349-6495(P) | 2456-1908(O)

A hybrid Algorithm for Deployment of Sensors with Coverage and Connectivity Constraints

Timóteo Holanda[1], Tiago Almeida[2], Paulo Cleber M. Teixeira[3], Anna Paula de S. P. Rodrigues[4], and Rafael Lima[5]

Department of Computer Science[1,2,3,5], Food Engineering[4] at Federal University of Tocantins, BRAZIL
Emails: {[1]timoteohss, [2]tiagoalmeida, [4]anna.rodrigues, [3]clebermt,[5]rafael.lima}@uft.edu.br

Abstract— *Finding optimal node deployment for a Wireless Sensor Network (WSN), while maximizing both coverage and connectivity as well as minimizing costs is a challenging task. In the considered scenario, coverage and connectivity are used as QoS (Quality of Service) measures for the desired wireless sensor network. In this case, the problem was handled as a multi-objective optimization problem. In this paper, we propose a hybrid optimization algorithm (GA-BPSO) based on Genetic Algorithm (GA) and Binary Particle Swarm Optimization (BPSO). In order to show the effectiveness of the proposed algorithm, we present some simulations and comparisons with existing methods in the literature.*
Keywords— ***Genetic Algorithms, Particle Swarm Optimization, Wireless sensor networks.***

I. INTRODUCTION

Technological advances brought several benefits in the communication field in the past few years. The combination of wireless communication elements and microcontrollers enabled the development of nodes with sensing capabilities. The joining of multiple nodes allowed the creation of comprehensive low-cost monitoring systems. While each node has restrictions such as power consumption, limited coverage, sensing capabilities and signal processing [1]–[3]; new challenges have been created. These systems of interconnected nodes are denominated in Wireless Sensor Networks (WSN).

Aiming an efficient operation, regardless of the used criteria, its nodes need to form a connected component. This way, it is possible that data can be transmitted by multiple sensors. However, maintaining connectivity coverage across the entire network is of utmost importance. Nodes have limited scope and power source or can be damaged, extinguishing their use in the network.

At the organizational level, each sensor can connect with neighboring sensors in order to reduce the assigned power consumption in its communication, minimizing external interference, and forming a connection network. By its nature, a WSN may run the risk of losing a partition of its network by some possible obstacle blocking the signals sent between sensors, whether by the existence of natural (mountains, trees, valleys, etc.) or artificial (buildings, monuments, walls, etc.) reasons. In this way, we must prevent such occurrence by requiring that each sensor has a defined range in order to have a finite number of neighboring sensors at any instant of time. Taking care in fulfilling this critical requirement may ensure that the sensor mesh remains connected [2], [4].

In order to avoid loss of connection, a network can make use of a restriction called *m*-connectivity. A WSN is said to be *m*-connected if, and only if, each sensor is connected to at least *m* other sensors. Thus, each sensor can hold up to *m*-1 faulty neighboring sensors [1]. Another QoS measure is related to the number of nodes covering a target. This constraint is given by the *k*-coverage restriction, i. e., each target must be covered by at least *k* different sensors [1, 2]. Fig 1 shows a WSN with $k=1$ and $m=1$.

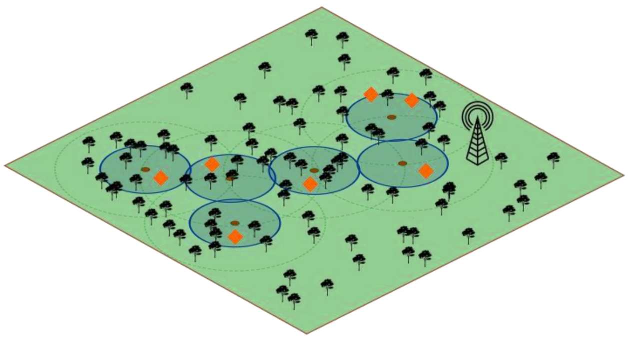

Fig. 1: Example of a one-connected and one-covered WSN.

