

EHPSO: An Enhanced Hybrid Particle Swarm Optimization Algorithm for Internet of Things

<https://doi.org/10.3991/ijoe.v14i06.8305>

Dashe Li^(✉), Dapeng Cheng, Jihong Qin
Shandong Business and Technology University, Yantai, Shandong, China
lidashe@126.com

Shue Liu
Binzhou Medical University, Yantai, Shandong, China

Pingping Liu
Yantai North Tea Promotion Center, Yantai, Shandong, China

Abstract—Internet of Things (IOT) has found broad applications and has drawn more and more attention from researchers. At the same time, IOT also presents many challenges, one of which is node localization, i.e. how to determine the geographical position of each sensor node. Algorithms have been proposed to solve the problem. A popular algorithm is Particle Swarm Optimization (PSO) because it is simple to implement and needs relatively less computation. However, PSO is easily trapped into local optima and gives premature results. In order to improve the PSO algorithm, this paper proposes the EHPSO algorithm based on Novel Particle Swarm Optimization (NPSO) and Hybrid Particle Swarm Optimization (HPSO). The EHPSO algorithm applies the principle of best neighbor of each particle to the HPSO algorithm. Simulation results indicate that EHPSO outperforms HPSO and NPSO in evaluating accurate node positions and improves convergence by avoiding being trapped into local optima.

Keywords—Internet of Things, particle swarm optimization, hybrid

1 Introduction

Currently, IOT plays an increasingly important role in many applications such as environmental monitoring, military surveillance, inventory tracking, and acoustic detection. When an IOT system with a large number of nodes is deployed, usually the nodes are scattered randomly throughout the region of interest. Since there is no a priori knowledge to refer to, an IOT system is ad hoc. However, it is critically important to obtain accurate node positions in most circumstances, especially when emergencies occur. Many measurement techniques and algorithms have been proposed by different scholars to compute the precise node coordinates in 2-dimension and 3-dimension space. For example, a localization method for the Internet of Things

consisting of two phases is introduced in [1]. A novel localization method (NLM) utilizing the relative received signal strength of neighboring nodes to build the fingerprint database was set forth in [2]. A matrix completion algorithm using a nonlinear conjugate gradient method was proposed in [3]. Centralized localization algorithms are suggested in [4,5]. Coordinate system stitching based distributed algorithms were depicted in [6]. Hybrid localization algorithms were discussed in [7]. An error propagation aware localization algorithm was described in [8]. A modified particle swarm optimization algorithm based on gravitational field interactions was proposed in [9]. A neural network approach was implemented in [10]. Algorithms based on PSO or variants were proposed in [11-13]. Combining the advantages of PSO and genetic algorithm (GA), an improved FPSO + FGA hybrid method was provided in [14]. An algorithm using information of the second global best and second personal best particles to improve the search performance of the original PSO was proposed in [15]. A new hybridized version of Particle Swarm Optimization algorithm with variable neighborhood search was proposed to solve the constrained shortest path problem in [16]. The MGBPSO-GSA algorithm (combination of Mean Gbest Particle Swarm Optimization and Gravitational Search Algorithm) was developed in [17].

In this paper, the Enhanced Hybrid Particle Swarm Optimization Algorithm for optimal node locations in Internet of Things is proposed. The proposed EHPSO algorithm can determine accurate node positions and improve convergence by avoiding being trapped into local optima.

The remainder of the paper is organized as follows. Section 2 provides a general description of IoT node localization using standard PSO. An improved approach and its mathematical model are explained in Section 3. Section 3 also provides the pseudo code for the implementation of the improved approach. Simulations are carried out to compare the proposed approach of this paper with other algorithms in Section 4. Conclusion is drawn in Section 5.

2 The Standard Particle Swarm Optimization

The Particle Swarm Algorithm, an evolutionary intelligence algorithm, was first proposed by Kennedy and Eberhart in 1995. Comparing to other algorithms, it has many advantages. For instance, it is not easy to indulge in local optimum but is easy to implement, has high robustness and low requirement for computer hardware.

