Journal of Network Intelligence ©2017 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **2**, Number **3**, November **2017**

Node Localization in WSN using Heuristic Optimization Approaches

Chin-Shiuh Shieh, Van-Oanh Sai, Tsair-Fwu Lee, and Quang-Duy Le
Department of Electronics Engineering
National Kaohsiung University of Applied Sciences, Taiwan
csshieh@cc.kuas.edu.tw

Yuh-Chung Lin
Department of Information Application and Management
Tajen University, Taiwan

Trong-The Nguyen*
Department of Information Technology
Haiphong Private University,Vietnam
35 Danlap Road, Du Hang Kenh, Le Chan, Hai Phong, Viet Nam
vnthe@hpu.edu.vn

Received Jun 2017; revised August 2017

Abstract. . *In wireless sensor networks, the localization of sensor nodes is a key issue for many applications. Normally, in localization problem, the positions of sensor nodes can be inferred based on position information of three or more anchor nodes whose locations are given in advance. We approach this problem with heuristic optimization technology. In the first part of this article, a comparative study is made among Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and three recently developed heuristic optimization methods, namely Grey Wolf Optimization (GWO), Firefly Algorithm (FA), and Brain Storming Optimization (BSO). Their relative performance, in terms of accuracy and convergence speed, on typical test functions for optimization study is obeserved. Next, these heuristic optimization methods are applied to the localization problem in WSNs to see their relative merits and limitations. We examine the number of nodes can be localized and the execution time.*
Keywords: Node localization; Heuristic optimization, Genetic algorithm, Particle swarm optimization; Grey wolf optimization; Firefly algorithm; Brain storming optimization.

1. **Introduction.** Wireless sensor networks (WSN) are widely being used in different situations to perform various monitoring tasks such as search and rescue in disasters, target tracking and a number of tasks in smart environments [1]. The location information of the sensors can be used in the design of efficient network routing algorithms. Thus, how to locate sensor nodes as accurate as possible is an important issue in many applications [2]. An obvious method of localization is to equip each node with a global positioning system (GPS). But when equipping each node with a global positioning system is not an attractive solution because of cost, size and energy constraints of sensor node [3]. The other reason is that GPS can only work outdoors but not indoors. Therefore in localization problem of WSN, unknown location nodes will be estimated by using known position of anchors nodes. Node localization in WSN is a two phase processes: ranging

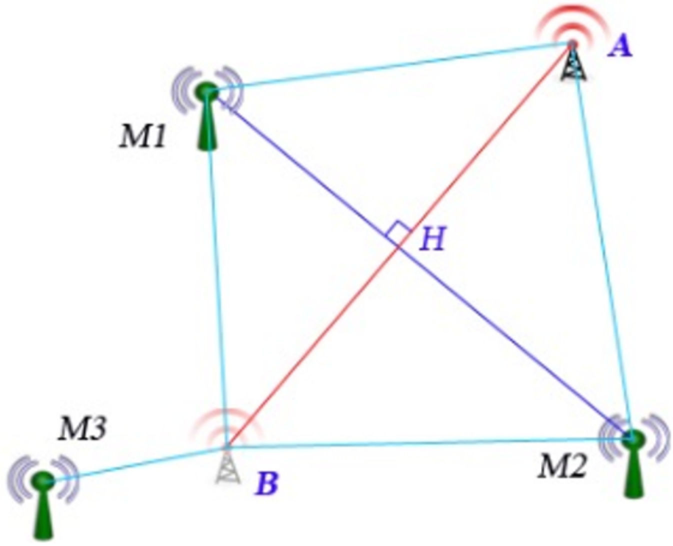

FIGURE 1. . Example situation with only 2 neighboring anchors

phase and estimation phase. In the ranging phase, sensor nodes estimated their distances to anchors by receive signal strength indicator (RSSI), angle of arrival (AoA), or any of the time-based on techniques, such as time of arrival (ToA), time difference of arrival (TDoA) technologies. In the estimation phase, the position of the target nodes is estimated by using the ranging information either by solving simultaneous equations or by minimizing the localization error using an optimization algorithm that minimizes localization error [3-5]. In this study, a comparative study is made among genetic algorithm, particle swarm optimization and three newly developed heuristic optimization algorithms, namely grey wolf optimization, firefly algorithm, and Brain storming optimization on typical optimization test functions. After that, these heuristic optimization methods are applied to the node localization problem to investigate their relate merits and limitations. The rest of this paper is organized as follows: a brief review of the node localization problem in WSN is given in Session 2. Session 3 lists heuristic optimization technologies used in this study. Section 4 explains an improvement we had made to deal with the case when there are only 2 anchors. Section 5 presents empirical results on optimization test functions and on node localization problem. Finally, we conclude our findings in Session 6 .

2. **Node Localization in WSNs.** The objective of node localization problem in WSN is to estimate the location of n target nodes using prior information about the location of m anchors and distance between these target nodes and the neighboring anchors. Assuming that the anchors coordinates is (x_i, y_i) where:

$$\mathrm{x_i} = [\mathrm{x_1}, ..., \mathrm{x_m}], \mathrm{y_i} = [\mathrm{y_1}, \ldots, \mathrm{y_m}] \tag{1}$$

And the locations of target nodes need to estimate are (x_j, y_j) ,where

$$\mathrm{x}_j = [\mathrm{x_1}, ..., \mathrm{x}_n], \mathrm{y}_j = [\mathrm{y_1}, \ldots, \mathrm{y}_n] \tag{2}$$