In [5], the author proposes a hybrid evolutionary algorithm, mixing the benefits of the Particle Swarm Optimization (PSO) algorithm followed by evolutionary operators, in the discovery of the optimal position of sensors in a WSN network. Simulation results show that such a method can not only improve the location accuracy but also reduce its location response time.

Authors in [6] used a LEACH (Low Energy Adaptive Clustering Hierarchy Protocol) based algorithm, mixed with genetic algorithms to achieve increased lifetime and energy efficiency in WSN. The genetic algorithm is used to select cluster heads and create efficient clusters for data transmission. Simulations results show that the proposed hybrid protocol results in prolonged network lifetime and optimal energy consumption for sensor nodes inside a wireless sensor network.

In this sense, this work has as its main objective the proposition of a hybrid algorithm, composed of a Genetic Algorithm and a Particle Swarm Optimization algorithm, similar to the work done by [5], however, discovering an optimal position of sensors, as well as maximizing their coverage and connectivity.

The remainder of this paper is organized as follows.

Section 2 cites the background of Multi-objective Optimization, GAs, PSO and BPSO. Section 3 defines the problem formulation. Section 4 presents GA-BPSO. Section 5 presents results in two case studies. Finally, Section 6 concludes this paper.

II. BACKGROUND

2.1 Multi-objective Optimization

Multi-objective optimization (MOO) aims to find the Pareto optimal solution, forming the *Pareto-front* in the objective space [7]. It can be defined as:

$$f(x) = [f_1(x), f_2(x), \cdots, f_n(x)] \quad (1)$$

where $f_i(x)$ is the ith objective, x is the decision vector for $n > 1$ objectives.

Pareto optimal solutions for a multi-objective problem are virtually infinite. Thus, it is necessary to incorporate various objectives in order to determine a single suitable solution. Methods such *a priori articulation* depends on user indicated preferences before running the optimization, allowing the algorithm to determine a single solution that reflects what the optimal solution should represent, alternatively, *posteriori articulation* requires the user to manually select a single solution from the Pareto optimal set [8].

2.2 Genetic Algorithms

Genetic Algorithms (GAs) are simulated biological evolutions used to solve the optimization of nonlinear problems[9]. Vectors are encoded as possible solutions, which are representations of individuals, and they are made up of binary, real or integer elements, which represents their individual genes. A group of individuals is denoted as a population[10]. A fitness function is used as a means of measuring how close a given individual is to the optimal solution.

A GA starts by generating a random initial population, and a fitness value is calculated to each individual. The higher an individual's fitness, the higher its likelihood of reproduction. Evolution takes place by means of *crossover* and *mutation* operations, producing offspring that replace part of the population. This is repeated until the convergence criteria are met, the fittest individual of the last population is assumed to be the optimal solution found.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a simulation based on the behavior of bird flocks and fish schools, also used as a means of finding optimal solutions for nonlinear problems. Individuals are represented by particles in a swarm and act according to self-acquired knowledge but also with the collective knowledge obtained by the swarm [11].

All particles move in a multidimensional space, where each particle has a position *x* and a speed vector *v* in relation to the time *t*. For each step of time, the velocity of each particle is updated according to the equation (2):

$$v_i^{t+1} = w \cdot \alpha_1 \cdot (p_i^b - x_i) + \alpha_2 \cdot (p_i^{gb} - x_i) \quad (2)$$

where $w$is the inertia factor, $p_i^b$is the best local solution found by the particle so far, p_i^{gb} is the global best position found by all particles of the swarm, α_1 ,$\alpha_2$are coefficients of local and global learning, respectively.

With the new velocity, each particle *i* has its position updated by equation (3):

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

3.2 Binary Particle Swarm Optimization (BPSO)

The BPSO modifies the original PSO algorithm, by using a similar methodology in a discrete binary search model. Therefore, since the position vector is binary, the speed is used as the probability of a bit to change. This way, the speed factor is limited to [0,1] using a Sigmoid function.

