A New Distributed Localization Algorithm Using Social Learning based Particle Swarm Optimization for Internet of Things

Ashish Rauniyar*, Paal Engelstad†, Jonas Moen‡
*†Autonomous Systems and Networks (ASN) Research Group, Department of Computer Science,
OsloMet - Oslo Metropolitan University, Norway
*†‡Autonomous Systems and Sensor Technologies Research Group, Department of Technology Systems, University of Oslo, Norway
†‡FFI, Division for Cyber Systems and Electronic Warfare, Norwegian Defense Research Establishment, Norway
Email: (*ashish.rauniyar,†paal.engelstad)@hioa.no, ‡jonas.moen@ffi.no

***Abstract*—Emerging applications in the Internet of Things (IoT) will depend on the accurate location of thousands of deployed sensors. However, accurate localization of deployed sensors nodes is a classical optimization problem which falls under NP-hard class of problems. Therefore in this work, we propose a new distributed localization algorithm using social learning based particle swarm optimization (SL-PSO) for IoT. With SL-PSO algorithm, we aim to do precise localization of deployed sensor nodes and reduce the computational complexity which will further enhance the lifetime of these resource constrained IoT sensor nodes. Extensive simulations are carried out to show the effective performance of SL-PSO algorithm in accurate localization. Experimental results depict that SL-PSO algorithm can not only increase convergence rate but also significantly reduce average localization error compared to traditional particle swarm optimization (PSO) and its other variants.**

I. Introduction

With the plethora of wireless devices, we are moving towards the connected world which is the guiding principle for the internet of things (IoT). IoT is a network of physical objects such as sensors which are further embedded with software, electronics and network connectivity that allows these physical entities to collect and exchange data between them [1][2]. There are numerous applications of IoT such as routing, target tracking, monitoring homes, cities, automation, health monitoring, transportation management and environment [3]. All these applications of IoT are possible due to the deployment of sensor nodes which continuously monitor the surrounding environment and entities, collect and send sensed data according to the application requirement in IoT. Accurate localization of sensor nodes is the prerequisite to run these emerging applications of IoT as it is staggeringly difficult to differentiate sensed data and employ sensing information of the nodes without location information [4]. That's why it is often believed that sensing without localization is meaningless. Besides, accurate localization of sensor nodes also help to tackle problem such as geographic routing [5], intrusion detection [6], traffic monitoring [7] and so on. However, accurate localization of deployed sensors nodes is a classical optimization problem which falls under NP-hard class of problems [8].

Using conventional ways, each sensor nodes can be localized by navigation system such as Global Positioning System (GPS) [9]. Further, it would be difficult to get the location information of the sensor nodes if it is deployed in urban or indoor environment where the satellite signal may be severely affected or blocked [10]. Moreover considering the size, cost and power consumption constraint of GPS receivers make it impossible for it's use on each resource constrained sensor nodes. It should be noted that these sensor nodes are battery operated and are deployed in a random fashion over a area or region. Therefore, conserving the energy consumption of these sensor nodes will enhance their lifetime operation [11].

An unknown IoT sensor node's location (X, Y) can be estimated if it is in the communication range of at least three anchor nodes which have a priori knowledge about their location information as (X_1, Y_1), (X_2, Y_2), (X_3, Y_3) respectively [12]. The process of localization consists of two phases namely ranging phase and estimation phase. In the ranging phase, an unknown sensor node estimate their distance based on Received Signal Strength Identification (RSSI), Time of Arrival (TOA) of received signal, Time Difference of Arrival (TDOA) of received signal [13] etc. The results obtained during the ranging phase is affected by the noise factor and thus likely to be inaccurate [14]. In the estimation phase, the position of an unknown sensor node is calculated using the ranging information from the first phase. This can be done either by using conventional mathematical optimization algorithms such as solving a set of simultaneous equations or by using stochastic optimization algorithms that minimizes the localization error. The focus of this work is on bio-inspired stochastic optimization algorithms. For clear representation of the localization of sensor nodes in the context of IoT, we have clearly shown it in Fig. 1.

