Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Bio-Inspired Node Localization in Wireless Sensor Networks

Raghavendra V. Kulkarni, Ganesh K. Venayagamoorthy
Real-Time Power and Intelligent Systems Laboratory
Missouri University of Science and Technology
Rolla, USA
e-mail:{arvie, gkumar}@ieee.org

Maggie X. Cheng
Department of Computer Science
Missouri University of Science and Technology
Rolla, USA
e-mail: chengm@mst.edu

***Abstract*—Many applications of wireless sensor networks (WSNs) require location information of the randomly deployed nodes. A common solution to the localization problem is to deploy a few special beacon nodes having location awareness, which help the ordinary nodes to localize. In this approach, non-beacon nodes estimate their locations using noisy distance measurements from three or more non-collinear beacons they can receive signals from. In this paper, the ranging-based localization task is formulated as a multidimensional optimization problem, and addressed using bio-inspired algorithms, exploiting their quick convergence to quality solutions.**

An investigation on distributed iterative localization is presented in this paper. Here, the nodes that get localized in an iteration act as references for remaining nodes to localize. The problem has been addressed using particle swarm optimization (PSO) and bacterial foraging algorithm (BFA). A comparison of the performances of PSO and BFA in terms of the number of nodes localized, localization accuracy and computation time is presented.

***Index Terms*—bacterial foraging algorithm, localization, particle swarm optimization, wireless sensor networks**

I. Introduction

Wireless sensor networks (WSNs) are networks of distributed autonomous nodes that can sense their environment cooperatively [1]. WSNs are used in diverse applications such as environment and habitat monitoring, structural health monitoring, healthcare, home automation, and traffic surveillance. In monitoring applications, WSN nodes perceive their environment through onboard sensors.

Location is critically important in the WSNs used in monitoring and tracking applications. Location information is used to detect and record events, or to route packets using geometric-aware routing [2], [3]. Equipping each node with a global positioning system is not an attractive solution because of cost, size and energy constraints. Node localization, which refers to creating location awareness in all the deployed sensor nodes, is an area of active research.

Definitions and Problem Formulation: A WSN consists of N nodes, each having a communication range of r, distributed in a mission field. The WSN is represented as the Euclidean graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of sensor nodes. $\langle i, j\rangle \in E$ if the distance between v_i and v_j is $d_{ij} \leq r$. Unknown nodes (also known as free or dumb nodes) are the set $\mathcal{U}$ of non-beacon nodes that do not know their localization information. Settled nodes are the set $\mathcal{S}$ of nodes that managed to estimate their positions using the localization algorithm. Given a WSN $G = (V, E)$, and a set of beacon nodes $\mathcal{B}$ and their positions (x_b, y_b), for all $b \in \mathcal{B}$, it is desired to find the position (x_u, y_u) of as many $u \in \mathcal{U}$ as possible, transforming the unknown nodes into settled nodes $\mathcal{S}$.

WSN localization is a two-phase process. In the first phase known as ranging, nodes estimate their distances from beacons (or settled nodes) using the signal propagation time or the strength of the received signal. Precise measurement of these parameters is not possible due to noise; therefore, results of the localization algorithms that use these parameters are likely to be inaccurate. In the second phase, position estimation of the nodes is carried out using the ranging information. This is done either by solving a set of simultaneous equations, or by using an optimization algorithm that minimizes the localization error. In iterative localization algorithms, the settled nodes serve as beacons and the localization process is repeated until either all nodes are settled, or no more nodes can be localized.

This paper proposes two bio-inspired optimization algorithms for distributed iterative node localization in a WSN. The first algorithm is the particle swarm optimization (PSO) [4], and the second is the bacterial foraging algorithm (BFA) [5]. Both the algorithms have become popular in recent years as simple but efficient multidimensional search algorithms.

The rest of this paper is organized as follows: Section II presents a survey of previous research in WSN localization. Section III explains PSO and BFA, the optimization algorithms used for localization in this study. Section IV explains how the localization problem is approached using the above mentioned optimization methods. Section V discusses numerical simulation and the results obtained. Finally, section VI presents conclusions and makes a projection on possible future research path.

