Journal of Information Hiding and Multimedia Signal Processing ©2019 ISSN 2073-4212
Ubiquitous International Volume **10**, Number **1**, January **2019**

Diversity Enhanced Ion Motion Optimization for Localization in Wireless Sensor Network

Jeng-Shyang Pan[1,2,3], Trong-The Nguyen[2*,4], Shu-Chuan Chu[5],

[1]College of Computer Science and Technology,
Sandong University of Science and Technology, Qingdao, China

[2]Fujian Provincial Key Lab of Big Data Mining and Applications,
Fujian University of Technology, Fujian, China
*Corresponding Author: jvnthe@gmail.com

[3]Department of Information Management,
Chaoyang University of Technology, Taiwan

[5]College of Science and Engineering,
Flinders University, Tonsley, 5042 South Australia

Thi-Kien Dao[4], Truong-Giang Ngo[4]

[4]Department of Information Technology
Hai-Phong Private University,
No.36 Dan Lap Rd., Le Chan, Hai Phong, Vietnam

Received July 2018; revised November 2018

Abstract. *A new optimization algorithm (named DIMO) based on diversity enhanced Ion Motion Optimization (IMO) is proposed. Diversity learning strategy and random perturbations are applied to improve IMO by modifying its individual evolutionary information. A set of selected benchmark functions and an estimation localization in Wireless sensor network (WSN) are used to test the performance of the proposed algorithm. The experimental results compared with the others algorithms in the literature shows that the proposed approach are superior to the other algorithms in optimization accuracy, convergence speed, and robustness.*
Keywords: DIMO, Wireless sensor network, Meta-heuristic.

---

1. **Introduction.** Wireless sensor network (WSN) has turned into an emerging technology, which is possible to acquire dynamic data from the environment [1, 10, 11]. Node location of sensor net is essential for some WSN applications e.g., monitoring, tracking, logistics[2]. The application used the measurements of correlation sensors with physical locations by even approximates [3]. Global Positioning System (GPS) can be applied to aware the location of sensor nodes, but this way is not often efficiency for thousands of nodes in a WSN with equipped GPS. Moreover, the environments such as indoor or underground or many obstacles are not suitable for GPS. Instead of all equipped GPS nodes, only a few nodes that referred as anchors to derive the positions by estimating their distances to nearby nodes by measuring RSS (received strength signal) and TOA (time of arrival). Meta-heuristic algorithm is one of the best ways to solve successfully the estimation and optimization problems [4].

A recent meta-heuristic algorithm in the field of evolutionary algorithms, Ion motion algorithm (IMO) is a kind of physical heuristic group intelligence optimization algorithm which simulates the movement rules of anions and cations [5]. IMO algorithm has several advantages such as simple operation, few parameter setting, and easy programming implement. However, like the other evolution processing development of the algorithms, the IMO algorithm also has several disadvantages such as low convergence speed and easy to fall into local optimum in the search process. Since it has just been put forward, it has not been paid much attention by scholars in various fields. At present, the performance improvement related articles have not been published, and the theoretical system of the algorithm is not perfect. Thus, in this mentioned point of view, an improving version based on diversity ion motion algorithm (DIMO) is implemented in this paper. Update the diversity individual and different components with evolutionary information in the solid stage are introduced to enhance the population diversity and the speed of convergence of the algorithm.

The experiments for the selected benchmark functions and estimation localization in WSN are compared with IMO algorithm, and lastest meta-heuristic algorithms in the literature such as adaptive particle swarm optimization (AD-PSO) [6] and Modified differential evolution (MDE) [7], the proposed algorithm has obvious improvement in convergence speed and accuracy

2. **Ion Motion Optimization algorithm.** The Ion Motion Optimization algorithm [5] divides the ion candidate solutions into two sets, namely, Anion (negative ion charge) set and Cation (positive ion charge) set. They perform different evolutionary strategies in the liquid phase and the solid phase and circulate between the liquid and the solid phase to achieve the purpose of optimation. Ions in IMO algorithm can move toward best opposite charges. It means that anions move toward the best cation, on the other hand, cations move toward the best anion. The movement of ions in this algorithm can guarantee the improvement of all ions over the course of iterations. Their movement power depends on the attraction/repulsion forces between them. The amount of this force specifies momentum of each ion. The process of the algorithm is represented by the following steps.

