Research Article   **Open Access**

Baldwinian Learning in Quantum Evolutionary Algorithms for Solving the Fine-Grained Localization Problem in Wireless Sensor Networks

Mahdi Aziz[1]* and Mohammad Meybodi[2]

[1]University of British Columbia Okanagan, University of British Columbia, Okanagan Kelowna, British Columbia, Canada
[2]AmirKabir University of Technology, Tehran

Abstract

A Local Search (LS) procedure is a search facilitator, giving memetic algorithms a hand to enhance their exploitation ability resulting in converging to higher quality solutions. In this paper, using the LS procedure in the form of Baldwinian Learning (BL) a Memetic Quantum Evolutionary Algorithm (QEA) is proposed for tackling the fine grained localization problem in Wireless sensor networks (WSNs). Since the QEA can be used only for binary-domain problems like the knapsack problem, we utilize the binary-to-real mapping procedure to make it suitable for solving the localization problem in WSNs. To provide good initial positions of sensor nodes, the algorithm employs a Multi-Trilateration (MT) procedure on the best observed solutions. To test the proposed algorithm, it is first compared with its two spin-offs (the proposed algorithm without the MT procedure and the proposed algorithm without the BL and MT procedures) and then compared with six existing optimization algorithms on ten randomly generated network topologies with four different connectivity ranges. The simulation results suggest that the proposed algorithm significantly outperforms the other algorithms in terms of estimating the positions of sensor nodes in WSNs. They also point out the effectiveness of applying the MT procedure and BL method to the proposed algorithm in solving the problem.

Keywords: Baldwinian learning; Quantum evolutionary algorithm; Localization problem; Wireless sensor network

Introduction

A wireless sensor network is a modern and advanced communication technology, used for sensing and controlling the physical environment. WSNs are often composed of thousands or even hundreds of thousands of cheap and resource limited sensor nodes collaboratively working together to track an object or to monitor a physical event. Although designed first for military purposes, they are now being used for crisis management, monitoring of temperature, humidity control, vehicular movement and lightning condition [1].

The localization problem are becoming one of the hottest topics in WSNs since the localization information is useful for routing [2], coverage [3], boundary detection [4], clustering [5] and topology control [6]. Algorithms for the localization problems in WSNs can be categorized into two classes: the range-free algorithms and the range-based algorithms [7]. The range-free algorithms use connectivity based information, obtained by radio communication, such as neighbourhood or hop count to measure inter-node distances [8]. These algorithms are somewhat simple and cheap, and they impose no requirement of additional hardware. However, they are less accurate than the range-based algorithms. The range-based algorithms, on the other hand, utilize range based information such as location, range or angle to measure pair-wise distances of nodes [9,10]. This information usually comes from Time of Arrival (TOA), Time difference of Arrival (TDOA) or Received Signal Strength (RSS) measurement techniques [11,12]. Although range-based algorithms require special hardware such as radio signal receivers or antenna to estimate the positions of sensor nodes, they provide pair-wise distances of nodes with higher accuracy. The former are called the course-grained algorithms, and the latter are called the fine-grained algorithms [13].

The geographical location of sensor nodes in WSNs can be obtained by several approaches. First, it can be acquired through manual configuration. If the number of sensor nodes is small and the area at which they were deployed is limited, this approach will be a good option.

However, it is practically impossible since a WSN often contains a large number of sensor nodes, making the localization process intractable. Second, all sensors nodes can be equipped by GPS receivers. Despite localizing sensor nodes with good accuracy, this method has several problems such as being prohibitively expensive in case of large-scale WSNs and not working well in the place surrounded by large buildings or indoor and underground sites [14]. Finally, GPS receivers can be embedded on a few nodes called anchor or reference nodes, and the other nodes called non-anchor nodes can locate themselves though these reference nodes [15]. Given this approach, algorithms for WSNs can be classified into three classes: multidimensional scaling, relaxation and stochastic techniques [16]. The multidimensional scaling is a connectivity-based technique using distance-based information to estimate the relative positions of nodes. This method works well on the RSS measurement. However, in this method, all sensor nodes need to be in the vicinity of each other so each node can estimate its location through the relative positions of the other nodes [17]. The relaxation is another approach that was firstly suggested by Doherty [18]. In this approach, to solve the localization problem in WSNS the localization problem is turned to a Semi-Definite Programming problem (SDP), which is more easily to solve; then, geometric constraints among sensor nodes in the network are represented as linear matrix inequalities (LMIs). Finally, these LMIs join together to form a single semi-definite problem, which is tackled to create a bounding region for each node. Although this method is promising in case of large-scale

network localization problems, the problem associated with this method is twofold. First, it cannot provide high-accuracy estimations for sensor nodes. In some applications such as fire detection in forests, it is necessary to know where sensor nodes are in order to do what it is necessary to do to cope with the issue immediately. Second, all geometrical constraints in the network cannot be formulated as LMIs and only geometrical constraints that form convex regions can be transformed to the LMIs [19].