Many localization algorithms have been proposed for the wireless sensor networks (WSN) to surmount the localization accuracy and increase lifetime of wireless sensor nodes and

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

have been documented in the literature [15][16]. A WSN node localization based on bio-inspired PSO has been proposed in [17][18]. However, PSO is likely to get trapped in local minima of the optimization problem. A brief introduction to PSO will be presented in the next section. Different variants of the PSO algorithm has been widely researched and proposed in the literature. A distributed localization for WSN using binary PSO (BPSO) has been proposed in [19]. The authors showed the fast computation of the BPSO algorithm on the WSN sensor node localization process at the expense of increased localization error. A distributed localization of WSN node based on differential evolution approach has been proposed in [20]. The authors demonstrated results for different scenarios delimited by walls and tested with inner obstacles to obtain a suboptimal solution. A recursive shortest path routing algorithm with it's application in WSN localization has been proposed in [21]. Their proposed recursive shortest path routing algorithm is capable of estimating the shortest distance between two non-neighbouring sensors in multi-hop wsn. A localization scheme for IoT has been proposed in [4]. The authors proposed scheme consists of two phases namely the partition phase and localization refinement phase. In partition phase, the target region is first divided into small grids. Then in localization refinement phase a higher accuracy of localization can be obtained by applying a compact algorithm which can easily implement two-dimensional plane localization with a regular deployment of reference nodes. An effective bio-inspired cuckoo search algorithm for sensor node localization in WSN has been proposed in [22]. The author showed the effective performance of cuckoo search algorithm on reducing average localization error and increasing convergence rate. Bio-inspired algorithms are known to be computationally efficient algorithms and are widely used for solving optimization problems. Out of all proposed bio-inspired algorithms in the literature so far, PSO is widely chosen optimization algorithm because of its simplicity and ease of implementation. An interesting new variant of the PSO algorithm inspired from social behaviour found in animals, which they called 'SL-PSO' has been proposed in [23]. The authors showed the superior performance of the 'SL-PSO' algorithm in solving scalable optimization problems. This 'SL-PSO' algorithm serves as the basis for our proposed localization algorithm in IoT as described in more detail in the next section.

The main contribution of this paper can be outlined as:

1) We first surveyed the bio-inspired algorithms, such as PSO and its variants BPSO and Modified BPSO algorithm, to tackle the distributed localization issue in IoT.
2) We then formulate and propose a new distributed SL-PSO localization algorithm and show its superiority over PSO and its variants.
3) We have performed extensive simulation to verify that the SL-PSO algorithm significantly reduces average localization error and that it is computationally more efficient than PSO and its variants.

The rest of the paper is organized as follows. Section II gives a brief survey of the PSO, BPSO, Modified BPSO and SL-PSO algorithms. In Section III, we explalin our proposed SL-PSO localization algorithm for IoT. Simulations and performance evaluation of the proposed SL-PSO localization algorithm and comparison with PSO, BPSO, and Modified BPSO localization algorithms is carried out in Section IV. Conclusion is drawn in Section V.

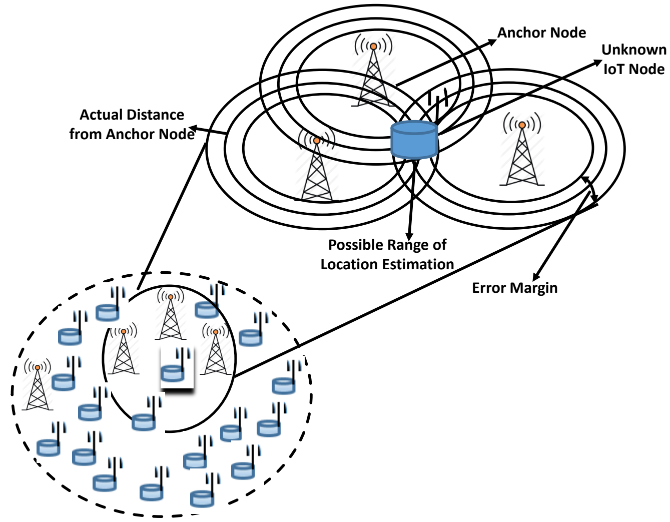

Fig. 1. Illustration of Localization of Sensor Nodes in Internet of Things

II. Standard Bio-inspired Algorithms

This section gives a the brief survey of standard bio-inspired algorithms like PSO, BPSO, Modified BPSO and SL-PSO.

A. *Particle Swarm Optimization (PSO)*

PSO is a widely used population based stochastic optimization algorithm developed by Kennedy and Eberhart in 1995 [24]. PSO is a bio-inspired algorithm, which takes the inspiration from social behaviour of bird flocking or fish schooling. It should be noted that unlike other genetic algorithms, there are no evolution operators such as crossover or mutation, and there are only a few paramenters to adjust, which makes PSO algorithm easy to implement. The birds or fishes representing particles in PSO algorithm fly through the problem space by learning from the current optimum particle to find the optimum value of the cost or objective function. Here, the cost or objective function represents the food for birds or fishes that needs to be found in search/problem space. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating the generations. In PSO, each particles is able to memorize the best position known as the global best or G_{best} found by the whole swarm in history, and the best position known as the personal best or P_{best} that has been found by each particle. The global optimum solution of the optimization problem is found by particles learning from the G_{best} and P_{best} positions. The learning process of the standard PSO algorithm can be represented by the following two equations:

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

$$V_{id}(t+1) = \omega V_{id}(t) + c_1 r_1(t)(P_{best_{id}}(t) - X_{id}(t)) + c_2 r_2(t)(G_{best_d}(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

where ω is inertia weight, t is the iteration or generation number, $V_{id}(t)$ and $X_{id}(t)$ are the velocity and position of i^{th} particle respectively, c_1, c_2 are the constant weight factors known as acceleration coefficients, $r_1(t)$, $r_2(t)$ are the two randomly generated vectors in the [0, 1] interval, $P_{best_{id}}(t)$ is the personal best of the i^{th} particle and $G_{best_d}(t)$ is the global best of the swarm.

There are other variants of PSO such as BPSO and Modified BPSO which will be explained in the next subsection. It is interesting to note that in PSO, a particle only learns through the personal best P_{best} solution and the global best G_{best} solution to solve the optimization problem.

B. Binary Particle Swarm Optimization (BPSO)

Simplifying the PSO algorithm was originally suggested by Kennedy and has been studied extensively in the literature to increase the optimization performance, such as the accuracy and convergence rate. BPSO is another variant of the PSO algorithm that searches the solution to the optimization problem in a binary discrete search space [19]. In BPSO, a sigmoid transformation is applied to the velocity component of the particles, which forces to take the value between '0' and '1' and squashes the particles position component values to be either 0's or 1's. Basically, BPSO is a slight modification of the PSO algorithm which differ by the velocity and particle positions update equation as follows:

First, the velocity is updated by replacing $V_{id}(t)$ with $sigmoid(V_{id}(t))$ in Eq. (1) of PSO as

$$sigmoid(V_{id}(t)) = \frac{1}{1 + e^{-V_{id}(t)}} \quad (3)$$

Second, the position of each particle is updated by using:

$$X_{id}(t) = \begin{cases} 1, \text{ if } & rand < sigmoid(V_{id}(t)) \\ & 0, \text{otherwise} \end{cases} \quad (4)$$

C. Modified Binary Particle Swarm Optimization (Modified BPSO)

Modified BPSO is another variant of the PSO algorithm. It is a slight modification of the basic BPSO algorithm. Here, all components remain the same except for the sigmoid transformation function, which is changed as [25]:

$$sigmoid'(V_{id}(t)) = 2 \times |sigmoid(V_{id}(t)) - 0.5| \quad (5)$$

$$X_{id}(t) = \begin{cases} 1, \text{ if } & rand < sigmoid'(v_{id}(t)) \\ & 0, \text{otherwise} \end{cases} \quad (6)$$

Algorithm 1 SL-PSO Algorithm Pseudocode

Initialize ω, c_1, c_2 and maximum number of iterations t_{max}
Initialize the objective/cost function $f(x, y)$
Initialize X_{min}, X_{max}, V_{min} and V_{max}
Generate the particles for SL-PSO i.e., Total Number of particles = N
for each particle i.e., particle i do
for each dimension d, do
Initialize learning probability of particles $P_{id}^L = (1 - (i - 1)/N)^{\alpha.log\lceil\frac{d}{N}\rceil}$
Initialize X_{id} randomly: $X_{min} \leq X_{id} \leq X_{max}$
Initialize V_{id} randomly: $V_{min} \leq V_{id} \leq V_{max}$
end for
end for
Iteration $t = 1$
while $(t \leq t_{max})$ do
for each particle i do
calculate cost/objective function $f(x_i, y_i)$
end for
/* Cost/Objective Function Sorting and Ranking */
Do sorting and ranking of cost/objective function $f(x, y)$ for each particles in increasing order of fitness evaluation
/* Behaviour Learning From All Best Particles */
for each particle i do
for each dimension d, do
$p_{id}(t) = rand(0, 1)$;
if $p_{id}(t) \leq P_{id}^L$ then
calculate velocity $V_{id}(t + 1)$ using Eq. (8) & (9)
determine position $X_{id}(t + 1)$ using Eq. (7)
restrict X_{id} to $X_{min} \leq X_{id} \leq X_{max}$
end for
end for
$t = t + 1$
end while