Thus, the speed is still obtained using equation (2), but the position is updated using the equation (4):

$$x_i^{t+1} = \begin{Bmatrix} 0 & if\ rand() \geq \ S(v_i^{t+1}) \\ 1 & if\ rand() < S(v_i^{t+1}) \end{Bmatrix} \quad (4)$$

where $rand() \in [0,1]$ and $S(v_i^{t+1})$ is given by equation (5):

$$S(v_i^{t+1}) = \frac{1}{1 + e^{v_i^{t+1}}} \quad (5)$$

III. OPTIMIZATION PROBLEM

This work approaches the problem of given a set of targets $T \subset \Re^2$ and a set of potential positions $P \subset \Re^2$, the *k*-coverage and *m*-connectivity deployment sensors

https://dx.doi.org/10.22161/ijaers.6.3.3 **ISSN: 2349-6495(P) | 2456-1908(O)**

problem is defined as selecting a subset $S \subseteq P$ such that each target in T is covered by at least k sensors and each sensor in S connected with at least m other sensors. In this context, a target is covered by a sensor, when within sensing range of that sensor. In addition, a sensor is said to be connected with another sensor whenever they are in each other connectivity range.

A solution S is said to be optimized if it minimizes the number of sensors while respecting the constraints. In addition, $S$is considered the global optimum if, for every solution $S' \subseteq P$, the number of sensors in $S$is less or equal to the number of sensors in S'.

This work uses the same mathematical model as [12]. In this way, this problem is modeled as an integer decision problem. The decision variables are stated in equations (6) to (8).

$$b_{ij} = \begin{Bmatrix} 1, & \text{if target } T_i \text{ is covered by } S_i \\ 0, & \text{otherwise.} \end{Bmatrix} \tag{6}$$

$$c_{ij} = \begin{Bmatrix} 1, & \text{if a sensor } S_i \text{ is connected} \\ & \text{to another sensor } S_j \\ 0, & \text{otherwise.} \end{Bmatrix} \tag{7}$$

$$q_i = \begin{Bmatrix} 1, & \text{if a potential position } P_i \\ & \text{is selected for node placement} \\ & 1 \leq i \leq |P| \\ 0, & \text{otherwise.} \end{Bmatrix} \tag{8}$$

where T_i is the ith element of T, and S_i is the ith element of S. Thus, the problem can be formulated as follows:

$$Minimize \sum_{i=1}^{|P|} q_i \tag{8}$$

subject to:

$$\sum_{j=1}^{|P|} b_{ij} \geq k, \ \forall i, \ 1 \leq i \leq |T| \tag{9}$$

$$\sum_{j=1}^{|P|} c_{ij} \geq m, \ \forall i, \ i \neq j, 1 \leq i \leq |P| \tag{10}$$

Constraint (9) ensures that every target is covered by at least $k$sensor nodes, while constraint (10) states that each sensor should be connected with at least m other ones.

IV. THE PROPOSED SOLUTION: GA-BPSO

In order to approach the considered problem, it is proposed a hybrid evolutionary algorithm combining the Genetic Algorithm and the Binary Particle Swarm Optimization (GA-BPSO).

4.1 Encoding

A sequence of potential positions S is encoded as a binary vector. Whether a position i of S has the value 1, it means that the ith potential position is selected to deploy a sensor. Fig 2 shows an example of such encoding.