The stochastic technique is another approach that was originally proposed by Kannan [20] where he showed that simulated annealing algorithm is a promising technique to solve the node localization in WSNs. It is easy to implement and requires small amount of computational effort, but when the flip ambiguity problem occurs, its performance plummets down. This problem crops up when three or more neighbours of the node being localized are almost collinear, and thus its location cannot be determined uniquely. This incorrect information then propagates to the entire network or a large region of it, causing mass confusion in finding the true locations of other sensor nodes. This phenomenon is shown in Figure 1, where three anchor nodes G, B and C are placed around a hypothetical line, and so the location of the non-anchor node D cannot be estimated correctly. To surmount this, Kannan then proposed a new version of SA (SAL), which uses a refinement phase to mitigate its effects [21]. Given the fine-grained localization problem as a two-objective optimization task, reference [22] proposed a two-objective evolutionary approach called the Pareto Archived Evolution Strategy (PAES), which attempts to address both the localization error and the flip ambiguity problem simultaneously. They showed that the PAES algorithm can successfully deal with the node localization problem and, in comparison with the SAL algorithm; it can provides solutions with higher quality.

Population-based algorithms are another group of optimizers, using a set of search agents (solutions) to localize sensor nodes. For instance, Liu and Yang [23] proposed a genetic algorithm named as Genetic Algorithm-based Localization (GAL) that employs two genetic operators, called the single-vertex-neighbourhood mutation and the descend-based arithmetic crossover, to localize sensor nodes in a WSN effectively. In another work, Mao and Fidan [24] proposed a new Particle Swarm Optimization Localization (PSOL) algorithm that applies a swarm of search agents working cooperatively to find good locations for sensor nodes.

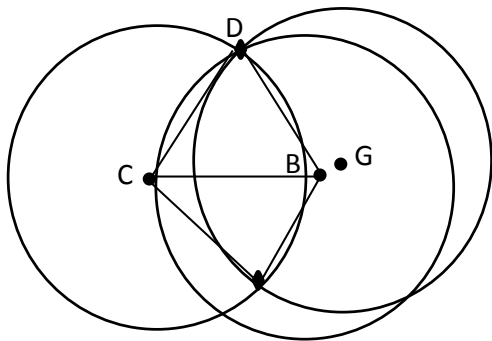In general, optimization techniques for the fine-grained localization problem in WSNs can be categorized into two different groups. The first group uses only stochastic technique such as SAL [16], PSOL [25] or GAL [26]. The second group, on the other hand, uses not only a stochastic optimizer but also an approximation stage such as multi-trilateration [27] or priori knowledge such as node-categorizing information [28] to find initial locations of sensor nodes.

QEA is a nature-inspired population-based algorithm, which is inspired from the quantum computation [29,30]. Once proposed, QEA has been used in a variety of optimization problems like transient identification of nu clear power plants [31], vehicle routing problem [32], Watermarking Algorithm [33], Economic Load Dispatch [34] and VAR Planning [35], Unit Commitment [36], and many researchers have tried to improve its performance [37,38].

Both local search procedures and global search algorithms (population-based algorithms) have some advantages and weaknesses. Local search algorithms are efficient and promising once they apply to simpler problems (those that have smaller number of local optima). But, for the problems with large number of local optima, they may easily get trapped in local optima [39,40]. On the other hand, population-based algorithms are effective for more difficult problems (those that have larger number of local optima) and they are less efficient in case of simpler problems [41]. So, one might say combining these two algorithms in a synergistic way can yield a better algorithm because the local search can cover the weakness of the population-based algorithm, that is, inability of focusing on a particular region of the search space, and the global search can cover the weakness of the local search, that is, inability of an escape from the basin of the attraction of the local optima being trapped in.

Making a trade-off between exploration and exploitation, Memetic Algorithms (MAs) are being proposed to solve not only small-scale optimization problems but also tackle large-scale optimization ones [42]. A memetic algorithm is an evolutionary algorithm, favoured by one or several local procedures. These local search procedures help the MAs to locate the local optima more quickly [43]. The memetic algorithms are inspired by the Neo-Darwinian idea in the natural evolution and Dawkin's cultural evolution unit called the "meme". In Dawkin's cultural evolution, a meme is the smallest unit of knowledge that can be reproduced, changed or improved. If a meme is an interesting one, it will be distributed with high probability within the entire population; if not, it will probably disappear in the next generations. On the other hand, in memetic algorithms, first coined by Moscato, a meme is a local learning procedure, which improves individuals in a population of solutions. These algorithms have recently drawn the attention of many researchers in solving a wide range of real-world problems, including quadratic assignment problem, flow shop scheduling [22], capacitated arc routing problem, DNA sequence compression, and university course timetabling.