D. Social Learning based Particle Swarm Optimization (SL-PSO)

In the PSO algorithm, as we can see from Eq. (1) and Eq. (2), the learning process of the particle is updated through the P_{best} and G_{best} components only. However, there may be other better particles in the generated particles population through which a particle can further learn and enhance its learning process to find the optimum solution to an optimization problem. The process of learning and imitating the behaviour of better individuals in a population is known as social learning, which can be widely discovered in social animals. Motivated by this social learning mechanism, reference [23] proposed an interesting and different variant of PSO known as the SL-PSO algorithm in which a particle is able to perform social learning i.e, learning and imitating the behaviours of any better particles or individuals in the population. The authors also claimed that their work is one of the first attempts of its kind to apply a social learning process

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

to meta-heuristic stochastic optimization algorithm like PSO. Without the burden of having individual trial and error by a particle to find the optimum solution to an optimization problem, social learning helps the particles to learn from any better particles. This speeds up the learning rates of the individual particles. Unlike PSO and its variants - BPSO and Modified-BPSO where the particles only learn from historical P_{best} and G_{best} positions, SL-PSO is executed on a sorted swarm where a particle in the current swarm can learn and imitate the behaviour of any better particles known as demonstrator. Imitators are the particles that learn or imitate the behaviours of the demonstrators in the current swarm. A pseudocode of the SL-PSO algorithm is given in Algorithm 1. In the SL-PSO algorithm, suppose the sorted swarm according to cost/fitness evaluation and ranking is $Particle_1, ...Particle_i, Particle_{i+1}, ...Particle_N$, where $Particle_N$ is the best particle which will never be an imitator and it will not be updated and $Particle_1$ is the worst particle which will never act as demonstrator. For $Particle_i$, any $Particle_j$ which satisfies the constraint $i < j \leq N$ can be it's demonstrators i.e., in sorted swarm, for $Particle_i$, $Particle_{i+1}, Particle_{i+2}....Particle_N$ will serve as demonstrator. The social learning process of SL-PSO can be updated through the following equations [23]:

$$X_{id}(t+1) = \begin{cases} X_{id}(t) + V_{id}(t+1), & \text{if} \quad p_{id}(t) \leq P_{id}^L \\ X_{id}(t), \text{otherwise} \end{cases} \tag{7}$$

$$V_{id}(t+1) = \omega r_1(t) V_{id}(t) + r_2(t) D_{id}(t) + c_3 r_3(t) A_{id}(t) \tag{8}$$

where

$$\begin{cases} D_{id}(t) = X_{jd} - X_{id}, \\ A_{id}(t) = \bar{X}_{id}(t) - X_{id}(t) \end{cases} \tag{9}$$

where $c_3 = (d/N) * 0.01$ is the constant weight factors known as acceleration coefficients, $r_1(t)$, $r_2(t)$ and $r_3(t)$ are the randomly generated vectors in [0, 1]. From the SL-PSO velocity Equation (8), we can see that the velocity update equation consists of three different components. The first component is same as the inertia component of general PSO algorithm. The real difference lies in the second and third component. As already discussed, instead of only learning from P_{best} like in the general PSO algorithm, a $particle\ i$ in the SL-PSO algorithm learns from any of its demonstrators which is denoted as D_{id} in the second component of Equation (8). The behaviour vector of X_{id} corrects its behaviour by learning from its demonstrator X_{jd} as shown in Equation (9). Also in the third component of Equation (8), we can see that, instead of learning only from the G_{best}, a $particle\ i$ learns from the mean behaviour of all particles, denoted as A_{id} in the third component of Equation (8). The behaviour vector of X_{id} correct its behaviour by learning from the collective mean behaviour of all the particles A_{id}. In A_{id} as shown in Equation (9), the mean behaviour of all particles in a swarm is represented as $\bar{X}_{id} = \frac{\sum_{i=1}^{N} X_i^d}{N}$. Also, as shown in Equation (7), P_{id}^L is imitiating/learning probability. If a particle in the generated swarm has better fitness/objective function value then it is less likely to learn from other individuals. The $particle_i$ will correct its behaviour only if randomly generated probability p_i satisfies the constaint $0 \leq p_{id}(t) \leq P_{id}^L = (1 - \frac{i-1}{N})^{\alpha.log\lceil \frac{d}{N} \rceil} \leq 1$ with $\alpha.log()$ being used as smoothen factor.