Initialization. An initial random population is randomly generated according to a uniform distribution within the lower and upper boundaries with D dimention.

Liquid phase. In the liquid phase, the anion group *(A)* and the cation group *(C)* are updated according to the following patterns, respectively.

$$A_{i,j} = A_{i,j} + AF_{i,j} \times (Cbest_j - A_j), \tag{1}$$

$$C_{i,j} = C_{i,j} + CF_{i,j} \times (Abest_j - C_j), \tag{2}$$

where $Cbest$ and $Abest$ are cation and anion optimization respectively. Subscript $i = 1, 2, 3, ...NP/2$, (NP/2 is the population size of ions), and $j = 1, 2, 3..D$, The optimal anion and cation are the anion and cation with the lowest fitness value in the entire anion group and the cation group respactively for minimization problem. The resultant of anions attracted force $AF_{i,j}$ and $CF_{i,j}$ are mathematically modeled as follows:

$$AF_{i,j} = \frac{1}{1 + e^{-0.1/AD_{i,j}}} \tag{3}$$

$$CF_{i,j} = \frac{1}{1 + e^{-0.1/CD_{i,j}}} \tag{4}$$

where $AD_{i,j}$ and $CD_{i,j}$ are the distances of the ith anion from the best cation, and cation from the best anion in in the jth dimension respectively. $AD_{i,j} = |A_{i,j} - Cbest_j|$

and $CD_{i,j} = |C_{i,j} - Abest_j|$.

Solid phase. The ion is gradually gathered with iteration near the optimal ion by the gravitational force. The solid phase was set for breaking the phenomenon of excessive concentration, and also providing diversity for the algorithm in case of over-concentration of ions to make the algorithm fall into a local optimum. The ion motion gradually slows down like the physical process as the iteration proceeds from the initial intense motion, and gradually the liquid state ions will recrystallize into crystals. The process of recrystallization was simulated in IMO was known as a solid phase.

$$A_j = \begin{cases} A_j + \varphi_1 \times (Cbest - 1)\,, & if\ rand > 0.5 \\ A_j + \varphi_1 \times (Cbest)\,, & otherwise \end{cases} \tag{5}$$

$$C_j = \begin{cases} C_j + \varphi_2 \times (Abest - 1)\,, & if\ rand > 0.5 \\ C_j + \varphi_2 \times (Abest)\,, & otherwise \end{cases} \tag{6}$$

Termination condition. Completion of the solid phase evolution strategy to determine whether to achieve the termination conditions of the algorithm. The termination conditions include the presupposition accuracy, the number of iterations, and so on. If it is reached, the optimal ion is directly output; otherwise, the anions and cations are returned to the liquid phase from the solid phase and continue to be iterated. In such a process, anions and cations are circulated in liquid phase and solid stage, and the optimal solution is gradually obtained with iteration.

3. **An Improved Ion Motion Algorithm.** Two aspects can be applied improvements for enhancing global optimization ability of IMO such as the generating ions in the solid stage and individual regeneration. The role of the solid phase is mainly to maintain the diversity of the population. This stage provides the modes for the evolution of ions included adding a (-1, 1) random disturbance and reinitializing individuals, however, the execution probability is very low, and so the effect on the convergence rate and the population diversity is minimal. The solid-state also does not consider the degree of evolution directly and directly depends on random disturbance to supplement the way of population diversity, which is very unfavorable for algorithm convergence. Moreover, the actual effect is to stretch in the direction of each dimension in the individual dimension. The way of generating decision has its own random unpredictability, specific analysis of the impact of the algorithm are as follows: in the early stage of evolution, population diversity random disturbance has a certain blindness, although it may have some other new individual local information, thus increasing the population diversity. However, since they do not have the information of evolutionary direction, the trend of evolution that originally has certain search direction is damaged to a certain extent, thus affecting the convergence speed of algorithm; in the later stage of evolution, individuals are generally closer to the globally optimal solution, and are disturbed by random step size. It is easy to get a relatively large offset, and cannot search fine around the optimal solution, so it is difficult to get the global optimal solution. Considering carry information about individuals for evolutionary process, we modify early search process with a random perturbation in the solid stage and propose alternative part as following in the formulas.