II. Related Work

Article [6] is a survey of localization systems for WSNs. An efficient localization system that extends GPS capabilities to non-GPS nodes in an ad hoc network is proposed in [7]. In this approach, anchors flood their location information to all nodes in the network. Then, each dumb node estimates its

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

location by triangularization. In the approach presented in [8], nodes improve their localization accuracy by measuring their distances from their neighbors. The issue of error accumulation is addressed in [9] through Kalman filter based least-square estimation. Node localization problem is addressed using convex optimization based on semi-definite programming in [10]. The semi-definite programming approach is further extended to non-convex inequality constraints in [11], and to a gradient search technique in [12].

WSN localization is treated as a multidimensional optimization problem and addressed though population-based techniques recently. PSO is proposed for centralized localization of WSN nodes in [13]. Performance evaluation of the approach shows that it provides more accurate localization compared to simulated annealing algorithm proposed in an earlier study [14]. This approach requires a large number of beacons in order to localize all dumb nodes. A genetic algorithm (GA) based node localization algorithm is presented in [15]. This centralized algorithm determines locations of all non-anchor nodes by using an estimate of their distances form all one-hop neighbors. A two-phase centralized localization scheme that uses simulated annealing and GA is presented in [16]. The two-phase centralized localization method that uses a combination of GA and simulated annealing algorithm proposed in [17] addresses the flip ambiguity problem.

Complexity and scalability issues in a WSN call for distributed localization algorithms which are executed on individual sensor node, rather than on a central base station. Each target node performs localization under imprecise measurement of distances from three or more neighboring anchors or settled nodes. The method proposed in this paper has following advantages over some of the earlier methods.

1) Each node estimates its location independently. This obviates communication with a central node, thus conserves energy and prevents congestion.
2) Localization accuracy is good.
3) Localization is robust against the noise associated with distance measurements.
4) Localization method is iterative. In each iteration, more nodes get settled. Thus, each node gets more references in its transmission range. This leads to correction of errors due to flip ambiguity, the situation that arises when the references are in near-collinear locations.

III. Bio-Inspired Techniques PSO and BFA for WSN Localization

Biology is a rich source of ideas for computer scientists. Popularity of bio-inspired algorithms is attributed to their accuracy, and their modest computational burden. The bio-inspired algorithms PSO and BFA are discussed in the following subsections.

A. Particle Swarm Optimization

PSO is a population based iterative parallel search algorithm that models social behavior of a flock of birds. Since its introduction in [4], PSO has seen many modifications and has been adapted to different environments [18]. Many versions of PSO have been proposed and applied to solve optimization problems in diverse fields [19].

PSO consists of a population (or a swarm) of s particles, each of which represents a potential solution. The particles explore an n-dimensional solution space in search of the global solution, where n represents the number of parameters to be optimized, x and y coordinates of a node this problem. Each particle i occupies a position X_{id} and moves with a velocity v_{id}, $1 \leq i \leq s$ and $1 \leq d \leq n$. Fitness of a particle is determined from its position in the search space. The fitness is defined in such a way that a particle closer to the global solution has higher fitness value than a particle that is far away. Each particle has a memory to store $pbest_{id}$, the position where it had the highest fitness, and $gbest_d$, the maximum of $pbest_{id}$s of all particles. The $gbest$ particle represents the best solution found so far. At each iteration k, velocity v_{id} and position X_{id} of each particle are updated using (1) and (2).

$$\begin{aligned} v_{id}(k+1) &= w \cdot v_{id}(k) + c_1 \cdot rand_1 \cdot (pbest_{id} - X_{id}) \\ &\quad + c_2 \cdot rand_2 \cdot (gbest_d - X_{id}) \qquad (1) \\ X_{id}(k+1) &= X_{id}(k) + v_{id}(k+1) \qquad (2) \end{aligned}$$

Here, $rand_1$ and $rand_2$ are random numbers that range between 0 and 1 with a uniform distribution. A pseudocode for PSO is given in Algorithm 1.