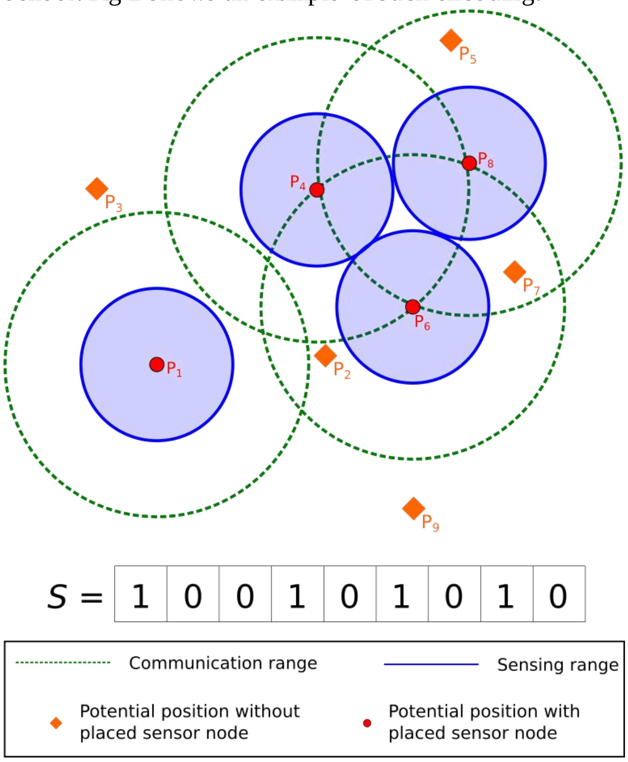

Fig 2: Example of encoding based on the potential position of sensor nodes.

4.2 Fitness

The fitness function is divided into three other objectives: F1, F2, and F3. F1 stands to minimize the number of potential positions selected by the algorithm. This quantity is related to the number of employed sensors of the network. F2 and F3 handle the k-coverage and m-connectivity restrictions, respectively.

Let $N = |P|$ be the total of potential positions of Pthat have been selected for placing sensor nodes, the first objective function is given by equation (11):

$$Minimize \ F_1 = \frac{N}{|P|} \tag{11}$$

https://dx.doi.org/10.22161/ijaers.6.3.3 **ISSN: 2349-6495(P) | 2456-1908(O)**

Let $Cov(T_i)$ be the set of sensors nodes within sensing range of target T_i, the second objective is then described by equation (12):

$$Maximize\ F_2 = \frac{1}{N \cdot k} \sum_{i=1}^{N} FCov(T_i) \quad (12)$$

where $FCov(T_i)$ defines the full coverage by sensor nodes based on the set of sensors covering every target. Function $FCov(T_i)$ is given by equation (13):

$$FCov(T_i) = \begin{cases} k, & \text{If } |Cov(T_i)| \geq k \\ |Cov(T_i)| - k, & \text{otherwise.} \end{cases} \quad (13)$$

Let $Com(P_i)$ be the set of sensors nodes within coverage range of P_i, the third objective can be described as follows:

$$Maximize\ F_3 = \frac{1}{N \cdot m} \sum_{i=1}^{N} FCom(P_i) \quad (14)$$

where $FCom(P_i)$ defines the full communication by sensor nodes based on the set of sensors covering every active sensor node. Function $FCom(P_i)$ is defined in equation (15):

$$FCom(P_i) = \begin{cases} m, & \text{If } |Com(P_i)| \geq m \\ |Com(P_i)| - m, & \text{otherwise.} \end{cases} \quad (15)$$

It is important to note that both F_2 and F_3 conflict with F_1, this happens because the objective aims to maximize the k-coverage and m-connectivity, this may be obtained by placing a substantial quantity of sensor nodes, shadowing the first objective. This way, the multiobjective is then modeled as a weighted sum. These weights can be applied without any transformation of the objective functions, as they merely represent the relative importance of the objectives [8].
Let W_i be a weight value applied to each objective, and all objectives are summed up into a single scalar objective function generating the following model:

$$\text{Maximize Fitness} = W_1 \times (1.0 - F_1) + W_2 \times F_2 + W_3 \times F_3 \quad (15)$$

Subject to:

$$0 \leq W_1, W_2, W_3 < 1 \quad (16)$$

where

$$W_1 + W_2 + W_3 = 1 \quad (17)$$

4.3 Description of GA-BPSO
The proposed approach is a combination of a Binary Particle Swarm Optimization (BPSO) algorithm and a Genetic Algorithm (GA). Following the logic presented by [13] and [14], this hybrid method is divided into two phases. In the first phase, the fitness of the generated population of size P_{size} is calculated, then the population is divided into two parts of equal size. The best individuals are used as input for the GA, while the worst ones are used as input for the BPSO algorithm. In this way, the approach takes the benefits of GA, which GAs has genetic operators, so the individuals can evolve and find better offspring. While PSO does not provide such operators, it can perform exploration of solutions, which hopefully can guide the particles to possibly finding global optimal solutions.
In the second phase, a new population is generated by GA operators using the fittest individuals, while the worst individuals are enhanced by the BPSO evolution. These new and evolved individuals are merged back into a single population of size P_{size} and sent back to phase 1 until the termination criteria are met.