$$A_j = A_j + \varphi_1 \times (Cbest - 1) + rand() \times (A_j - A_n)\,, \tag{7}$$

$$C_j = C_j + \varphi_2 \times (Abest - 1) + rand() \times (C_j - C_n)\,, \tag{8}$$

where φ_1 and φ_2 are random arrange [-1,1]; *rand()* is random arrange [0,1]. A_n and C_n is are anionic and individuals different from A_j and C_j respectively. The new model could be the best individual of the population in evolution for exchange information. Compared with the original model, the new one has more the combination that would help to increase diverse population, and reduce the risk of falling into the local optimal. Moreover, the new model is more likely to get the new individual than the original individual, so it would improve the convergence speed of the whole algorithm to obtain the global optimal solution.

In the other hand, individuals are generated through the liquid phase and the solid phase, however, a key factor affecting the convergence speed of the algorithm hasn't been improved yet. A probability has introduced the literature [8] that the probability of current individual is farther away from the optimal position than its reverse individual. If current individuals are improved in current iteration, reversed individuals also have the ability to get the best solution that provide other search locations differed from the most individuals of population. So the participation of reversed individuals in the next iteration can provide new evolutionary information for other individuals. The individuals can effectively increase the diversity of population. The probability of reversed individual and the current individual are generated for next generation as one to one competition.

$$\overline{X_i} = lb_j + ub_j - X_j \tag{9}$$

where lb_j and ub_j are the minimum and maximum values for each dimension of the contemporary candidate solutions, respectively. The following steps are as follows:

Step 1. Initialization

Generate number population for ions randomly with dividing into anion and cation groups, max iterations, boundaries; Calculation all ion fitness values; Select worst anions and cations

Step 2. Perform the liquid phase

Update anion and cation according to Eq. (1, 2).

Step 3. Enhance diversity population

Obtain new anionic and cation population according to Eq.(7,8). Check if the individual is out of boundaries, get the renewed individual in boundaries according to Eq.(9).

Step 4. Calculate fitness

Get the fitness for anion and cation population, if the fitness of current ions <= fitness of the best previous ions, distribute anions and cations around best ions.

Step 5. Termination condition

Determines whether the termination condition is achieved, such as the preset accuracy and the maximum iteration number. The algorithm ends and outputs the best result. Otherwise, it will go back to step 2 enter the liquid stage again.

4. **Experiment Simulation for the Optimization Problem.** The selected test functions as shown in Table 1 from the CEC2005 [9] function list are used to test the performance of the proposed algorithm. The results of the proposed algorithm compared with IMO algorithm, new neighborhood-based as adapted particle swarm optimization AD-PSO [6], and a recent improved differential evolution algorithm (M-DE) [7]. The performance comparison of test algorithms mainly focuses on two aspects included accuracy and convergence of the proposed algorithm as shown in Table 2 and Figures 1 and 2.

The accuracy of the optimal solution: when the number of iterations reaches the maximum iterations, the accuracy of the optimal solution obtained by the algorithm is the best, the worst, the mean and the variance of the optimal solution obtained in many independent operations. Convergence speed: before the function evaluation number reached the maximum value, times consumed when the optimal solution obtained by the algorithm reaches the predetermined termination error value. Setting parameters: NP is set to 50,

Table 1. The selected benchmark functions

Test Functions	D	Range
$f_1(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	30	$-100 \le x_i \le 100$
$f_2(x) = \sum_{i=1}^{D} (10^6)^{i-1/D-1} \times x_i$	30	$-100 \le x_i \le 100$
$f_3(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$-600 \le x_i \le 600$
$f_4(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$-5.12 \le x_i \le 5/12$
$f_5(x) = -\sum_{i=1}^{D} c_i \times \exp(\sum_{j=1}^{3} a_{ij} (x_j - P_{ij})^2)$	3	$-0 \le x_i \le 1$

the predetermined error value of the test function *Ter_Err* is set to *1.0e-10*, *MaxIter* is set 200, each algorithm runs independently is set to 30.