Considering an M -dimensional search space. N denotes the volume of the swarm population. The position of the i th particle is expressed as a vector $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $i = 1, 2, \dots, N$. The vector $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$, $i = 1, 2, \dots, N$ represents the velocity of the i th particle. Assume that $p_{besti} = (p_{i1}, p_{i2}, \dots, p_{in})$, $i = 1, 2, \dots, N$ is the best position that the i th particle has been searched by now. The optimum of p_{besti} is defined as the best position of the whole swarm which is expressed as $g_{best}(k) = (p_{g1}, p_{g2}, \dots, p_{gd})$. The particles update their positions and velocities according to the following formulas

$$v_{id}(k+1) = \omega v_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) + c_2 r_2 (p_{gd}(k) - x_{id}(k)) \quad (1)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1)$$

where ω is the inertial weight, which is essential to the convergence and usually is chosen as 0.7; c_1 and c_2 are termed as learning factors which determine the influence of cognitive and social scaling coefficients and their recommended values are $c_1 = c_2 = 1.494$. r_1 and r_2 are random numbers uniformly distributed in the interval $[0,1]$.

3 IoT Node Localization Based on NPSO

3.1 Problem Statement

The principals accomplishing IoT node localization utilize the a priori information of anchor nodes to determine the approximate or accurate coordinates of the target nodes. Assume that the coordinate of i th anchor is (x_i, y_i) ; the coordinate of the target node to be determined is (x, y) , then the distance d_i between the i th anchor node and target node can be calculated as

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

Therefore, the coordinates of the target nodes can be calculated as follows:

$$\begin{cases} d_1 = \sqrt{(x-x_1)^2 + (y-y_1)^2} \\ \vdots \\ d_n = \sqrt{(x-x_n)^2 + (y-y_n)^2} \end{cases} \quad (3)$$

The aforementioned equations hardly give a unique answer. PSO is introduced to solve the problem. Suppose that \hat{d}_i is the value of d_i , the fitness function of PSO can be described as follows:

$$f(x, y) = \frac{1}{M} \sum_{i=1}^M (d_i - \hat{d}_i) \quad (4)$$

where M is the number of total anchor targets.

3.2 HPSO

In order to avoid being trapped in a local optimum, HPSO algorithm is suggested. During the process of hybrid, some particles are selected as parent particles in the swarm at fixed proportion in advance. Child particles as many as parent particles in volume are obtained after being charged and muted. If children's fitness function value is lower than parent's, child particles will replace parent particles and then a new swarm is achieved. The whole process can be expressed as the following equations:

$$\begin{cases} \text{child}_1(x_{id}) = r_i \times \text{parent}_1(x_{id}) + (1 - r_i) \times \text{parent}_2(x_{id}) \\ \text{child}_2(x_{id}) = r_i \times \text{parent}_2(x_{id}) + (1 - r_i) \times \text{parent}_1(x_{id}) \end{cases} \quad (5)$$

$$\text{child}_1(v_{id}) = \frac{\text{parent}_1(v_{id}) + \text{parent}_2(v_{id})}{|\text{parent}_1(v_{id}) + \text{parent}_2(v_{id})|} \times |\text{parent}_1(v_{id})|$$

$$\text{child}_2(v_{id}) = \frac{\text{parent}_1(v_{id}) + \text{parent}_2(v_{id})}{|\text{parent}_1(v_{id}) + \text{parent}_2(v_{id})|} \times |\text{parent}_2(v_{id})| \quad (6)$$

where $\text{child}(x_{id}), \text{child}(v_{id})$ are the child's position and velocity ly. $\text{parent}(x_{id}), \text{parent}(v_{id})$ are the parent's position and velocity respectively.

Whereas if children's fitness function value is larger than parent's, that is to say, $f(p_{\text{child}}) < f(p_{\text{parent}})$, HPSO is no longer an efficient approach to solve the local optimum in that circumstance. The following method is provided to solve the problem.

3.3 Position Estimation Using PSO

In the standard PSO, the particle may be trapped into a local optimum solution and not easy for the particle to escape from it because it is only guided by its historical best solution and the global best solution. To solve this problem, an IoT-related HPSO is put forth based on NPSO proposed in [13].

According to the above discussion, the best positions of each particle and the best positions of whole swarms are critical for the particle's movement in the search space. Especially, in some certain circumstance, the best neighbor or the neighbor's neighbor may provide much better information.