In the ranging phase, anchors node will estimate distance from their position to neighboring target nodes. The effect of measurement noise is simulated as a Gaussian additive white noise. The obtained distance from a sensor to an anchor i:

$$\hat{d}_1 = d_i + n_i \tag{3}$$

where d_i is the actual distance and n_i is the measurement noise. The actual distance is given as:

$$\mathrm{d_i} = \sqrt{(\mathrm{x}-\mathrm{x_i})^2 + (\mathrm{y}-\mathrm{y_i})^2} \tag{4}$$

where (x, y) is the location of target node, and (x_i, y_i) is the location of i^{th} neighboring anchor. The measurement noise n_i has a random value uniformly distributed therefore the range of obtained distance $\widehat{\mathrm{d_i}}$ can be expressed as $[d_i - d_i * p_n/100, d_i + d_i * p_n/100]$. In second phase, the localization error function can be formulated as:

$$\mathrm{f}(\mathrm{x},\mathrm{y}) = \frac{1}{\mathrm{M}}\sum_{\mathrm{i}=1}^{\mathrm{M}}\left(\sqrt{(\mathrm{x}-\mathrm{x_i})^2 + (\mathrm{y}-\mathrm{y_i})^2} - \hat{d_i}\right)^2 \tag{5}$$

where $M \geq 3$ is the number of anchors within the transmission radius of target nodes (x,y). The localization function in Equation (5) is exactly the object function subject to minimization.

3. **Heuristic Optimization Methods.** Numerous heuristic optimization methods have been proposed in recent decades. Although with heuristic approaches global optimum, even bounds to global optimum, can't be guaranteed. In practice, they are well recognized in locating near optimum with justified cost. Here we list heuristic optimization methods used in our study.

3.1. **Genetic algorithm.** Genetic Algorithm (GA) is one of the evolutionary algorithms which was invented by John Holland in the early 1970's. GA are robust search and optimization techniques that were developed based on ideas and techniques from genetic and evolutionary theory [6].

3.2. **Particle Swarm Optimization.** Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by simulations of various interpretations of the movement of individuals in a bird flock or fish school [7].

3.3. **Grey Wolf Optimizer algorithm.** The Grey Wolf Optimizer (GWO) algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. GWO was proposed by Mirjalil in 2014 [8]. Group hunting is another interesting social behavior of grey wolves.

3.4. **Firefly Algorithm.** Firefly algorithm (FA) is a heuristic algorithm which is inspired by the interesting natural characteristics of fireflies' flashing behavior. According to the characteristics of fireflies, Xin-She Yang has developed firefly algorithm at Cambridge University in late 2007 and 2008 [9-10].

3.5. **Brain Storm Optimization.** Brain Storm Optimization (BSO) is a population-based swarm intelligence algorithm based on the collective behavior of human creative problem-solving process. BSO was proposed by Yuhui Shi in 2011. The ideas in creating this algorithm was inspired by the observation that human are social animals and are the most intelligent animals in the world [11].

```
For each sensor node X:
  Evaluate the number of neighboring anchors, M, of node X.
  if (M≥3)
     Estimate the location of node X by minimizing the localization error in (5).
  else if (M=2)
     Assume that there are 2 neighboring M1 and M2.
     Estimate the location of this node by minimize localization error in (5) with M=2.
     Assume that position B is obtained.
     Evaluate the number of neighboring anchors, m, of location B.
     if (m ≥ 3)
        Node X is located at position A.
     else
        Node X is located at position B.
  else
   Node X is not localizable.
```

FIGURE 2. Pseudo code for cases with only 2 neighboring anchors

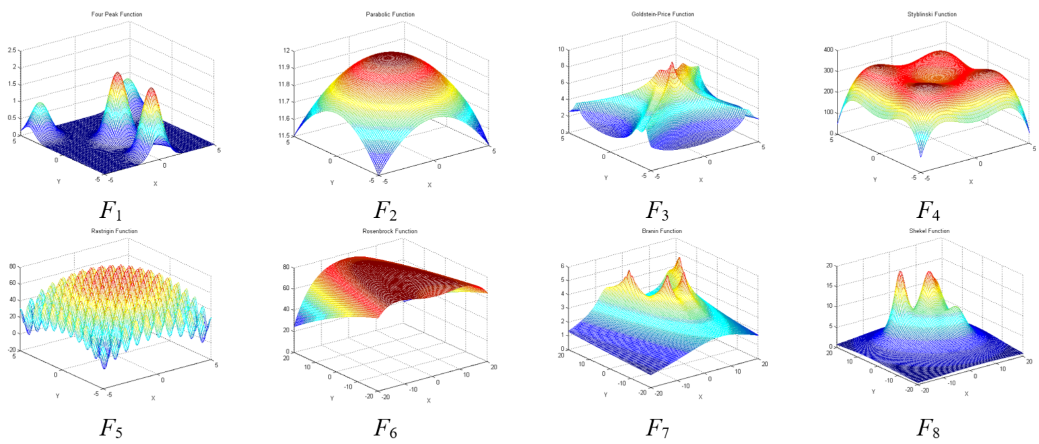

FIGURE 3. The optimization test functions

4. **Improvement in Node Localization.** Generally, the localization error function in (5) is only meaningful when there are 3 or more neighboring anchors (anchors nodes within the the transmission range of this target sensor node). Thus, to localize more target nodes, it is required to enlarge the transmission range of sensor nodes in order to be reached by 3 or more anchors. However, larger transmission range implies higher power consumption which is infeasible under power constraint for more sensor nodes. Here we consider a special case with which a sensor has only 2 neighboring anchors but still localizable. Figure 1 is an illustration for the situation under consideration. A sensor node X covered by 2 anchor nodes M1 and M2 can locate at either location A or B. It is obviously that node X is located at position A if node X is reachable by only 2 anchors. Otherwise, node X is located at position B.