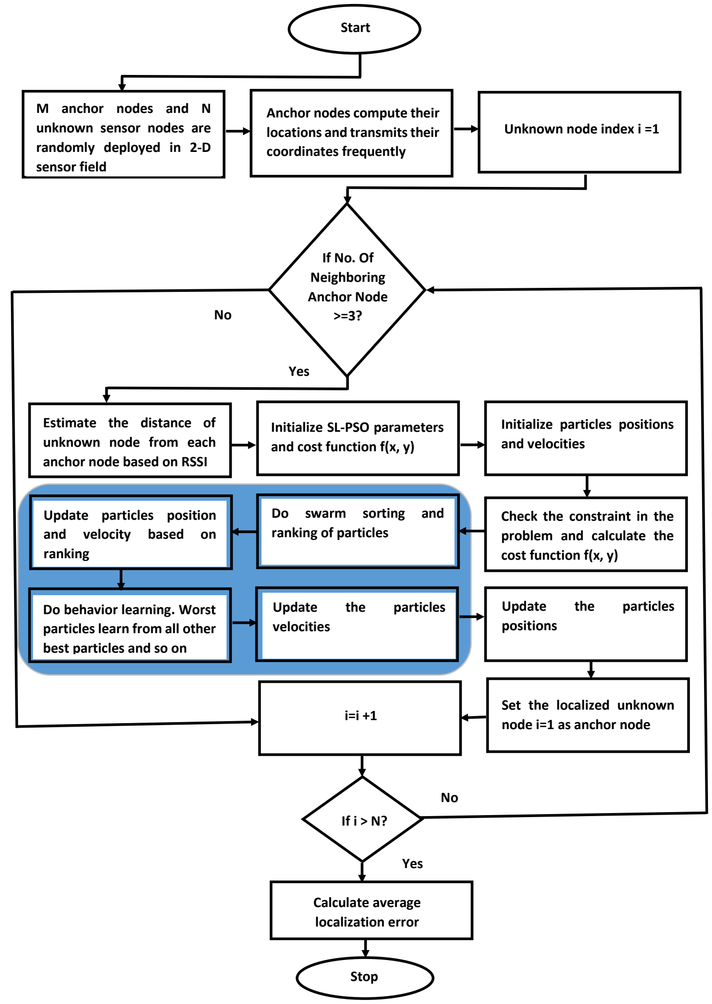

Fig. 2. Procedural Flow of the SL-PSO Algorithm for Localization of Sensor Nodes in Internet of Things

III. Distributed IoT Sensor Node Localization Process based on SL-PSO Algorithm

In this paper, the main objective of the sensor node localization in IoT is to accurately estimate the unknown coordinates of N IoT sensor nodes based on M stationary anchor nodes in two-dimension (2-D) sensor field i.e., $d = 2$. The procedural flow of distributed IoT sensor node localization process based on SL-PSO algorithm is shown in Fig. 2. As shown in Fig. 2., the distributed localization process of the IoT sensor nodes mainly involves the following steps:

1) M anchor nodes and N unknown IoT sensor nodes are randomly deployed in 2-D sensor field with R being the

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

communication range for each sensor.

2) The anchor nodes are aware of their location coordinates and transmits/broadcasts their location coordinate information frequently.
3) An unknown i^{th} IoT sensor node is considered localizable if it is in the communication range of three or more anchor nodes. Otherwise, the node is not localizable. Considering the simple implementation, hardware complexity and low cost for the deployment of the sensors, in this paper, we have assumed RSSI based ranging approach to estimate the distance between neighbouring nodes. RSSI signal can be measured without additional energy consumption which will further benefit the resource constrained IoT sensor nodes. Usually RSSI based distance estimation approaches have ranging error which follows a zero-mean Gaussian distribution with variance σ^2. If d_{nm} is the actual distance between n^{th} unknown sensor node whose location coordinate is (x_n, y_n) and m^{th} anchor node whose location coordinate is (x_m, y_m), then the distance d_{nm} which is also known as Euclidean distance is given as:
$$d_{nm} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} \tag{10}$$
As discussed above, due to ranging error i.e., environmental effect, the actual distance d_{nm} is usually different from the measured distance $\hat{d}_{nm}$. We have considered the environmental effect as a Gaussian noise P_n (noise percentage). Therefore, $\hat{d}_{nm}$ = d_{nm} + P_n.
4) Let us define the cost/objective function. The mean of square of ranging/localization error between anchor nodes and unknown sensor node is formulated as the cost/objective function $f(x_n, y_n)$ which is given as:
$$f(x_n, y_n) = \frac{1}{M}\sum_{m=1}^{M}(\sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} - \hat{d}_{nm})^2 \tag{11}$$
where $M \geq 3$ is the number of anchor nodes within communication range, an n^{th} unknown sensor node can estimate its location coordinate (x_n, y_n) by running the SL-PSO algorithm which minimizes the cost/objective function $f(x_n, y_n)$.
5) At the end of each SL-PSO algorithm iteration, an unknown node that gets localized will act as an additional anchors for other unknown sensor nodes in the next iteration.
6) Step 2 to Step 4 are executed repeatedly until all the unknown IoT sensor nodes have been localized or the termination conditions have been reached.
7) Calculating the average localization error E_{ALE}. If (X_n, Y_n) is the actual unknown sensor node location and (x_n, y_n) is the computed location through SL-PSO algorithm then the E_{ALE} can be given as:
$$E_{ALE} = \frac{\sum_{n=1}^{N}\sqrt{(X_n - x_n)^2 + (Y_n - y_n)^2}}{N} \tag{12}$$
It should be noted that the main objective of the localization algorithm is to reduce E_{ALE} to have a better localization performance.