Table 2 shows the proposed DIMO compared with AD-PSO, M-DE and IMO for the selected test functions shows that DIMO is very competitive with other meta-heuristic algorithms. The present algorithm can hence provide very good exploitation. Results are averaged accuracy of the optimal solution, the standard deviation, best value, and the worst values. The highlight is its result reach to known optimization values. Clearly, the most highlight belongs to DIMO.

The success rate of achieving the preset accuracy when the algorithms converged to the required precision level in the 30 independent experiments. Investigates the convergence speed of the algorithm, and the parameters of each algorithm remain with the same condition running.

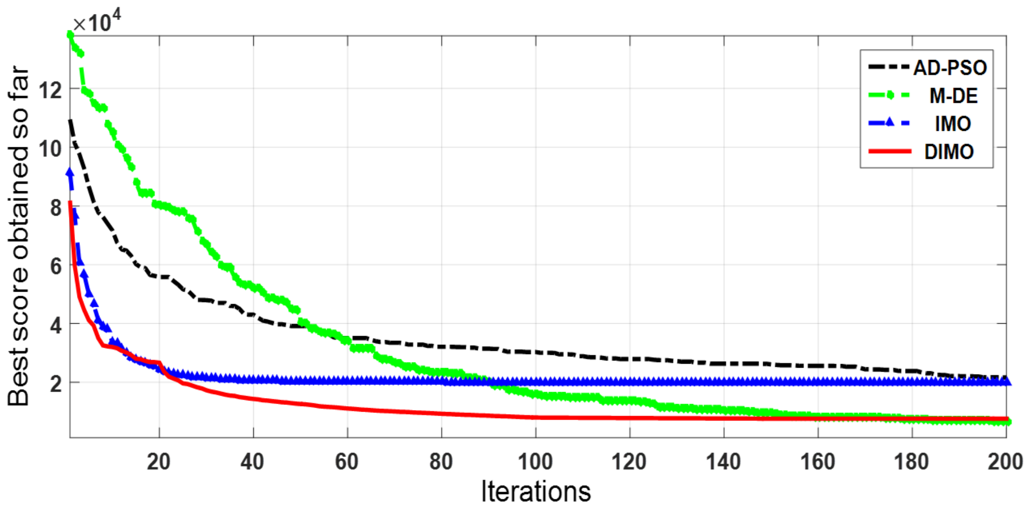

Figure 1. Comparison of convergence curves of DIMO with AD-PSO, M-DE and IMO for first benchmark function

Observably in Fig. 1 and 2, the proposed DIMO algorithm can obtain the optimal fastest in terms of the converging rate.

TABLE 2. The comparative performance of the proposed method of DIMO with the AD-PSO, M-DE, and IMO respectively for the selected test functions

#	Result categories	DIMO	AD-PSO[6]	M-DE[7]	IMO[5]
f_1	Mean best value	3.21E-35	3.96E-01	1.26E+04	3.21E-35
	Standard deviation	1.76E-34	2.09E-01	4.27E+03	1.76E-34
	Optimal value	**0.00E+00**	0.00E+00	5.76E-01	0.00E+00
	Worst value	9.62E-34	1.15E+00	2.46E+04	9.62E-34
f_2	Mean best value	6.44E-35	1.30E+05	1.30E+05	1.30E+05
	Standard deviation	2.77E-34	6.37E+05	6.37E+05	6.37E+05
	Optimal value	**0.00E+00**	1.00E-04	1.00E-04	1.00E-04
	Worst value	1.46E-33	3.49E+06	3.49E+06	3.49E+06
f_3	Mean best value	4.32E-03	6.59E-02	1.27E-02	1.65E-01
	Standard deviation	5.66E-02	2.05E-01	3.40E-01	3.01E-01
	Optimal value	**1.00E-12**	1.40E-06	1.00E-06	1.20E-04
	Worst value	6.73E-01	1.02E+00	9.10E-01	9.88E-01
f_4	Mean best value	1.34E-02	9.39E+00	5.34E-02	1.34E-02
	Standard deviation	1.89E-01	3.97E+01	2.89E-01	1.89E-01
	Optimal value	**1.32E-06**	1.42E-07	1.12E-06	1.32E-06
	Worst value	1.58E+00	2.19E+02	1.38E+00	1.58E+00
f_5	Mean best value	-3.78E+00	-1.78E+00	-2.71E+00	-2.78E+00
	Standard deviation	1.61E-01	1.61E-01	1.19E-01	1.61E-01
	Optimal value	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00
	Worst value	2.95E+00	3.95E+00	3.37E+00	4.95E+00