B. Bacterial Foraging Algorithm

BFA is a new evolutionary optimization algorithm introduced in [5] that mimics the foraging behavior of *Escherichia coli* (commonly called E. coli) bacteria that live in human intestine. There are successful applications of BFA and its hybrids in optimization problems such as PID controller tuning [20], and economic load dispatch [21].

An E. coli bacterium moves to a nutrient-rich location using a pattern of two types of movements, tumbling and swimming. Tumbling refers to randomly changing the direction of movement; and swimming refers to moving without changing the direction. A bacterium in a neutral medium alternates between tumbling and swimming moves. A bacterium that is moving in a direction towards better nutrient locations keeps moving in the same direction in a swimming movement. But if swimming takes it to a location having lower nutrient concentration, it takes a tumble movement and swiftly changes its direction. With a tumble followed by a few swimming steps collectively called a chemotactic round, the bacterium succeeds in attaining a favorable location. After a series of chemotactic steps, the bacteria that achieve good foraging split into two; and the others die. This is called a reproduction step.

Suppose that it is desired to search for a position in a p-dimensional space where function $J(P), P \in \Re^p$ has the global minimum. Let P_i be the initial position of bacterium i in the search space, $i = 1, 2, \cdots, S$, where S is the number of bacteria. Let $J(P_i)$ represent an objective function. Let $J(P_i) < 0$, $J(P_i) = 0$ and $J(P_i) > 0$ represent the bacterium at location P_i in nutrient rich, neutral and noxious

Algorithm 1 The global-best version of PSO for minimization of a cost function

Initialize w, c_1 and c_2
Initialize maximum allowable iterations $k_{\max}$
Initialize the target fitness f_T
Initialize $X_{\min}$, $X_{\max}$, $v_{\min}$ and $v_{\max}$
for each particle i **do**
  for each dimension d **do**
    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 $k = 0$
while ($k \leq k_{\max}$) AND ($f(gbest) > f_T$) **do**
  for each particle i **do**
    Compute $f(X_i)$
    if $f(X_i) < f(pbest_i)$ **then**
      for each dimension d **do**
        $pbest_{id} = X_{id}$
      end for
    end if
    if $f(X_i) < f(gbest)$ **then**
      for each dimension d **do**
        $gbest_d = X_{id}$
      end for
    end if
  end for
  for each particle i **do**
    for each dimension d **do**
      Compute velocity $v_{id}(k+1)$ using (1)
      Restrict v_{id} to $v_{\min} \leq v_{id} \leq v_{\max}$
      Compute position $X_{id}(k+1)$ using (2)
      Restrict X_{id} to $X_{\min} \leq X_{id} \leq X_{\max}$
    end for
  end for
  $k = k + 1$
end while

Algorithm 2 Bacterial Foraging Algorithm for minimization of a cost function

Initialize bacteria positions $C(i), i = 1, 2, \cdots, S$
Initialize $p, S, N_c, N_{re}, N_{ed}, p_{ed}, N_s, d_a, w_a, h_r$ and w_r
Set the loops indices j, k and l to 0.
//Elimination-Dispersal loop:
while $l \leq N_{ed}$ **do**
  $l = l + 1$
  //Reproduction loop:
  while $k \leq N_{re}$ **do**
    $k = k + 1$
    //Chemotaxis loop:
    while $j \leq N_c$ **do**
      $j = j + 1$
      for each bacterium $i = 1, 2, \cdot, S$ **do**
        Compute $J(i, j, k, l)$
        Let $J(i, j, k, l) = J(i, j + 1, k, l) + J_{cc}(P_i(j, k, l), \mathcal{P}(j, k, l))$
        Let $J_{last} = J(i, j, k, l)$
        //Tumble:
        Generate a p-dimensional random vector $\Delta_m(i), i = 1, 2, \cdots, p$ on [-1,1]
        //Move:
        Let $P_i = P_i(j + 1, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$
        Compute $J(i, J + 1, k, l)$
        Let $J(i, j + 1, k, l) = J(i, j + 1, k, l) + J_{cc}(P_i(j + 1, k, l), \mathcal{P}(j + 1, k, l))$
        Swim: Let $m = 0$
        while $m < N_s$ **do**
          Let $m = m + 1$
          if $J(i, j + 1, k, l) > J_{last}$ **then**
            Let $J(i, j + 1, k, l) = J_{last}$
            Let $P_i = P_i(j + 1, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$
            Use this P_i to compute new $J(i, j+1, k, l)$
          else
            $m = N_s$
          end if
        end while
      end for
    end while
    Compute for each bacterium i, for given k and l
    $J^i_{health} = \sum_{j=1}^{N_c+1} J(i, j, k, l)$
    Eliminate S_r bacteria with highest J_{health} and split the other S_r bacteria at the same locations as the original ones.
  end while
  For each bacterium, with probability P_d eliminate the bacterium and create a new one at a random position.
end while