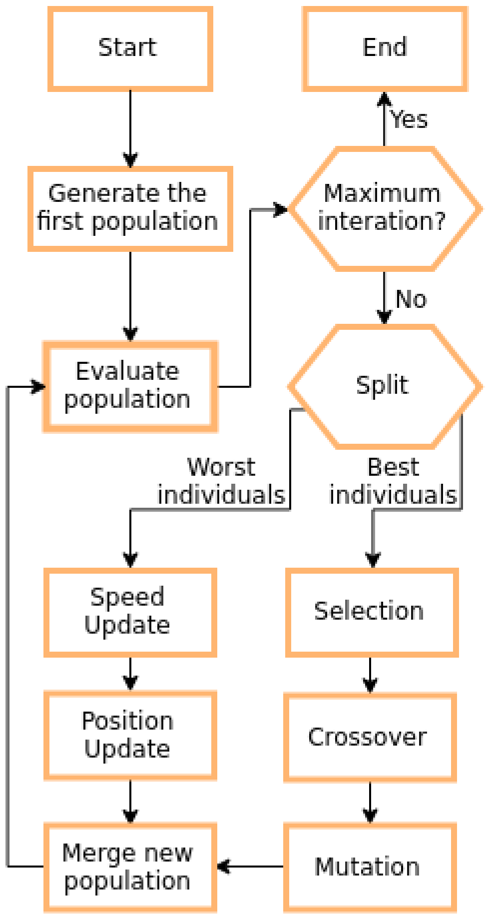

Fig 3: GA-BPSO diagram

4.4 Advantages of GA-BPSO

PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population. Both use fitness values to evaluate the population. Both update the population and search for the optimum using stochastic algorithms. But, PSO is distinctly different from other evolutionary type methods in a way that it does not use the filtering operation and the members of the entire population are maintained through the search procedure so that information is socially shared among individuals to guide the search towards the best position in the search space [15], [16].

One of the advantages of GAs is the ability to finding local optima, by means of genetic operators that gradually improve the fitness of its individuals throughout generations. However, GAs can do less exploration for global search when compared to PSO solutions [17]. One disadvantage of PSOs is premature convergence. In order to avoid this effect, the PSO can be used to find better solutions from individuals with smaller fitness values in the population. On a PSO solution, every individual shares information among themselves, this way, such individuals converges to a better solution faster than GA [18]. Therefore, the proposed algorithm (GA-BPSO) combines the advantages of both GAs and PSO.

V. RESULTS

The evaluation of the GA-BPSO algorithm is done using the same two case studies as used by [12]. In both cases it is assumed an sensing field of $300m^2$. Case Study I considered that each potential position could be positioned only on cross-points over a grid pattern with steps of $25m$. In the other hand, Case Study II assumed random potential positions inside the given sensing field.

Table.1: Presents all the simulation parameters.

Table 1: Simulation parameters.

Parameter	Value
Max iterations	100
Number of target points	100
No. of potential positions	100-500
Communication range	100 m
Sensing range	50 m
Initial population size	60
Mutation rate	3%
Elitism rate	50%
W_1, W_2, W_3	0.4, 0.3, 0.3
$BPSO - [V_{min}, V_{max}]$	[-6,6]
$BPSO - \alpha_1, \alpha_2$	2
$BPSO - \nabla w$	[0.6-0.2]

Fig. 4 and Fig. 6 depict results in terms of the number of selected potential positions by varying the number of given potential positions, ranging from 100 to 500, with steps of 100. In both scenarios, a total of 100 target points were given and (k, m) values vary from (1,1) to (3,2).