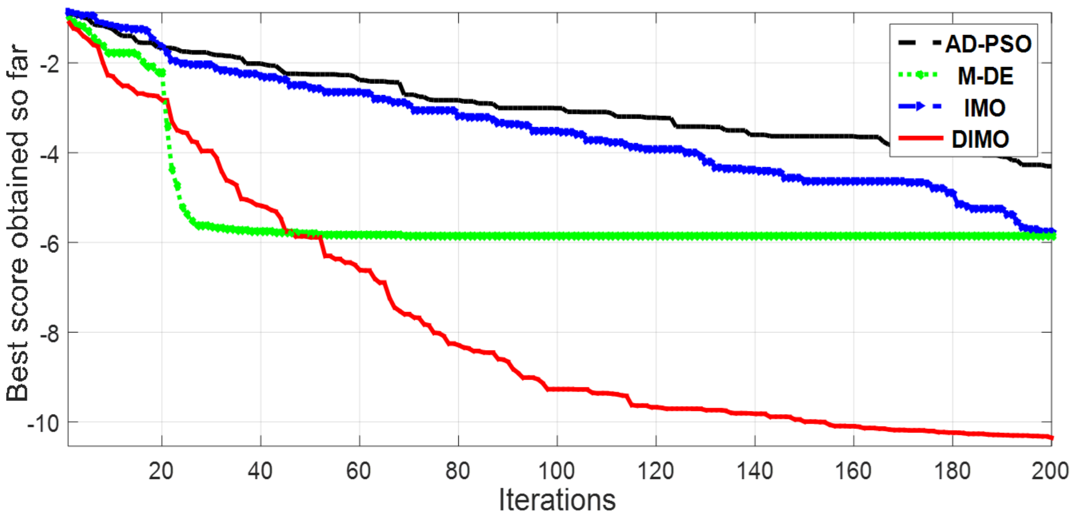

FIGURE 2. Comparison of convergence curves of DIMO with AD-PSO, M-DE and IMO for second benchmark function