environments, respectively. Chemotaxis is a foraging behavior that captures the process of optimization where bacteria try to climb up the nutrient concentration (i.e., bacteria try to achieve positions having lower values of $J(P_i)$) and avoid being at positions P_i where $J(Pi) \geq 0$) [5]. A detailed pseudocode for BFA is given in Algorithm 2.

IV. Iterative Localization using PSO and BFA

The objective of WSN node localization is to perform distributed estimation of coordinates of the maximum of N target nodes using M stationary beacons which know their locations. This study approaches node localization in a WSN in the following way:

1) N dumb nodes and M beacons are randomly deployed in a 2-dimensional sensor field. Each dumb node and each beacon has a transmission radius of r units. Beacon nodes possess location awareness, and they frequently transmit their coordinates. The nodes that get settled at the end of an iteration serve as references. They transmit their location information as the beacons do.

2) Each node that falls within transmission radii of 3 or more non-collinear references (beacons or settled nodes) is referred to as a localizable node.
3) Each localizable node in the deployment estimates its distance from each of its neighboring beacons or settled nodes. The effect of measurement noise is simulated as a Gaussian additive white noise. A node estimates its distance from a beacon i as $\hat{d}_i = d_i + n_i$, where d_i is the actual distance given by $d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$. Here (x, y) is the location of the target node, and (x_i, y_i) is the location of the i^{th} beacon in the neighborhood of the target node. The measurement noise n_i has a random value uniformly distributed in the range $d_i \pm d_i \frac{P_n}{100}$.
4) Two case studies are conducted: In case study 1, each localizable node runs PSO to localize itself. In case study 2, each localizable node runs BFA to localize itself. Both PSO and BFA find the coordinates (x, y) that minimize the objective function that represents the error defined in (3).

$$f(x,y) = \frac{1}{M}\sum_{i=1}^{M}\left(\sqrt{(x-x_i)^2 + (y-y_i)^2} - \hat{d}_i\right)^2, \tag{3}$$

where $M \geq 3$ is the number of beacons or settled nodes within the transmission radius of the target node.
5) PSO and BFA search for best values of (x, y) that minimize the error, therefore the dimensionality of the search space is 2.
6) After all the N_L localizable nodes determine their coordinates, the total localization error is computed as the mean of squares of distances between actual node locations (x_i, y_i) and the locations $(\hat{x}_i, \hat{y}_i)$, $i = 1, 2, \cdots, N_L$, determined by PSO or BFA. This is computed as (4).

$$E_l = \frac{1}{N_L}\sum_{i=1}^{L}\left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right) \tag{4}$$

7) Steps 2 to 6 are repeated until either all dumb nodes get localized or no more nodes can be localized. The performance of the localization algorithm is determined by the doublet (N_{NL},E_l), where $N_{NL} = N - N_L$ is the number of nodes that could not be localized. The lower the values of N_{NL} and E_l, the better the performance is.

As the iterations progress, the number of localized nodes increases. This increases the number of references available for already localized nodes. A node that localized using just three references in an iteration k may have more references in iteration $k+1$. This decreases the probability of flip ambiguity. On the other hand, If a node has more references in iteration $k+1$ than in iteration k, the time required for localization increases. This is avoided in this study by restricting the maximum number of reference to six, which is arbitrarily chosen.