Definition: the neighbors of the i th particle are identified by the mean Euclidean Metric between the i th particle and the rest of the particles. Assume that $d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ is the Euclidean Metric between the i th particle and the j th particle, then md_i can be calculated as follows,

$$md_i = \frac{\sum_{j=1}^M d_{ij}}{M-1} \quad (7)$$

Thus, if $md_i < r$, we can define the j th particle as the neighbor particle. After the best neighbor p_{Ni}^{best} has been chosen, the following formulas are utilized to guide the movement of each particle. If $f(p_{Ni}^{\text{best}}) < f(p_i)$, then

$$\begin{cases} \text{child}(v_{id}(k+1)) = \omega \text{parent}(v_{id}(k)) + c_1 r_1 (\text{parent}(p_{id}(k)) - \text{parent}(x_{id}(k))) + \\ \quad c_2 r_2 (\text{parent}(p_{Ni}^{\text{best}}(k)) - \text{parent}(x_{id}(k))) \\ \text{child}(x_{id}(k+1)) = \text{parent}(x_{id}(k)) + \text{child}(v_{id}(k+1)) \end{cases} \quad (8)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1)$$

If $md_i > r$ or the particle is trapped into a local optimum solution, we will redefine the neighbor as follows,

$$\text{smd}_i = \frac{\sum_{j=1}^M d_{ij}}{M-2} < r$$

Otherwise,

$$\begin{cases} v_{id}(k+1) = \omega v_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) + c_2 r_2 (p_{gd}(k) - x_{id}(k)) \\ x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \end{cases} \quad (9)$$

3.4 Pseudo-Code

Based on the aforementioned descriptions, the pseudo-code of the EHPSO algorithm are provided as follows:

- (1) Initialization: A series of parameters should be initialized in advance, including a population of M particles with random position x_i within the transmission range, the maximum iteration, random best position of the i th particle p_{besti} and the best position of the whole swarm $g_{best}(k)$.
- (2) The initial velocity is set to zero; ω_{min} , ω_{max} , v_{min} , v_{max} are set to fixed values;
- (3) Set $p_i = x_i$ ($i = 1, \dots, SN$) and find p_g ;
- (4) while $f(pchild) < f(pparent)$ do
- (5) Update the particle velocity by equations (6)
- (6) Update the particle position by equations (5)
- (7) while $f(pchild) > f(pparent)$ do
- (8) if $md_i < r$ do
- (9) Update the particle velocity by equations (8)
- (10) Update the particle position by equations (1)
- (11) while $md_i > r$ and $\text{smd}_i < r$ do
- (12) Update the particle velocity by equations (9)
- (13) Update the particle position by equations (1)

4 Simulation Results and Discussion

The IoT system with 15 target nodes and 5 anchor nodes is randomly deployed in a $20m \times 20m$ open area where the communication between nodes will not be disturbed by some other factors. The dimension of the EHPSO is fixed to be 50. When the precision of the fitness function attains 10^{-6} or the maximum iteration number approaches 100, the simulation will be terminated. According to [18], the cognitive parameters $c_1=c_2$ and inertia constant ω are set to be 1.494 and 0.6 respectively.

MatlabTM 13.0 is utilized to analyze the performance of EHPSO comparing to other algorithms. Simulation platform which carries out the simulation is a laptop with Intel Core I7, 16G memory and Windows 7.

4.1 Localization Error

The aim to improve localization algorithms is to lower the localization error, which is one of the most important criterion to localization algorithm. By altering the loss index n , the amount of swarm N and the iteration number, some experiments are carried out to test the advantages of EHPSO over other localization algorithms.

As shown in Figure 1, with the value of pass loss index n increasing, the localization error will vary correspondingly. In the rank of PSO, QJ-CMPSO and NPSO, the localization error value of PSO is the largest and EHPSO's is lower than QJ-CMPSO's at the same pass loss index.

In Figure 2, it dedicates the relationship between swarm amount and localization error when the parameters are fixed as $n=2.5$ and N varying from 10 to 60. Comparing the three approaches, the localization error is shrinking with the development of swarm amount approximately. At the same swarm amount, the value of EHPSO's localization error is the lowest and the value of PSO is the highest.