IV. Simulation Results and Performance Evaluation

TABLE I
Simulation Parameters

Parameter	Value
Sensor Field Size	100 x 100 m^2
Anchor Nodes, M	10
Unknown Nodes, N	50
Transmission Range, R	25 m
Noise Percentage, P_n	2
Maximum Iteration, t_{max}	150
Inertia weight, ω	0.7
Acceleration constant, c_1, c_2	2.0
Acceleration constant, c_3	0.002
Random numbers, r_1, r_2, r_3	[0,1]
Particle positions	$x_{\min}$ = 0, $x_{\max}$ = 100

For the performance evaluation of our proposed distributed SL-PSO localization algorithm as compared to PSO and its variants like BPSO and modified BPSO to tackle the localization issue in IoT, we have designed and carried out our experiments in MATLAB R2016b. The simulation parameters for our experiments are listed in Table I. The proposed SL-PSO algorithm and other localization algorithms were run on the same PC with an Intel Core i7-6500 2.5 GHz CPU and Microsoft Windows 7 enterprise edition SP1 64-bit operating system. Also, for comparison purpose in our experiments, the deployment of the anchor nodes is kept identical for the SL-PSO and all other algorithms. The estimated IoT sensor node localization using our proposed SL-PSO localization algorithm is shown in Fig. 3. Similarly, the localization error using our proposed SL-PSO localization algorithm is shown in Fig. 4. The comparison of localization error for all the algorithms is shown in Fig. 5. Clearly, we can observe that, our proposed SL-PSO algorithm has significantly less localization error than PSO and its other variants BPSO, and Modified BPSO. Hence, we are able to do localization of sensor nodes more accurately with the SL-PSO localization algorithm compared to PSO, BPSO, and Modified BPSO algorithms.

Further, in our simulation experiments, the effect of randomness of proposed SL-PSO and other localization algorithms and anchor nodes is eliminated by testing the localization algorithm with 30 different test network and taking the average value by repeatedly running the algorithm for 30 times. To illustrate the effectiveness of the SL-PSO algorithm on localization, we have used 95% confidence interval (CI) of the average localization error E_{ALE}. A 95% confidence interval ($CI_{95\%}$) is an indicator of our measurement precision for the E_{ALE}.

The E_{ALE} within $CI_{95\%}$ range and computation time for each of these bio-inspired localization algorithms are shown in Table II and Table III respectively. From Table II., we can see that the SL-PSO algorithm can significantly reduce

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

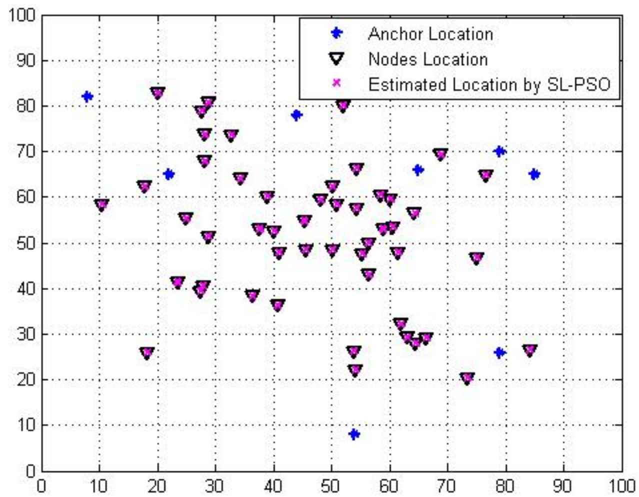

Fig. 3. Localization Using Social Learning based Particle Swarm Optimization (SL-PSO)