V. Numerical Simulation and Results

Simulation of the WSN and its performance evaluation is done in Matlab™. 50 target nodes and 10 beacons are randomly deployed in a sensor field having dimensions of 100×100 square units. Each beacon has a transmission radius of $r = 25$ units. Simulation settings specific to case studies 1 and 2, and the result obtained are presented in the following subsections.

A. Case Study 1: PSO-based Localization

In this case study, each localizable target node runs a 2-dimensional PSO to localize itself. PSO parameters are set as follows:

- Population = 30, Iterations = 150
- Acceleration constants $c_1 = c_2 = 2.0$
- Inertial weight is decreased linearly from 0.9 in the first iteration to 0.4 in the last iteration
- Limits on particle positions: $X_{\min}$=0 and $X_{\max}$=100

30 trial experiments of PSO-based localization are conducted for P_n = 2 and P_n = 5. Average of total localization error E_l defined in (4) in each iteration in 30 runs is computed. Average of total localization error E_l defined in (4) is computed.

B. Case Study 2: BFA-based Localization

In this case study, each localizable target node runs a 2-dimensional BFA to localize itself. BFA parameters are set as follows:

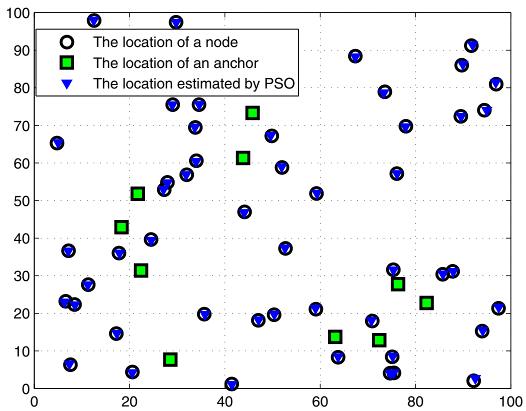
- Population = 30
- Number of chemotactic steps N_c = 5,
- Number of swims N_s = 20
- Number of reproduction rounds N_{re} = 5
- Number of elimination-dispersion rounds N_{ed} = 5
- Fraction of bacteria that split in each reproduction round S_r = 0.5
- Probability that a bacterium is eliminated in an elimination-dispersion round p_{ed} = 0.1;

30 trial experiments of BFA-based localization are conducted for P_n = 2 and P_n = 5. Average of total localization error E_l defined in (4) is computed. Both the algorithms studied here are stochastic, therefore they do not produce the same solutions in all trials even with identical initial deployment. This is the reason why the results of multiple trial runs are averaged. Besides, initial deployment is random, so the number of localizable nodes in each iteration is not the same. This affects the total computing time.

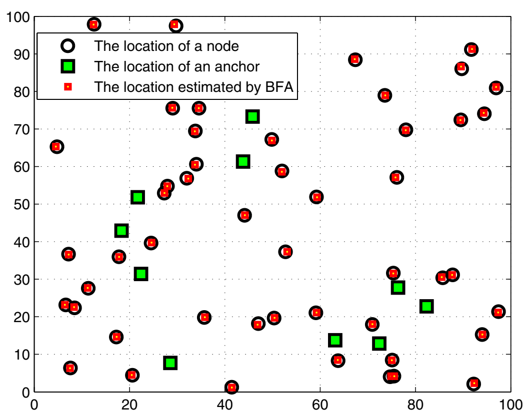
C. Discussion on the Results

The actual locations of nodes and beacons, and the coordinates of the nodes estimated by PSO and BFA in a trial run are shown in Figure 1.

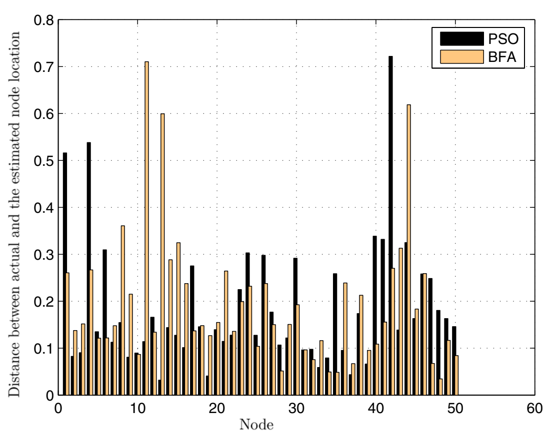
The initial deployment of nodes and beacons for PSO and BFA-based localization is the same in a trial run. Results of PSO and BFA-based localization summarized in Table I show that both stochastic algorithms used here have performed fairly well in WSN localization. The effect of P_n, percentage noise in distance measurement, on localization accuracy can be