5. **Experimental Results.** To examine the feasibility and efficiency of those heuristic optimization methods, we apply them to typical optimization test functions in Section 4.1 and to the node localization problem in Section 4.2.

5.1. **Experiments on test functions.** To compare their performance in solving non-linear optimization problems, those methods listed in Section 3 were applied to find out

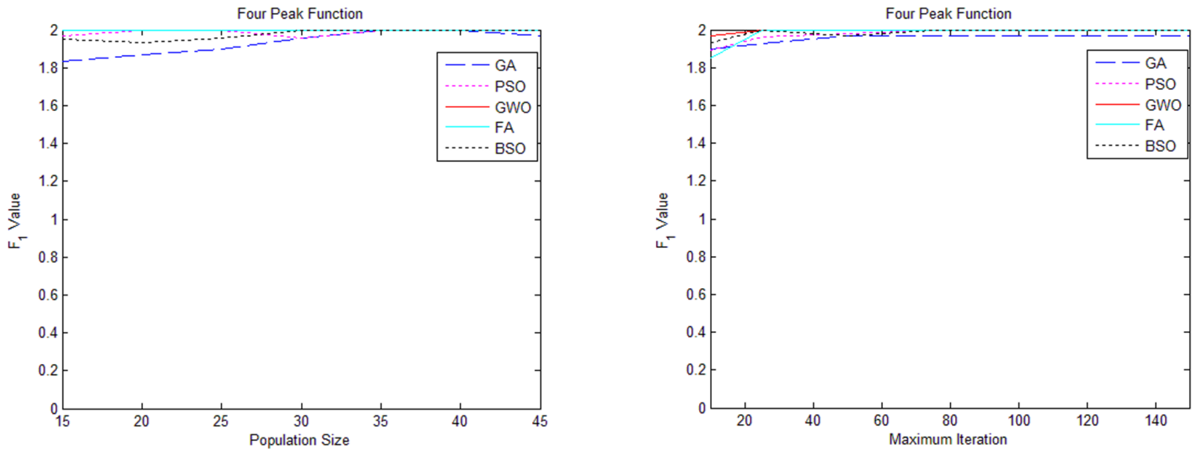

FIGURE 4. Experiment results on F1

global optimization value (maximum value) of various test functions [12], as follows and in Figure 1:

Four Peak function:

$$F_1(x,y) = exp(-(x-4)^2 - (y-4)^2) + exp(-(x+4)^2 - (y-4)^2) + \\ 2[exp(-x^2 - y^2) + exp(-x^2 - (y+4)^2)] \tag{6}$$

Parabolic Function:

$$\mathrm{F}_2\left(\mathrm{x},\mathrm{y}\right) = 12 - \left(\mathrm{x}^2 + \mathrm{y}^2\right)/100 \tag{7}$$

Goldstein-Price Function

$$\begin{gathered} \mathrm{F}_3\left(\mathrm{x},\mathrm{y}\right) = 10 + \log_{10}\left(1/(A*B)\right); \\ A = 1 + (1+\mathrm{x}+\mathrm{y})^2\left(19 - 14\mathrm{x} + 3\mathrm{x}^2 - 14\mathrm{y} + 6\mathrm{xy} + 3\mathrm{y}^2\right) \\ B = 30 + (2\mathrm{x}-3\mathrm{y})^2\left(18 - 32\mathrm{x} + 12\mathrm{x}^2 + 48\mathrm{y} - 36\mathrm{xy} \;\; + 27\mathrm{y}^2\right) \end{gathered} \tag{8}$$

Styblinski Function:

$$\mathrm{F}_4\left(\mathrm{x},\mathrm{y}\right) = 275 - \left[\left(\mathrm{x}^4 - 16\mathrm{x}^2 + 5\mathrm{x}\right)/2 + \left(\mathrm{y}^4 - 16\mathrm{y}^2 + 5\mathrm{y}\right)/2 + 3\right] \tag{9}$$

Rastrigin Function:

$$\mathrm{F}_5\left(\mathrm{x},\mathrm{y}\right) = 80 - \{20 + \mathrm{x}^2 + \mathrm{y}^2 - 10[\cos\left(2\pi\mathrm{x}\right) + \cos\left(2\pi\mathrm{y}\right)]\} \tag{10}$$

Rosenbrock Function:

$$\mathrm{F}_6\left(\mathrm{x},\mathrm{y}\right) = 70^*\left\{\left[\left(20 - \left\{\left(1 - \mathrm{x}/\left(-7\right)\right)^2 + \left[\mathrm{y}/6 + \left(\mathrm{x}/\left(-7\right)\right)^2\right]^2\right\}\right) + 150\right]/170\right\} + 10 \tag{11}$$

Branin Function:

$$\mathrm{F}_7\left(\mathrm{x},\mathrm{y}\right) = 5 - \log_{10}\left[\mathrm{y} - 5.1\mathrm{x}^2/\left(4\pi^2\right) + \left(5\mathrm{x}/\pi - 6\right)^2 + \left(10 - 5/4\pi\right)^*\cos\left(\mathrm{x}\right) + 10\right] \tag{12}$$

Shekel Function:

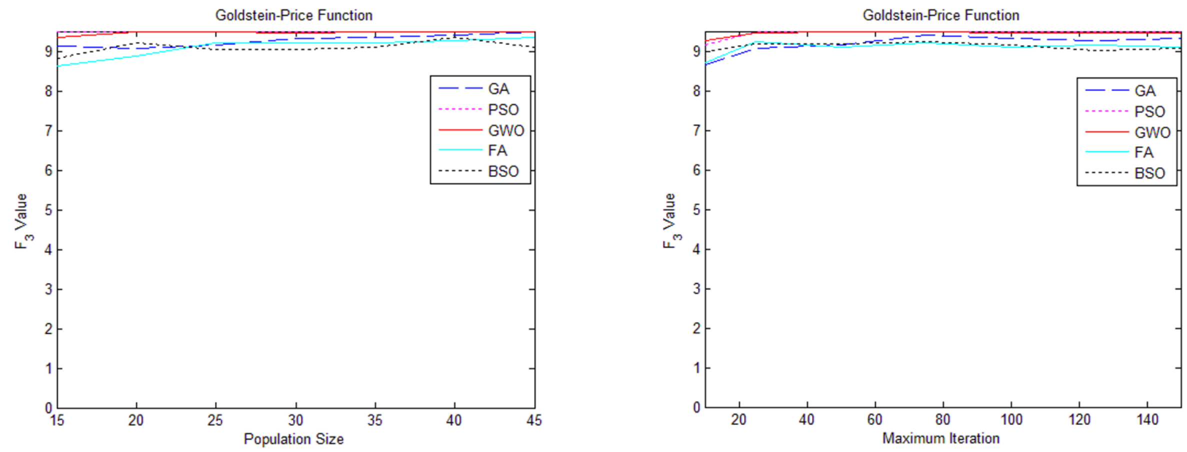

FIGURE 5. . Experiment results on F3

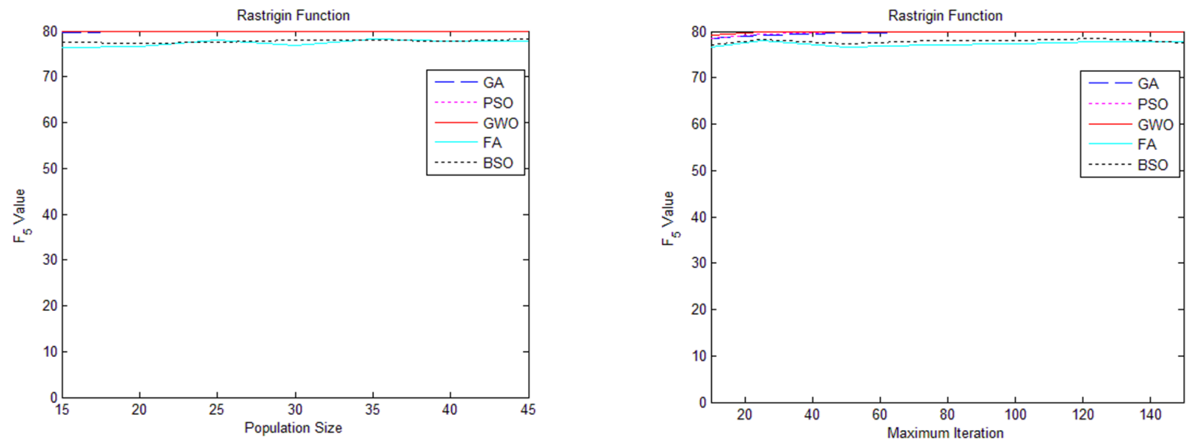

FIGURE 6. Experiment results on F5

$$\begin{gathered}
F_8\left(x,y\right) = 100 * \left\{\frac{1}{A} + \frac{1}{B} + \frac{1}{C} + \frac{1}{D} + \frac{1}{E}\right\} \\
A = \left[9 + \left(x-4\right)^2 + \left(y-6\right)^2\right] \\
B = \left(20 + x^2 + y^2\right) \\
C = 14 + \left(x-8\right)^2 + \left(y+3\right)^2 \\
D = 11 + \left(x-8\right)^2 + \left(y-8\right)^2 \\
E = 6 + \left(x+6\right)^2 + \left(y-7\right)^2
\end{gathered} \tag{13}$$

The parameter settings for GA, PSO, GWO, FA, and BSO are listed as follows: Genetic Algorithm (GA): Crossover Rate (CR) decrease from 0.9 to 0.5. Mutation Rate (MR): 0.05. Particle Swarm Optimization (PSO): Learning factor: $c_1 = c_2$=2. Inertia weight (w): Decrease from 0.9 to 0.4. Grey Wolf Optimizer (GWO): Parameter a: linearly decreased from 2 to 0 Firefly Algorithm (FA): Absorption coefficient: $\gamma = 1.0$. Randomness reduction: $\delta = 0.97$. Initial randomness scaling factor: $\alpha_0 = 0.2$ Brain Storm Optimization (BSO): $m = 5, p_5a = 0.2, p_6b = 0.8, p_6biii = 0.4, p_6c = 0.5, k = 20$.