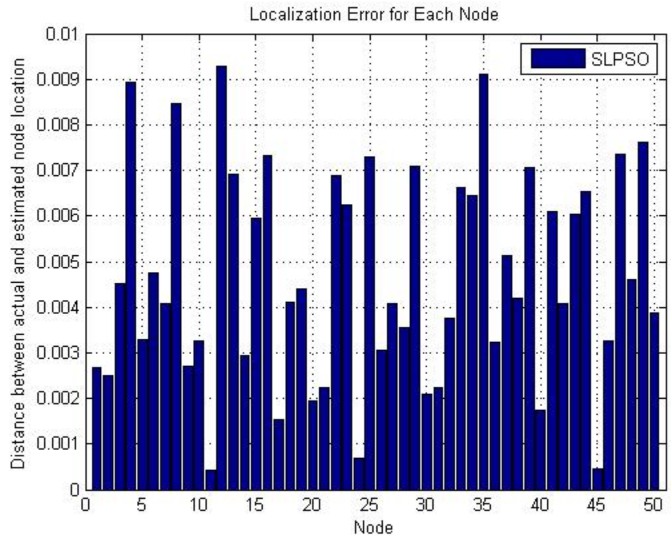

Fig. 4. Localization Error Using Social Learning based Particle Swarm Optimization (SL-PSO)

TABLE II
COMPARISION OF LOCALIZATION ERROR

Algorithm	Average Localization Error, E_{ALE} (m)	95% CI Lower Range, $LwrCI$ (m),	95% CI Upper Range, $UppCI$ (m)
SL-PSO	0.0024	0.0014	0.0040
PSO	0.0710	0.0674	0.0806
BPSO	0.2494	0.1684	0.3098
Modified BPSO	0.2494	0.1684	0.3098

TABLE III
COMPARISION OF COMPUTATION TIME

Algorithm	Computation Time (s)
SL-PSO	63.63875
PSO	139.34383
BPSO	100.43645
Modified BPSO	66.65743

localization error compared to PSO and its variants BPSO and Modified-BPSO. BPSO and Modified-BPSO algorithm have almost identical localization performance. For the computational complexity evaluation of the algorithms, we have used the tic-toc function of MATLAB. As shown in Table III, our proposed SL-PSO algorithm takes less computation time to run the localization algorithm. It is due to the fact that the SL-PSO algorithm learns from all other better particles and also learns from the mean value of the particles in the current swarm which helps to converge to optimization solution rapidly compared to PSO, BPSO and Modified-BPSO where the implicit learning process takes through only P_{best} and G_{best} vectors. With less computation time, our proposed SL-PSO algorithm can enhance the lifetime of these resource constrained and battery operated IoT sensor nodes.

The superior performance i.e., localization accuracy and convergence rate as shown by computation time of our proposed SL-PSO localization algorithm compared to PSO and its variants is quite evident from Table 2, Table 3 and Fig. 5.

V. CONCLUSION

Sensor nodes are the main components of IoT which bring the idea of IoT into reality. Apparently, emerging applications in IoT will depend on the accurate localization of thousands of these deployed sensors which is a classical optimization problem. Bio-inspired algorithms are known to be computationally efficient in accurate localization. Therefore, in this paper, we proposed a new distributed localization algorithm using SL-PSO for IoT. SL-PSO algorithm is inspired by the social learning mechanism which is widely observed in animals. We showed that the implicit social learning process through any better particles and mean behaviour of all particles in the current swarm helps SL-PSO algorithm in reducing localization error significantly and converge rapidly in finding global optimization solution unlike PSO algorithm where the implicit learning process is only through the P_{best} and G_{best} vectors. Extensive simulations have been performed to show the effective performance of SL-PSO algorithm over PSO and its variants like BPSO, Modified-BPSO on localization accuracy and computation performance. Simulations results showed that SL-PSO can outstandingly reduce localization error and further enhance the lifetime of these resource constrained and battery operated sensor nodes in IoT.

REFERENCES

[1] C.-W. Tsai, C.-F. Lai, and A. V. Vasilakos, "Future internet of things: open issues and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2201–2217, 2014.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665