(a) Locations estimated by PSO



(b) Locations estimated by BFA



(c) Distances between actual locations and those estimated by PSO and BFA

Fig. 1. Results of a trial run of PSO and BFA-based iterative localization for the identical deployment with $N = 50$, $M = 10$, $r = 25$ units and the sensor field size of 100×100 square units.

clearly seen. Average localization error in both PSO and BFA is reduced when P_n is changed from 5 to 2. The performance metric doublet (N_{NL},E_l) for BFA is less than that for PSO, indicting superior performance of BFA. However, computing time required for BFA is significantly more than that for PSO, which is a weakness of BFA. In addition, the amount of memory required for BFA is more than that for PSO. This clearly calls for a trade off. A choice between PSO and BFA is influenced by how constrained the nodes are in terms of memory and computing resources, how accurate the localization is expected to be and how quickly that should happen.

The detailed observations made in the first five trial runs out of the 30, are summarized in Table II. This table depicts increasing N_L, the number of localized nodes, in each iteration. It also shows the correction of large errors due to flip ambiguity.

VI. Conclusions and Future Work

This paper has discussed PSO and BFA, bio-inspired algorithms for determining coordinates of the nodes in a WSN in a distributed and iterative fashion. The localization problem is treated as a multidimensional optimization problem and addressed through the aforementioned population-based optimization algorithms. Distributed localization proposed here has the advantage of reduced number of transmissions to the base station, which helps the nodes conserve their energy, which is a serious concern in most WSN applications. The paper has briefly outlined the algorithms and presented a statistical summary of their results for comparison. The results show that the proposed algorithms have a trade off issue. While the PSO determines the node coordinates more quickly, the BFA does so more accurately.

This work can be extended in several directions. Literature is rife with centralized localization algorithms. In a possible future study, both PSO and BFA can be used in centralized localization method in order to compare the performances of centralized and distributed localization methods. Such a comparison with an emphasis on energy awareness will be particularly useful. Besides, a comparison of the stochastic localization methods with the available deterministic methods can give an useful insight. This can be another outgrowth of the study conducted in this paper.

In summary, both PSO and BFA have performed fairly well on distributed iterative localization in WSNs. Both algorithms have their own strengths and weaknesses, which have been pointed out in the paper. A judicial choice between the algorithms depends on memory and computing resources on the node and desired localization speed and accuracy.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.

TABLE I

A SUMMARY OF RESULTS OF 30 TRIAL RUNS PSO AND BFA-BASED WSN NODE LOCATION. $N = 50$, $M = 10$, $r = 25$ UNITS, AND THE SENSOR FIELD SIZE = 100×100 SQUARE UNITS.

	Percentage noise in distance measurement $P_n = 5$			Percentage noise in distance measurement $P_n = 2$		
	Mean number of non-localized nodes N_{NL}	Mean localization Error E_l	Computing time* (s)	Mean number of non-localized nodes N_{NL}	Mean localization Error E_l	Computing time* (s)
PSO	1.1753	0.8953	**261.5371**	0.3210	0.3511	**288.4117**
BFA	**0.7923**	**0.5822**	1022.1973	**0.1956**	**0.2137**	902.1132
*All experiments are conducted on the same computer.						

TABLE II

A SUMMARY OF RESULTS OF PSO AND BFA-BASED WSN NODE LOCATION WITH THE SAME INITIAL DEPLOYMENT OF NODES AND ANCHORS WITH $N = 50$, $M = 10$, $r = 25$ UNITS, AND THE SENSOR FIELD SIZE = 100×100 SQUARE UNITS.