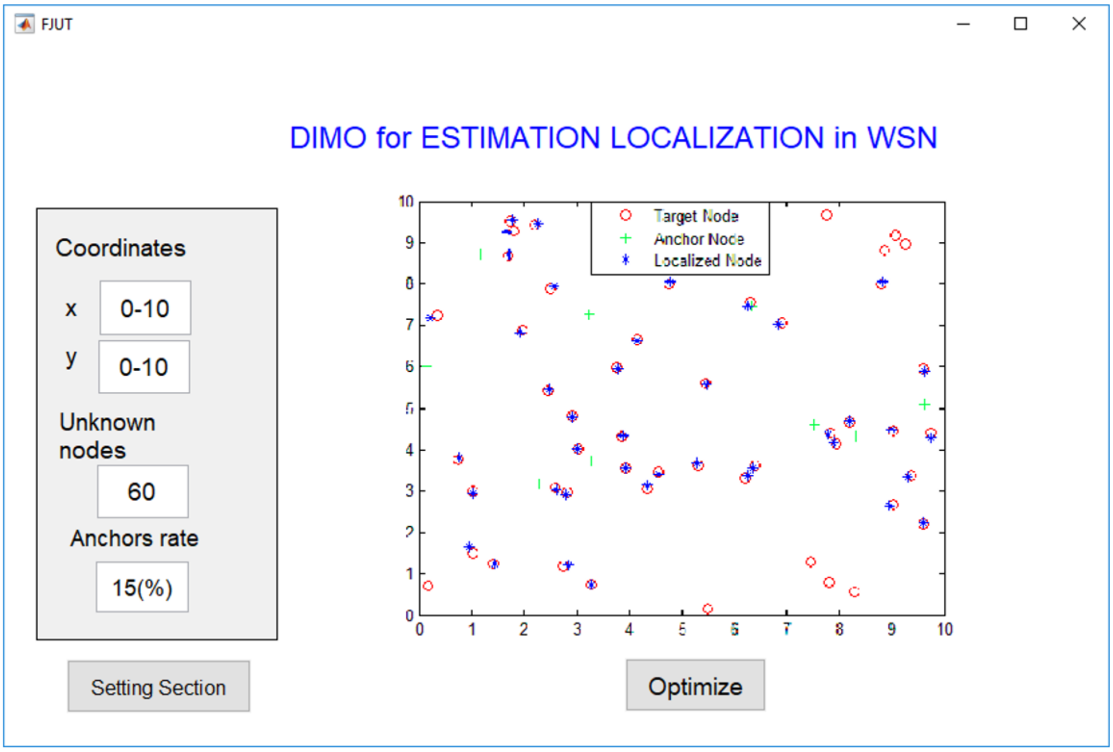

FIGURE 3. A GUI of the DIMO for estimation nodes localization in WSN

5. Apply to Node Localization in WSN. .

5.1. **Model for WSN:.** Let anchors in WSN, unknown nodes would be n-m (where is a number of nodes in the network, $;$). The objective function is to estimate locations of unknown nodes based on referring anchors with the prior information [4]. RSS indicator or signal propagation is used to estimate the nodes positions based on anchor distances. It means that position estimation can be carried out by ranging information. Ranging distance of internodes can be calculated by RSS indicator as ranging technology. The anchor nodes would estimate its distances to neighboring nodes as follows.

$$d_{i,j} = r_{i,j} + n_{i,j} \tag{10}$$

where $d_{i,j}$ is distancing estimation of two nodes i and j being in communicating radius, $r_{i,j}$ is the actual distance and $n_{i,j}$ is a ranging error. The actual distance $r_{i,j}$ is computed as.

$$r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{11}$$

where (x_i, y_i) and (x_j, y_j) are coordination of node i and j location. The neighbor effect factor $\bar{N}_i$ is set to $j \in 1, .., n, j \neq i$ if $r_{ij} \leq R$, and its complement (N_i) is set to $j \in 1, .., n, j \neq i$ if $r_{ij} > R$. Where R is the communication of node i. The measurement noise n_{ij} is the ranging error of RSSI which follows a zero mean Gaussian distribution with variance δ^2 in a random value uniformly distributed. In second phase, the objective function for the space distance constraint can be framed as.

$$f(x,y) = \sum_{i=m+1}^{n} \left(\sum_{j \in N_i} \left(\hat{d}_{i,j} - d_{i,j} \right)^2 \right) \tag{12}$$

where $hatd_{i,j}$ is the estimated distance between nodes i and j, defined by

$$\widehat{d_{i,j}} = \begin{cases} \sqrt{(\widehat{x}_i - x_j)^2 + (\widehat{y}_i - y_j)^2}, \ if\ j\ is\ an\ anchor \\ \sqrt{(\widehat{x}_i - \widehat{x}_j)^2 + (\widehat{y}_i - \widehat{y}_j)^2}, \ otherwise \end{cases} \tag{13}$$

The localization error is minimized by using the optimization algorithm.

5.2. **Simulation Results.** The compared results of the proposed method (DIMO) with the original IMO [4], AD-PSO[6], and Improved Flower pollination algorithm (I-FPA)[4] methods in terms of solution quality and speed taken in objective function evaluations. Figure 3 shows the GUI setting parameters in sensor network and the result of the proposed method applied for estimation nodes localization. The sensor network is supposed