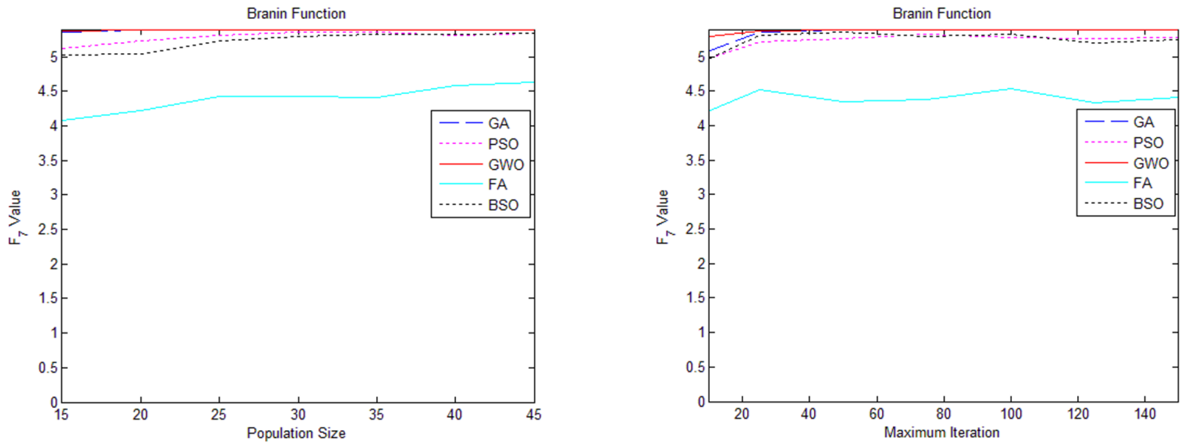

FIGURE 7. . Experiment results on F7

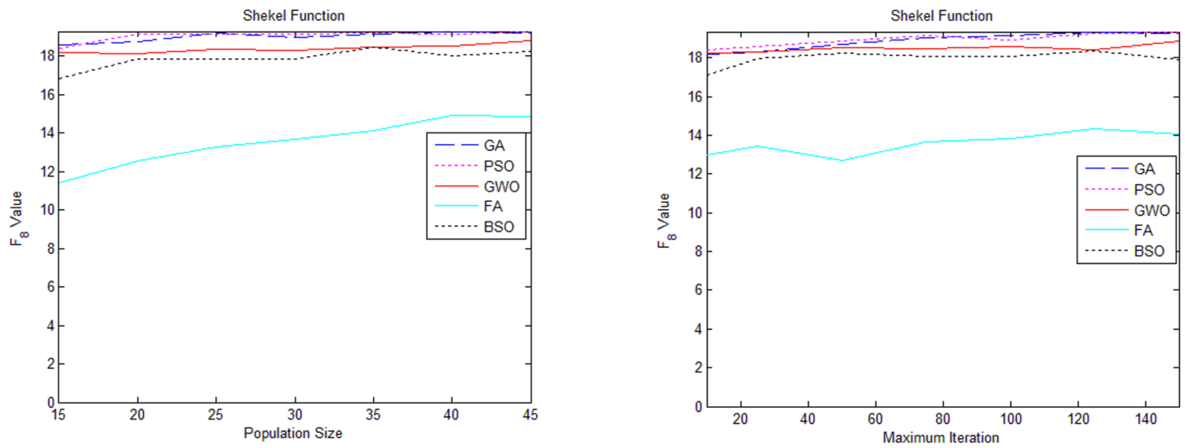

FIGURE 8. Experiment results on F8

All the experiment data are average of thirty runs. We examine all the algorithms mentioned above from two perspectives. In case 1, we set the max number of iteration/generation to 100 and change the population size from 15 to 45. In case 2, the population size is set to 30 and the maximal number of iterations varies from 10 to 150. In the case of single peak functions like Parabolic, Styblinski and Rosenbrock functions, all methods almost immediately get the global optimization value even when the population size and number of iteration are small. In the graph, the lines for different algorithms overlap with each other. The global optimization values of Parabolic, Styblinski and Rosenbrock functions that are obtained by these algorithms are approximately equal to 12, 350 and 80, respectively. In our results, GWO outperforms, in terms of accuracy and execution time, all others for most of the test functions except $F_8(x, y)$-Shekel function, as shown in Figure 4 to Figure 9. The performance of GA is good when the population size and the max iteration are large enough. Otherwise, in some test functions, GA will be trapped at local optimum and lead to inaccurate result. PSO algorithm is good in almost all the cases except for $F_7(x, y)$ - Branin functions shown in Figure 7. Generally, the accuracy of FA and BSO is not good as the others, as shown from Figure 5 to Figure 8. Especially for Branin and Shekel functions, FA is easily to be trapped at local peaks as we can see in Figure 7 and Figure 8 Figure 9 shows the processing time in dealing