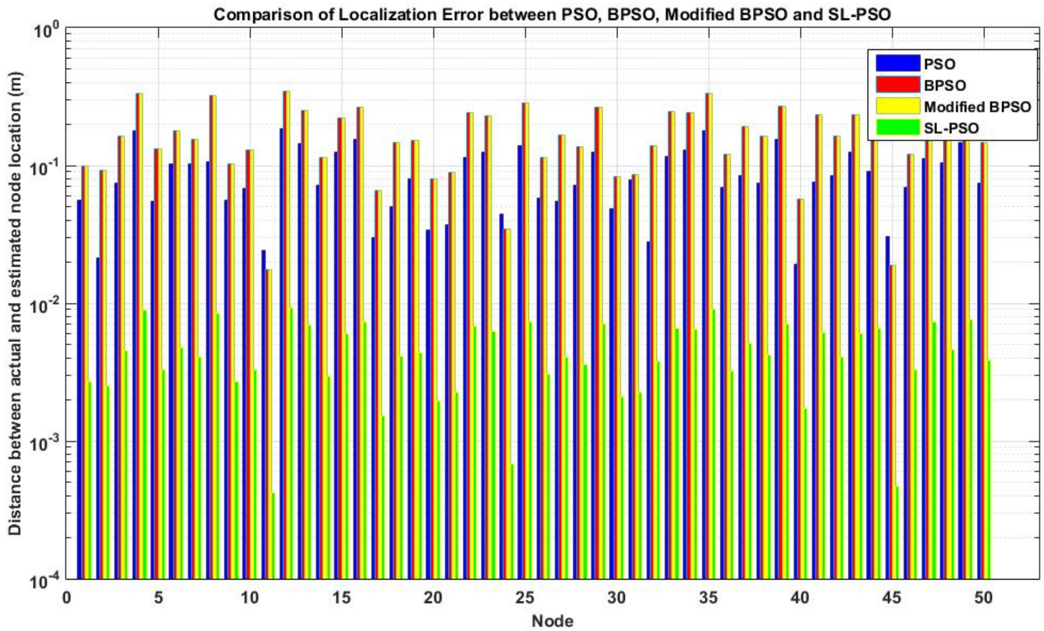

Fig. 5. Comparision of Localization Error Using PSO, BPSO, Modified BPSO, and SL-PSO Algorithm

[2] S. Hasan and E. Curry, "Thingsonomy: Tackling variety in internet of things events," *IEEE Internet Computing*, vol. 19, no. 2, pp. 10–18, 2015.

[3] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 68–75, 2011.

[4] Z. Chen, F. Xia, T. Huang, F. Bu, and H. Wang, "A localization method for the internet of things," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 657–674, 2013.

[5] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri, "A scalable and self-configuring architecture for service discovery in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 508–521, 2014.

[6] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[7] L. Zhou and H.-C. Chao, "Multimedia traffic security architecture for the internet of things," *IEEE Network*, vol. 25, no. 3, pp. 35–40, 2011.

[8] S. Sivakumar and R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks," *Applied Soft Computing*, vol. 36, pp. 506–518, 2015.

[9] J. Hightower and G. Borriello, "A survey and taxonomy of location systems for ubiquitous computing," *IEEE computer*, vol. 34, no. 8, pp. 57–66, 2001.

[10] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.

[11] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[12] S. Pandey and S. Varma, "A range based localization system in multihop wireless sensor networks: A distributed cooperative approach," *Wireless Personal Communications*, vol. 86, no. 2, pp. 615–634, 2016.

[13] M. Aziz, M.-H. Tayarani-N, and M. R. Meybodi, "A two-objective memetic approach for the node localization problem in wireless sensor networks," *Genetic Programming and Evolvable Machines*, vol. 17, no. 4, pp. 321–358, 2016.

[14] M. X. Cheng and W. B. Wu, "A model-free localization method for sensor networks with sparse anchors," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–7.

[15] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges," *Network protocols and algorithms*, vol. 2, no. 1, pp. 45–73, 2010.

[16] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[17] A. Gopakumar and L. Jacob, "Localization in wireless sensor networks using particle swarm optimization," in *Wireless, Mobile and Multimedia Networks, 2008. IET International Conference on*. IET, 2008, pp. 227–230.

[18] 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, no. 2, pp. 262–267, 2011.

[19] I. F. M. Zain and S. Y. Shin, "Distributed localization for wireless sensor networks using binary particle swarm optimization (bpso)," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*. IEEE, 2014, pp. 1–5.

[20] A. Céspedes-Mota, G. Castañón, A. F. Martínez-Herrera, and L. E. Cárdenas-Barrón, "Optimization of the distribution and localization of wireless sensor networks based on differential evolution approach," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[21] J. Cota-Ruiz, P. Rivas-Perea, E. Sifuentes, and R. Gonzalez-Landaeta, "A recursive shortest path routing algorithm with application for wireless sensor network localization," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4631–4637, 2016.

[22] J. Cheng and L. Xia, "An effective cuckoo search algorithm for node localization in wireless sensor network," *Sensors*, vol. 16, no. 9, p. 1390, 2016.

[23] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[25] S. Khezri, K. Faez, and A. Osmani, "Modified discrete binary pso based sensor placement in wsn networks," in *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, Nov 2010, pp. 200–204.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/VTCSpring.2018.8417665