It should be noted that the number of given potential positions does not affect the quality of generated solutions. This is due to the fact that the optimal solution for any objective function is not mutable by the search parameters. It can also be observed the difference of selected potential positions varying k values, this is explained by a rise in complexity of the network mesh, when trying to adjust itself aiming to met its objective. The GA-BPSO results are compared with [12]. Fig. 5 shows the comparison results of Study Case I, as well as Fig 7, shows the comparisons results of Study Case II.

Comparing Fig 4 and Fig 6, there is a difference generated by the initial distribution of the network mesh. As depicted by Fig 4, on a grid-like pattern, GA-BPSO performs better. This is expected due to a guaranteed consistent distribution of sensor nodes on the field. This does not happen when using random potential positions instead. It is important to note the lack of substantial improvement when considering scenarios (k=2, m=1) and (k=2, m=2). A large communication range is responsible for keeping the sensors from disconnecting from each other while the algorithm evolves its population. This would not happen on a larger field though.

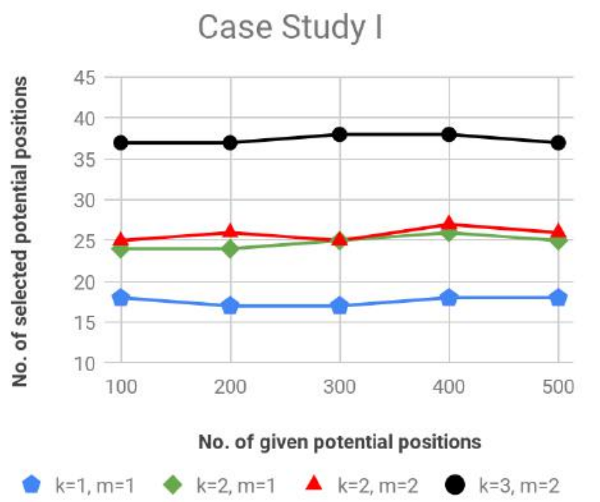

Fig 4: Comparison in terms of the number of selected sensor nodes for Case Study I.

Observing Fig 7, with instance (k=3, m=1) and Fig 6 with instance (k=4, m=1), it can be seen that GA-BPSO performed worse than the algorithm proposed by [12]. An investigation should be carried out in order to determine the sensitivity of GA-BPSO using (k > 4) scenarios as well as on larger sensing fields.