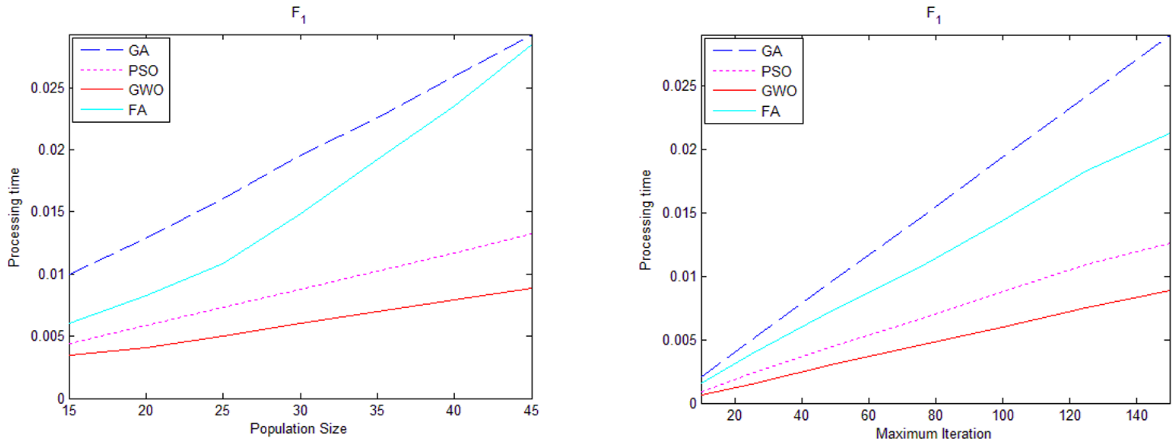

FIGURE 9. Experiment results on F1

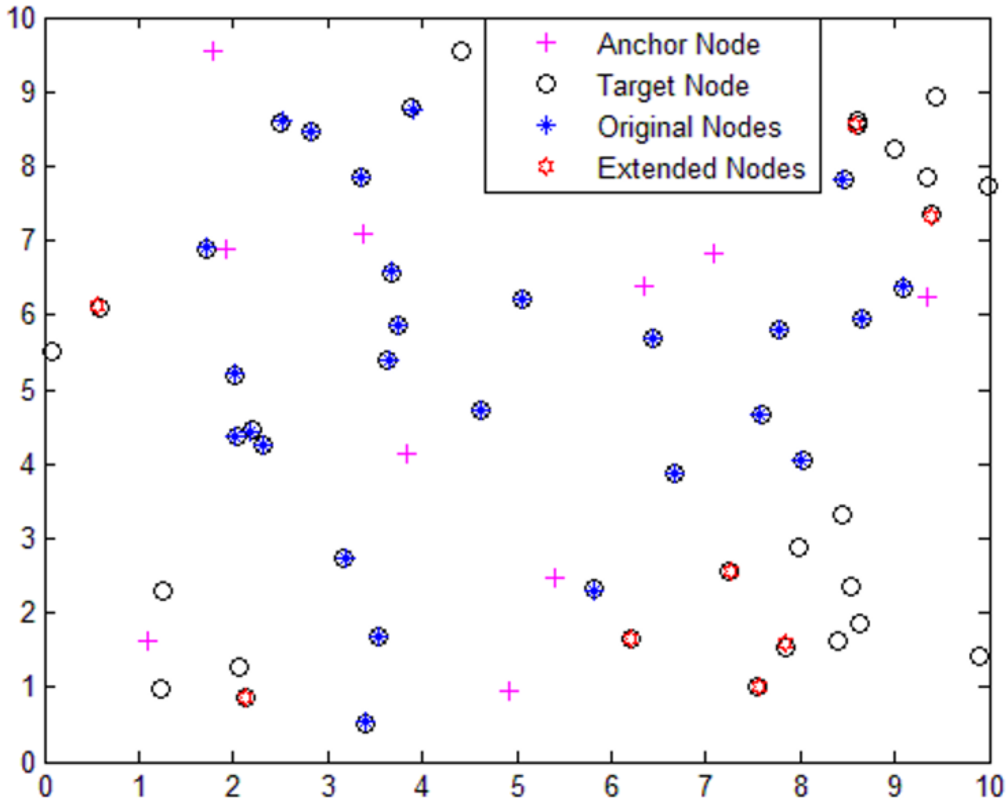

FIGURE 10. A scenario of node localization

with $F_1(x, y)$ function. We ignore the BSO because the processing time of BSO is several times longer than others. The processing time for other test functions has similar pattern as for$F_1(x, y)$. When considering the processing time, GWO is the best choice. Next is PSO, then followed by FA and GA. BSO has the longest execution time than others.

5.2. **Experiments on node localization problem.** In this subsection, the position of target nodes is estimated by minimizing the localization error function in (5) using an optimization algorithm. The performance of different heuristic optimization methods will be compared. In our scenario, there are 30 target nodes and 10 anchors randomly deployed in a 10×10 sensing field, as shown in Figure 9. The transmission radius of each

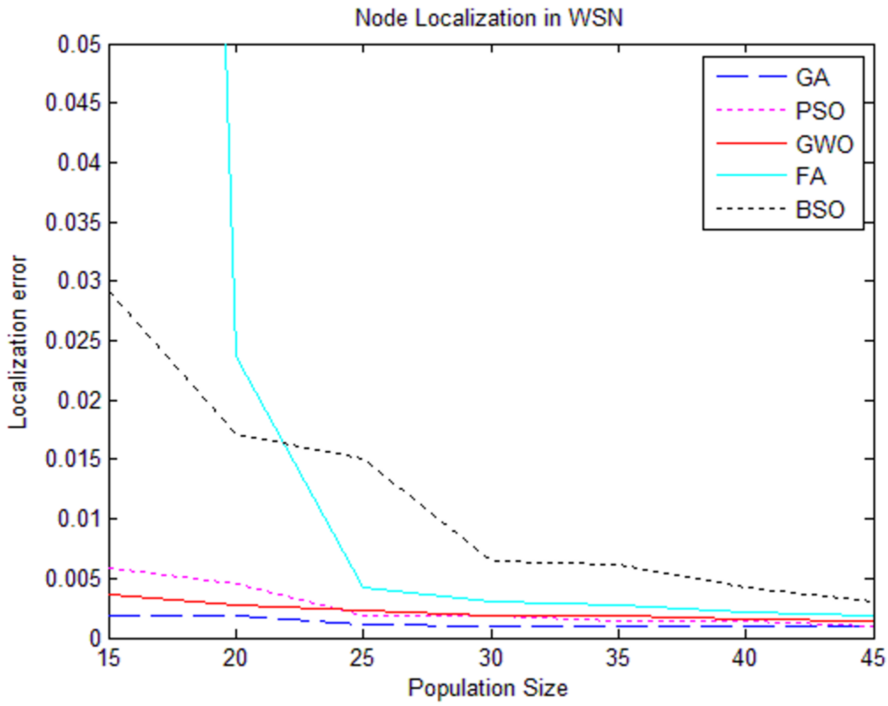

FIGURE 11. Localization error value vs. population size

sensor nodes is set to R=4 units and the parameter of measurement noise is set to $P_n = 2$. The parameters for algorithms in discussion are the same as those in Subsection 5.1. We also examine the performance of individual methods from 2 different perspectives: fixed maximal number of iteration set to 100 with variable population size from 15 to 45, and a fixed population size set to 30 and variable maximal number of iterations from 10 to 150.