		PSO					BFA				
Trial		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1	N_L	23	36	40	45	49	23	35	42	45	50
	E_l	0.2611	0.0701	0.0564	0.0548	0.0595	0.1376	0.0740	9.1351	0.0768	0.0744
	T_l	9.2810	30.5310	52.8600	71.0940	80.8750	34.6410	108.4850	165.3120	225.1720	313.7660
2	N_L	27	36	40	42	47	27	36	42	48	
	E_l	38.3963*	0.0434	0.0304	47.6885	0.0439	38.4539*	0.1818	0.1545	0.0216	
	T_l	12.4380	37.6410	60.8280	77.5940	92.9690	45.9530	142.5000	232.4530	297.5940	
3	N_L	24	43	50			24	39	46	49	50
	E_l	0.0599	0.0733	0.0768			1.0438	18.9131*	0.1099	0.0925	0.0692
	T_l	10.5160	31.6870	58.5160			38.1720	108.6100	190.8900	246.3910	302.9060
4	N_L	26	44	50			26	41	48	50	
	E_l	13.5332*	0.0659	0.0470			0.1142	0.1185	0.0831	0.0213	
	T_l	9.5940	29.9370	50.1100			37.4530	108.2820	184.0780	224.3220	
5	N_L	23	38	44	46	49	23	39	47	49	49
	E_l	50.8360*	30.6656	0.0929	0.0910	0.0554	0.1220	11.0965	0.1664	0.0543	0.0215
	T_l	9.4690	31.8750	53.0000	69.1400	91.9220	35.1720	112.4220	181.9840	243.2660	283.8280
All experiments are conducted on the same computer											
* These cases represent large errors due to flip ambiguities, which the system corrected in subsequent iterations.											

[3] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, and P. Belhumeur, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, Perth, Australia, Jan. 1995, pp. 1942–1948.

[5] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, June 2002.

[6] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 14, no. 6, pp. 6–12, December 2007.

[7] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE Global Telecommunications Conf. GLOBECOM '01*, vol. 5, 25–29 Nov. 2001, pp. 2926–2931.

[8] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *in USENIX Technical Annual Conf.*, 2002, pp. 317–327.

[9] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems," 2002.

[10] L. Doherty, K. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE 20th Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2001*, vol. 3, 22–26 April 2001, pp. 1655–1663.

[11] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 188–220, 2006.

[12] T.-C. Liang, T.-C. Wang, and Y. Ye, "A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization," Tech. Rep., 2004.

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

[14] A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based localization in wireless sensor network," in *Proc. 30th Anniversary IEEE Conf. on Local Computer Networks*, 2005, pp. 2 pp.–.

[15] G.-F. Nan, M.-Q. Li, and J. Li, "Estimation of node localization with a real-coded genetic algorithm in WSNs," in *Proc. Int. Conf. on Machine Learning and Cybernetics*, vol. 2, 2007, pp. 873–878.

[16] M. Marks and E. Niewiadomska-Szynkiewicz, "Two-phase stochastic optimization to sensor network localization," in *Proc. Int. Conf. on Sensor Technologies and Applications SensorComm 2007*, 2007, pp. 134–139.

[17] Q. Zhang, J. Huang, J. Wang, C. Jin, J. Ye, W. Zhang, and J. Hu, "A two-phase localization algorithm for wireless sensor network," in *Proc. Int. Conf. on Information and Automation ICIA 2008*, pp. 59–64.

[18] X. Hu, Y. Shi, and R. Eberhart, "Recent advances in particle swarm," in *Proc. CEC2004 Congress on Evolutionary Computation*, vol. 1, 19–23 June 2004, pp. 90–97.

[19] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[20] W. M. Korani, "Bacterial foraging oriented by particle swarm optimization strategy for PID tuning," in *GECCO '08: Proc. 2008 GECCO Conf. companion on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2008, pp. 1823–1826.

[21] A. Y. Saber and G. K. Venayagamoorthy, "Economic load dispatch using bacterial foraging technique with particle swarm optimization biased evolution," in *Proc. IEEE Swarm Intelligence Symposium SIS 2008*, 21–23 Sept. 2008, pp. 1–8.