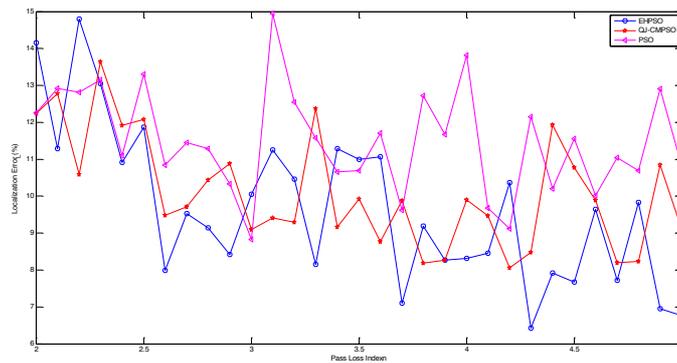


Fig. 1. Relationship between pass loss index and localization error

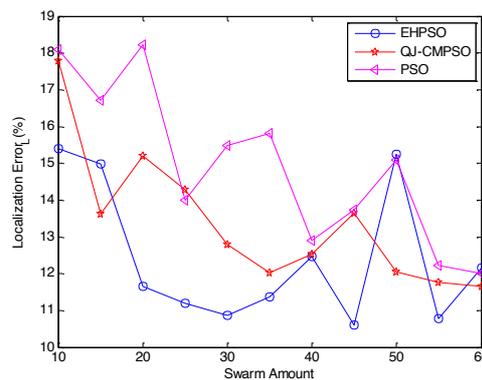


Fig. 2. Relationship between swarm amount and localization error

4.2 Algorithm Convergence

The fitness function can be viewed as the localization absolute error of sensor nodes in an IoT system, whose convergence value can be used to assess the localization accuracy and algorithm convergence.

From Figure 5, it can be deduced that all the three algorithms PSO, QJ-CMPSO and EHPSO have convergence attribute, however EHPSO's convergent velocity is fastest among the three algorithms, QJ-CMPSO is the second fast.

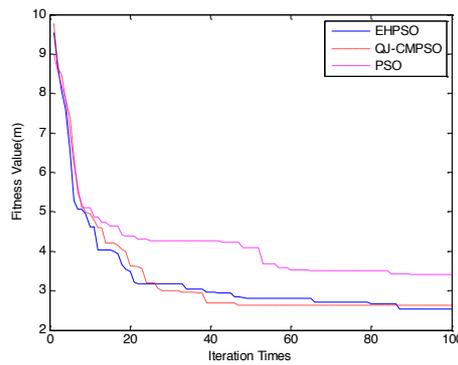


Fig. 3. Relationship between iteration times and fitness value

4.3 Localization Performance

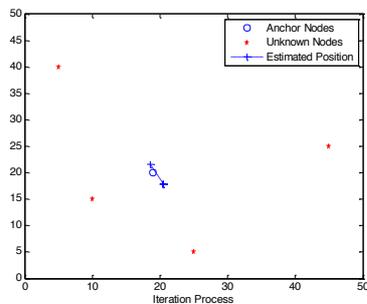


Fig.4. Iteration process

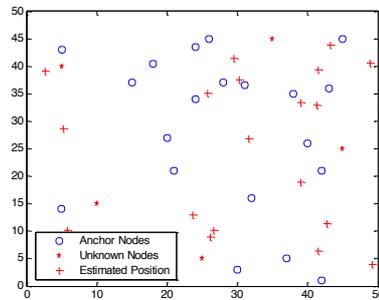


Fig.5. Localization performance

Figure 4 shows the process of approaching global extrema during iterating. The estimated node positions approach the real positions gradually and the estimated errors become smaller with the iteration times increasing.

Figure 5 shows what is achieved after an one-time localization of 20 unknown nodes, where the parameters are set as $n=2.5, N=40, T_{max}=100$. After 10 iterations, 16 active nodes are observed and the average error is 6.47%.

5 Conclusion

Based on RSSI, after defining the best neighbor, the EHPSO algorithm has been proposed to determine the localization of nodes of an IoT system quickly. The pseudo-code of the EHPSO algorithm is provided and simulations have been carried out. The analysis of the simulation results leads us to the conclusion that EHPSO is a good alternative because it is simpler to implement than the PSO algorithm or the QJ-CMPSO algorithm. The EHPSO algorithm also outperforms the standard PSO algorithm and the QJ-CMPSO algorithm in many aspects.

6 Acknowledgement

This work is partially supported by National Natural Science Foundation (No.61070175), Shandong Province Natural Science Foundation (ZR2013FL017, ZR2013FL018), and Project Development Plan of Science and Technology of Yantai (2015ZH062, 2015ZH062). The authors gratefully acknowledge the suggestions of the reviewers which have helped improve the presentation.