Figure 10 and 11 show the localization error values given by different heuristic optimization methods. The results of GA and PSO are good even the population size is small. But in GA, with the number of max iteration (generation) is small (smaller than 50) the results are not so good. In recent heuristic methods, the obtained results using GWO compared with GA and PSO are quite competitive, not only in the case of the population size is small but also when we change the max iteration. FA is easy to be trapped in local optimization value when the population size is smaller than 25 and the max iteration is smaller than 50. BSO does not have a good performance in accuracy with both cases. We next examine the effectiveness of the proposed improvement discussed in Section 4. As indicated in Figure 9, shows an example of wireless sensor network deployment. There are 10 anchors and 50 target nodes randomly deployed in area of 10 × 10 square units. 26 out of 50 target nodes are able to be localized when using the original localization algorithm in (5). With the improvement proposed in Section 4, number of localizable sensors increase from 26 to 34.

In Figure 12, the number of target nodes and anchors are set to 50 and 10 respectively. The transmission radius of each sensor node is changed from 2.5 to 6 units. In Figure 13, the transmission radius of each sensor node is set to 3 units and then the number of target nodes varies from 30 to 100 nodes. The number of anchors will be set to 20% of that number of target nodes.

Figure 13 and Figure 14 show that, in both scenarios, our proposed improvement can significantly increase the percentage of localized nodes. It means that the number of

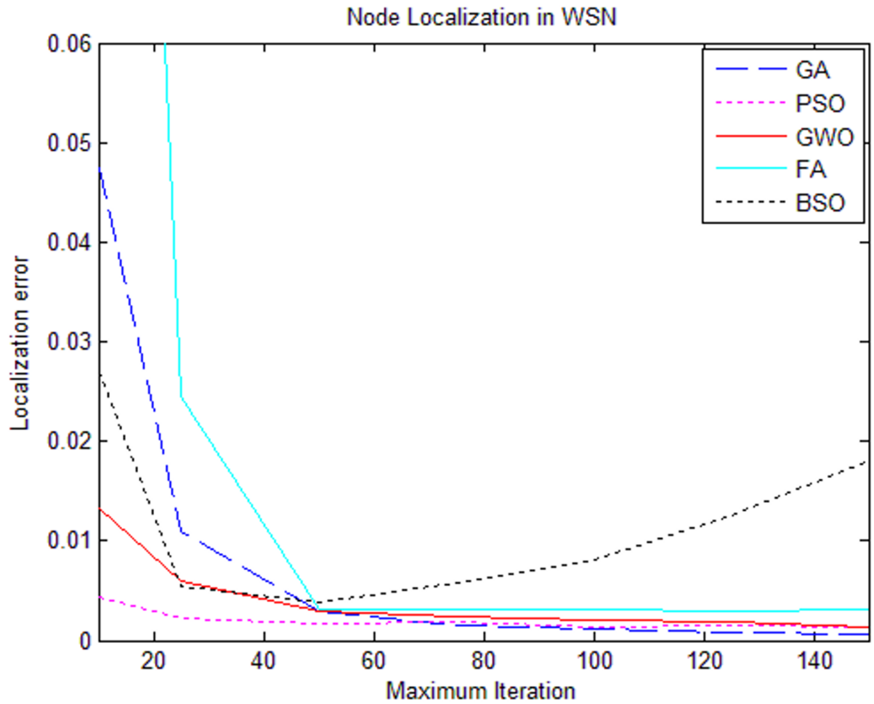

FIGURE 12. Localization error value vs maximal number of iterations

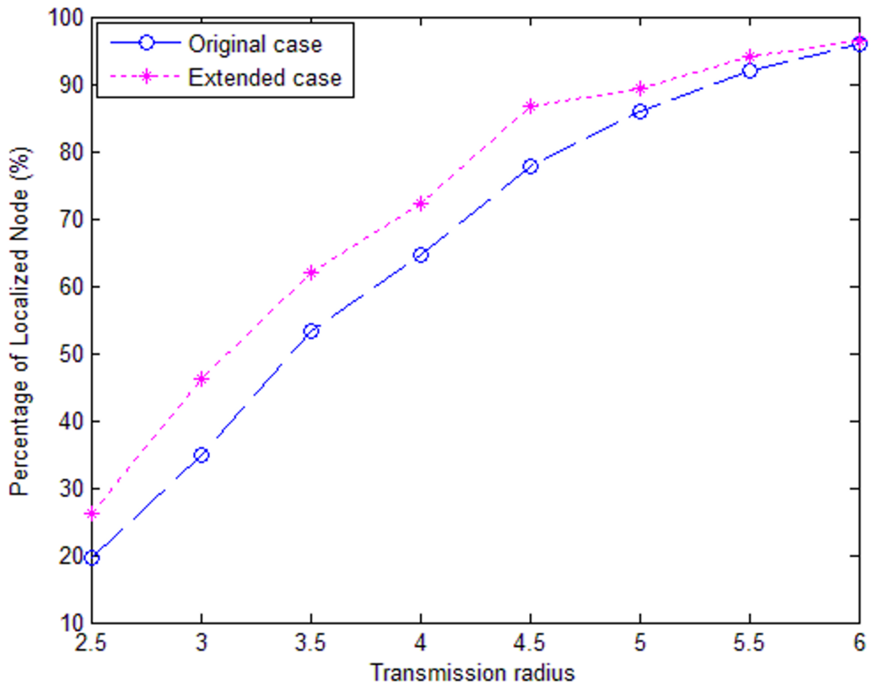

FIGURE 13. Percentage of localized node vs. transmission radius

localized nodes is efficiently higher than in original case at a cost of slightly increase in the computational cost.