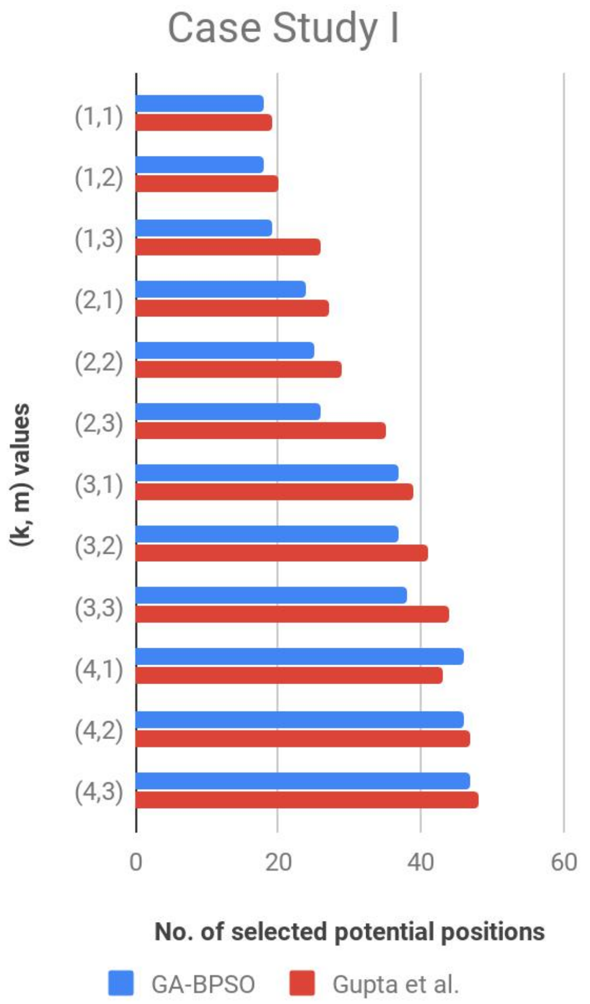

Fig 5: Comparison in terms of selected sensor nodes for Case Study I

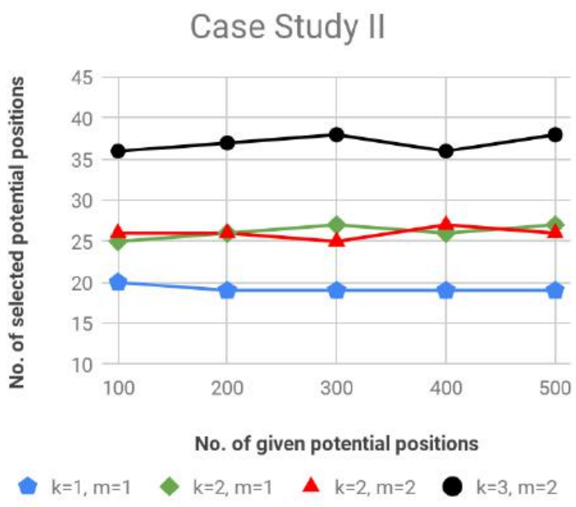

Fig 6: Comparison in terms of the number of selected sensor nodes for Case Study II.

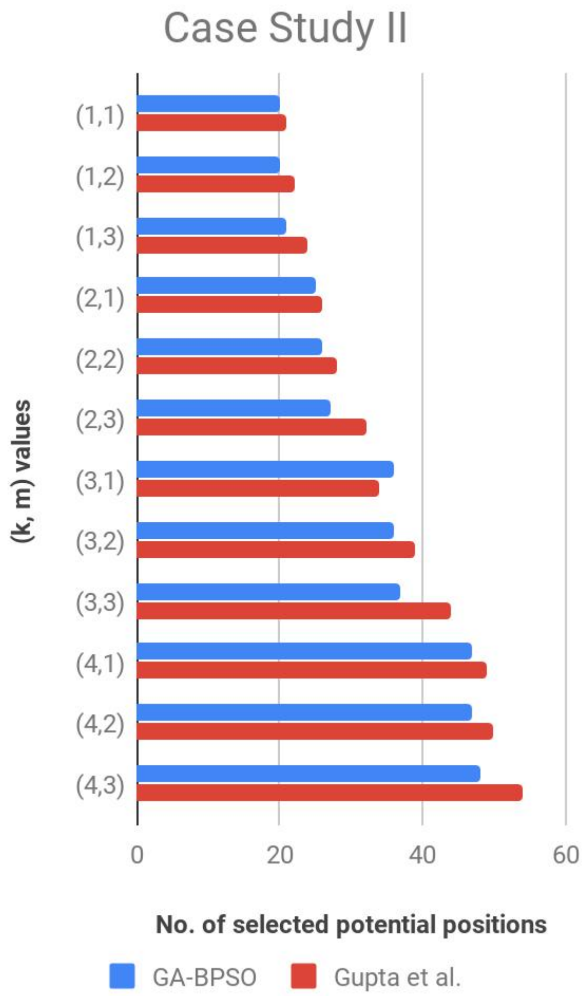

Fig 7: Comparison in terms of selected sensor nodes for Case Study II

VI. CONCLUSION

This paper proposed an alternative method, called GA-DPSO, for finding optimized solutions to the problem of sensor deployment, with *k*-coverage and *m*-connectivity restrictions. The GA-DPSO uses a combination of PSO algorithm and GA, mixing up the global search feature provided by the PSO algorithm (exploration) while using the local search with a GA (exploitation).

Results suggest that a large connectivity field can, in fact, make the network rely on the *k*-coverage for any further optimization. In some cases such as Case Study I instance (k=2, m=3), this algorithm found a solution at least 27% better than the results reported in [12]. The comparison results between both methods conclude that GA-BPSO performs better than the proposed in [12]. It improved, not only on finding reasonable less active sensors solutions, but also balancing the contradiction between the number of active sensors, coverage, and connectivity, improving on the WSN localization efficiency.

https://dx.doi.org/10.22161/ijaers.6.3.3 ISSN: 2349-6495(P) | 2456-1908(O)

As future work, some more experimental studies with larger requirement parameters should be conducted. A clustering algorithm can be implemented utilizing AG-BPSO internally, intending to improve its processing power. In addition, mobile sensor nodes should also be considered on experiments.