7 References:

- [1] Zhikui Chen, Feng Xia, Tao Huang, Fanyu Bu, Haozhe Wang(2013). A localization method for the Internet of Things. *The Journal of Supercomputing*, 63: 657–674 <https://doi.org/10.1007/s11227-011-0693-2>
- [2] Kai Lin, Min Chen, Jing Deng, Mohammad Mehedi Hassan , Giancarlo Fortino(2016). Enhanced Fingerprinting and Trajectory Prediction for IOT Localization in Smart Buildings. *IEEE Transactions on Automation Science and Engineering*, 13:1294-1307 <https://doi.org/10.1109/TASE.2016.2543242>
- [3] Luong Nguyen, Sangtae Kim, Byonghyo Shim(2017). Localization in Internet of Things network: Matrix completion approach. *Information Theory and Applications Workshop*, La Jolla, CA, USA.
- [4] A. Pal(2010) . Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges. *Network Protocols and Algorithms*, 2:45-73. <https://doi.org/10.5296/npa.v2i1.279>
- [5] A.A. Kannan , Guoqiang Mao , B. Vucetic(2006). Simulated Annealing based Wireless Sensor Network Localization. *Journal of Computers*, 2:15-22. <https://doi.org/10.4304/jcp.1.2.15-22>
- [6] D. Moore, J. Leonard, D. Rus. Robust distributed network localization with noisy range measurements. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems(2004)*, November 3-5 2004, Baltimore, MD, pp. 50-61. <https://doi.org/10.1145/1031495.1031502>
- [7] King-Yip Cheng, King-Shan Lui, Vincent Tam. Localization in Sensor Networks with Limited Number of Anchors and Clustered Placemen. In *Proceedings of Wireless Communications and Networking Conference(2007)*, March 11-15 2007, Hong Kong, China, pp. 4425–4429.
- [8] N. A. Alsindi, K. Pahlavan, B. Alavi. An Error Propagation Aware Algorithm for Precise Cooperative Indoor Localization. In *Proceedings of IEEE Military Communications Con-*

- ference(2006),November 12-15 2006,Washington, DC, USA, pp. 1-7, <https://doi.org/10.1109/MILCOM.2006.302311>
- [9] M. Spichakova(2016). Modified particle swarm optimization algorithm based on gravitational field interactions. Proceedings of the Estonian Academy of Sciences,65:15–27. <https://doi.org/10.3176/proc.2016.1.01>
- [10] Shiu Kumar, Ronesh Sharma(2016). Localization for Wireless Sensor Networks: A Neural Network Approach. International Journal of Computer Networks & Communications, 8:61-71 <https://doi.org/10.5121/ijcnc.2016.8105>
- [11] A. Gopakumar , Lillykutty Jacob. Localization in wireless sensor networks using particle swarm optimization. IET International Conference on Wireless, Mobile and Multimedia Networks(2008), February 11-13 2008, Beijing,China,pp. 227-230
- [12] Xinyi Hu, Shuo Shi, and Xuemai Gu(2012). An Improved Particle Swarm Optimization Algorithm for Wireless Sensor Networks Localization. Journal of Experimental Biology, 1:1-4 <https://doi.org/10.1109/WiCOM.2012.6478418>
- [13] Chun-Feng Wang, Kui Liu(2016). A Novel Particle Swarm Optimization Algorithm for Global Optimization. Computational Intelligence and Neuroscience, 2016:1-10 <https://doi.org/10.1155/2016/9482073>
- [14] F. Valdez, P. Melin, O. Castillo(2014). Modular Neural Networks architecture optimization with a new nature inspired method using a fuzzy combination of Particle Swarm Optimization and Genetic Algorithms, Information Sciences, 270:143–153. <https://doi.org/10.1016/j.ins.2014.02.091>
- [15] Y.B. Shin , E. Kita(2014). Search performance improvement of particle swarm optimization by second best particle information. Applied Mathematics and Computation,246: 346–354. <https://doi.org/10.1016/j.amc.2014.08.013>
- [16] Y. Marinakis, A. Migdaldas, A. Sifaleras(2017). A hybrid Particle Swarm Optimization – Variable Neighborhood Search algorithm for Constrained Shortest Path problems. European Journal of Operational Research,261:819-834 <https://doi.org/10.1016/j.ejor.2017.03.031>
- [17] N. Singh, S. Singh, S B Singh(2017). A New Hybrid MGBPSO-GSA Variant for Improving Function Optimization Solution in Search Space. Evolutionary Bioinformatics, 13:1-13. <https://doi.org/10.1177/1176934317699855>
- [18] R.C. Eberhart, Yuhui Shi. Tracking and optimizing dynamic systems with particle swarms. Proceedings of the 2001 Congress on Evolutionary Computation(2001),May 27-30, Seoul, South Korea,pp. 94-100 <https://doi.org/10.1109/CEC.2001.934376>

8 Authors

Dashe Li, Dapeng Cheng, and Jihong Qin are with the College of Computer of Science and Technology at Shandong Business and Technology University, Yantai, China.

Shue Liu is the researcher of Binzhou Medical University, Yantai, China.

Pingping Liu is with the Yantai North Tea Promotion Center, Yantai, Shandong, China.

Article submitted 25 January 2018. Resubmitted 20 February 2018 . Final acceptance 13 May 2018. Final version published as submitted by the authors.