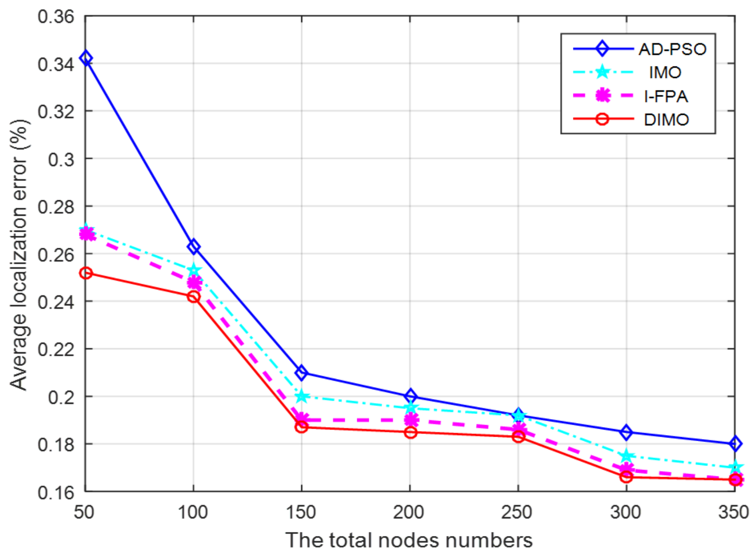

FIGURE 4. The comparison of results localization error of the proposed method with AD-PSO, IMO, and I-FPA

of setting in an area of 10×10 square units (e.g., the meters). The number of sensor nodes is set to 60 that are randomly deployed in the space 2D. The anchors' rate is set to 15%. Assumed the first node's location is known, then we started deploying the sensor nodes. After a sensor node is deployed, its microcontroller collects the RSSI information from previously deployed nodes. The estimation process continues until the deployment of the last sensor node. Distances are simulated corrupted by estimating the distance from anchor node to the target nodes with additive Gaussian white noise e.g. about 2 percentage noise in the distance measurement. Figure 4 shows comparison of the average localization errors of the proposed DIMO with I-FPA, AD-PSO and IMO. Obviously, the proposed DIMO got the localization errors lowest, that means the DIMO provides the good results of the estimation localization in WSN.

6. **Conclusions.** In this paper, a new optimization algorithm based on diversity enhanced population (named DIMO) is presented. Diversity learning strategy and random perturbations were used to modify individual evolutionary information in Ion Motion Optimization (IMO). The experimental results were compared with the recently other methods in the literature for selected test functions and the localization problem in Wireless sensor network (WSN) shows that the proposed DIMO offers an improved IMO in optimization accuracy, convergence speed, and the localization accuracy in WSN.

REFERENCES

[1] J.-S. Pan, T.-T. Nguyen, T.-K. Dao, T.-S. Pan, and S.-C. Chu, Clustering Formation in Wireless Sensor Networks: A Survey, *J. Netw. Intell.*, vol. 02, no. 04, pp. 287 - 309, 2017.

[2] M. Vecchio, R. Lopez-Valcarce, and F. Marcelloni, A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks, Appl. Soft Comput. J., vol. 12, no. 7, pp. 1891 - 1901, 2012.

[3] T. K. Dao, T. S. Pan, and T. T. Nguyen, A Compact Articial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *J. Inf. Hiding Multimed. Signal Process.*, vol. 6, no. 2, pp. 297 - 310, 2015.

[4] J. S. Pan, T.-K. Dao, T. S. Pan, T. T. Nguyen, at el., An Improvement of Flower Pollination Algorithm for Node Localization Optimization in WSN, *J. Inf. Hiding Multimed. Signal Process.,* vol. 08, no. 02, pp. 500 - 509, 2017.

[5] B. Javidya, A. Hatamloua, and S. Mirjalili, Ions motion algorithm for solving optimization problems, *Appl. Soft Comput. J.,* vol. 32, pp. 72 - 79, 2015.

[6] W. Sun, A. Lin, and H. Y. et Al., All-dimension neighborhood based particle swarm optimization with randomly selected neighbors, *Inf. Sci. (Ny).*, vol. 405, no. c, pp. 141 - 156, 2017.

[7] X. Li and at el., Modified differential evolution with self- adaptive parameters method, *J. Comb. Optim.*, vol. 31, no. 2, pp. 546 - 576, 2016.

[8] R. M. Sauter, Elementary Probability With Applications. 2006.

[9] J. J. Liang and P. N. Suganthan, Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search, *IEEE Congr. Evol. Comput.,* vol. 1, no. 1, pp. 522 - 528, 2005.

[10] T.-T. Nguyen, J.-S. Pan, S.-C. Chu, J. F. Roddick, and T.-K. Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *J. Netw. Intell.*, vol. 1, no. 4, p. 130138, 2016.

[11] T. K. Kien Dao, T. S. Pan, T. T. Nguyen at el., A Compact Articial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *J. Inf. Hiding Multimed. Signal Process.* , vol. 06, no. 3, pp. 297310.