REFERENCES

[1] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive Mob. Comput.*, vol. 4, no. 3, pp. 303–334, 2008.

[2] C. Lo and N. Ansari, "The Progressive Smart Grid System from Both Power and Communications Aspects," *IEEE Communications Surveys Tutorials*, vol. 14, no. 3, pp. 799–821, 2012.

[3] B. Wang, "Coverage Problems in Sensor Networks: A Survey," *ACM Comput. Surv.*, vol. 43, no. 4, pp. 32:1–32:53, Oct. 2011.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[5] S. Liu, "Optimization analysis of WSN location process based on hybrid PSO algorithm," in *2017 IEEE International Conference on Unmanned Systems (ICUS)*, 2017, pp. 78–80.

[6] T. Sharma, G. S. Tomar, R. Gandhi, S. Taneja, and K. Agrawal, "Optimized Genetic Algorithm (OGA) for Homogeneous WSNs," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 4, pp. 131–140, 2015.

[7] Z.-K. Feng, W.-J. Niu, J.-Z. Zhou, C.-T. Cheng, H. Qin, and Z.-Q. Jiang, "Parallel Multi-Objective Genetic Algorithm for Short-Term Economic Environmental Hydrothermal Scheduling," *Energies*, vol. 10, no. 2, p. 163, Jan. 2017.

[8] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Struct. Multidiscip. Optim.*, vol. 41, no. 6, pp. 853–862, Jun. 2010.

[9] F. Pistolesi, B. Lazzerini, M. D. Mura, and G. Dini, "EMOGA: A Hybrid Genetic Algorithm With Extremal Optimization Core for Multiobjective Disassembly Line Balancing," *IEEE Trans. Ind. Inf.*, vol. 14, no. 3, pp. 1089–1098, Mar. 2018.

[10] M. A. Rodriguez-Guerrero, A. Y. Jaen-Cuellar, R. D. Carranza-Lopez-Padilla, R. A. Osornio-Rios, G. Herrera-Ruiz, and R. de J. Romero-Troncoso, "Hybrid Approach Based on GA and PSO for Parameter Estimation of a Full Power Quality Disturbance Parameterized Model," *IEEE Trans. Ind. Inf.*, vol. 14, no. 3, pp. 1016–1028, Mar. 2018.

[11] G. Srinivasan and S. Visalakshi, "Application of AGPSO for Power loss minimization in Radial Distribution Network via DG units, Capacitors and NR," *Energy Procedia*, vol. 117, pp. 190–200, Jun. 2017.

[12] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k -coverage and m - connected node placement in target based wireless sensor networks," *Comput. Electr. Eng.*, vol. 56, pp. 544–556, 2016.

[13] A. Agnihotri and I. K. Gupta, "A hybrid PSO-GA algorithm for routing in wireless sensor network," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad, 2018, pp. 1–6.

[14] C. Li, R. Zhai, H. Liu, Y. Yang, and H. Wu, "Optimization of a heliostat field layout using hybrid PSO-GA algorithm," *Appl. Therm. Eng.*, vol. 128, pp. 33–41, 2018.

[15] L. He, W. Li, Y. Zhang, and J. Cao, "Review of Swarm Intelligence Algorithms for Multi-objective Flowshop Scheduling," in *Lecture Notes in Computer Science*, 2018, pp. 258–269.

[16] S. Bulkan, "Comparison of Genetic Algorithm and Particle Swarm Optimization for Bicriteria Permutation Flowshop Scheduling Problem," *Int. J. Comput. Intell. Res.*, vol. 4, no. 2, 2008.

[17] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA for global maximization," *Int. J. Open Problems Compt. Math*, 2009.

[18] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*.