6. **Conclusions.** Wireless sensor network (WSN) is an integral part of the Internet of Things (IoT), and it makes huge of devices to share data for improving the environmental

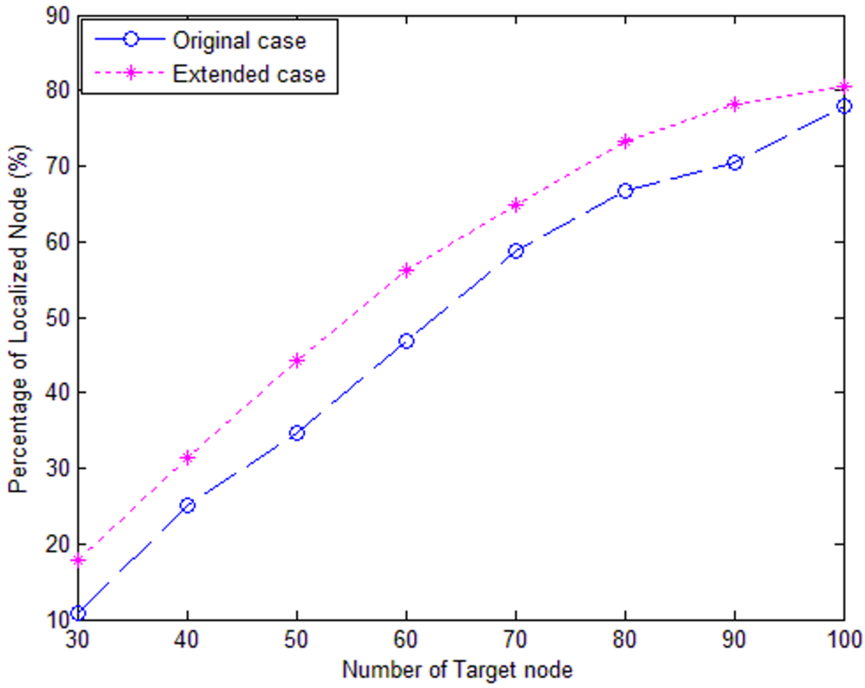

FIGURE 14. Percentage of localized nodes vs. number of target nodes

user control. As sensor nodes are energy constrained, reducing the energy consumption and maximizing the network lifetime are the major research challenge in WSN. Clustering formation is the modern energy efficient techniques, but it suffers from hot spot problem and introduces connectivity issues in the network. Unequal Clustering evenly distributes the load, eliminates the hot spot problem and maximizes the network lifetime. Many advanced algorithms are for energy efficient WSN via equal and unequal grouping. In this paper, we explained various equal and unequal clustering algorithms with their objectives, characteristics, classification, merits, and demerits. All the reviewed algorithms are also compared based on different cluster properties, Cluster Head (CH) properties, and clustering process. We examined the literature of clustering formation and presented tables and discussion. The design of an appropriate equal and unequal clustering algorithms depends on the user and application requirements.

Acknowledgment. We are grateful to the support from Ministry of Science and Technology, Taiwan, R.O.C. to this study, under the grant numbers MOST 104-2221-E-151-068-.

REFERENCES

[1] A. Pal, Localization algorithms in wireless sensor networks: Current approaches and future challenges, *Network Protocols and Algorithms*, vol. 2, pp. 45-73, 2010.

[2] Van-Oanh Sai, Chin-Shiuh Shieh, Trong-The Nguyen, Yuh-Chung Lin, Mong-Fong Horng, and Quang-Duy Le, Parallel firefly algorithm for localization algorithm in wireless sensor network, *Proc. of the Third International Conference on Robot, Vision and Signal Processing*, pp. 300-305 , 2015

[3] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, Bio-inspired node localization in wireless sensor networks, *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp. 205-210, 2009.

[4] A. Gopakumar and L. Jacob, Localization in wireless sensor networks using particle swarm optimization, *Proc. of IET International Conference on Wireless, Mobile and Multimedia Networks*, pp. 227230, 2008.

[5] R. V. Kulkarni and G. K. Venayagamoorthy, Particle swarm optimization in wireless-sensor networks: A brief survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, pp. 262-267, 2011.

[6] W.-Y. Lin, W.-Y. Lee and T.-P. Hong, Adapting crossover and mutation rates in genetic algorithms, *Journal of Information Science and Engineering*, vol. 19, pp. 889-903, 2003.

[7] J. Kennedy, Particle swarm optimization, *Encyclopedia of Machine Learning*, pp. 760-766, Springer, 2010.

[8] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[9] X.-S. Yang, Firefly algorithms for multimodal optimization, Stochastic Algorithms: *Foundations and Applications*, pp. 169-178, Springer, 2009.

[10] X. S. Yang and X. He, Firefly algorithm: recent advances and applications, *International Journal of Swarm Intelligence*, vol. 1, p. 36-50, 2013.

[11] Y. Shi, Brain storm optimization algorithm, Advances in Swarm Intelligence, pp. 303-309, Springer, 2011.

[12] V. O. Sai, C. S. Shieh, Y. C. Lin, M. F. Horng, T. T. Nguyen, Q. D. Le and J. Y. Jiang, Comparative study on recent development of heuristic optimization methods, *Proc. Of the Third International Conference on Computing Measurement Control and Sensor Network*, pp. 68-71, May 20-22, 2016.