Having been proposed by James Baldwin [3], BL [9], which is also called Baldwinian evolution [8] or Baldwin effect [41], is a natural evolution in the population suggesting that the individuals with higher level of adaptability to the changes in the environment have a better chance to live, and they survive longer than their competitors in the population. In MAs, likewise, the individuals with greater fitness values remain alive longer in the population through being selected in the next generations. Baldwinian-based MAs have been used for solving a range of optimization problems, including Terminal Assignment in Communications Networks [40], Numerical Functions [9,41], Feature Weighting in K-MEANS-based algorithms [8] and describing continuous-valued problem spaces [37].

Figure 1: The flip ambiguity phenomenon in the wireless sensor networks. The nodes G, B and C are anchor nodes roughly located on a straight line; therefore, the non-anchor node D cannot be localized correctly.

In this paper, we propose a new memetic algorithm that uses QEA as the global search and the Baldwinian local search as the local search. The Baldwin local search helps the algorithm boost its exploitation ability and thus mitigating the stagnation tendency when solving the localization problem in WSNs. The binary-to-real mapping procedure makes the algorithm appropriate for solving the localization problem in WSNs. The proposed algorithm is also favoured with the MT procedure, claimed to be very efficient in providing good starting locations for sensor nodes [28]. From our best knowledge, it is the first time that the memetic algorithm is used for solving the localization problem in WSNs. To test the proposed algorithm (QEA+MT+BL), it is network; anchor nodes are the ones that are fully aware of their positions. This knowledge comes from their GPS receivers or their individual records. The non-anchor nodes, on the other hand, are the ones, which do not know their positions. The aim is to find out the positions of non-anchor nodes by using the geographical information of anchor nodes.

All sensor nodes have an equal connectivity range, r and compared with the proposed algorithm without BL (QEA+MT); they are distributed uniformly in a two-dimensional region with a range of R2. [43], PSO [5], ICA [30], TSA [28] and PAES [34] on ten randomly created network topologies and four different connectivity ranges. The results show that the proposed algorithm performs best on the localization problem in WSNs.

The remainder of this paper is structured as follows. In Section 2, the fined-grained localization problem is described. The QEA is presented in Section 3. The proposed algorithm is introduced in Section 4. In Section 5, we compare the pro- posed algorithm with two different variant of QEA and six optimization techniques based upon simulation results. Finally, the paper is concluded in Section 6.

Problem Definition

In this section, we define the fine-grained localization problem in WSNs, concentrating on the system model, the evaluation of the proposed algorithm during the optimization, and the assessment of the performance of the algorithm after the optimization.

System model

A wireless sensor network can be considered as a network consisting of anchor nodes and non-anchor nodes. In this network, anchor nodes are the ones that are fully aware of their positions. This knowledge comes from their GPS receivers or their individual records. The non-anchor nodes, on the other hand, are the ones, which do not know their positions. The aim is to find out the positions of non-anchor nodes by using the geographical information of anchor nodes. All sensor nodes have an equal connectivity range r, and they are distributed uniformly in a two-dimensional squared region with a range of $[0,1] \times [0,1] \subset R^2$. We use RSSI measurement to estimate the internode distances $\tilde{d}_{ij}$ since it was shown that it can provide measurements with good accuracy and the low cost of hardware [24]. We assume that $\tilde{d}_{ij}$ is computed as,

$$\tilde{d}_{ij} = d_{ij} \times (1.0 + \gamma \times NoiseFactor) \qquad (1)$$

where $\tilde{d}_{ij}$ and d_{ij} are the measured, and the true distance between the ith and j^{th} nodes, respectively and γ is a Gaussian noise, added to the distances because of the measurement noise. The mean and standard deviation of the Gaussian noise are equal to 0 and 1, respectively. We assume that the measurement errors are distributed uniformly across the network. We use a simple disk model that are typically used in the literature [16,28,34] for network communication in that sensor nodes can communicate with each other as long as the actual distances among

them are to be less than the communication range. For instance, node i can communicate with node j if di j<r.

Objective function and performance evaluation

In this paper, we use the following objective function [28] for evaluating the proposed algorithm during the optimization process. The objective function CX is found as

$$CX = \sum_{k=1}^{m} \sum_{j \in S_k} (d_{kj}^{\wedge} - d_{kj}^{-})^2 + \sum_{i=1}^{n} \sum_{j \in S_i} (d_{ij}^{\wedge} - d_{ij}^{-})^2 \qquad (2)$$

Where m and n are the number of anchor nodes and non- anchor nodes respectively; $d^{\wedge}kj$ is the estimated distance between the anchor node k and the non-anchor node j, calculated $d_{kj}^{\wedge} = ||a_k - x_j^{\wedge}||, d_{ij}^{\wedge} = ||x_i^{\wedge} - x_j^{\wedge}||$ as where a_k is the real position of anchor node k and $x_i^{\wedge}$, $x_j^{\wedge}$ are the estimated positions of nodes i, j. Further, in Formula 2, $d_{kj}^{\wedge}$ represents the measured distance between non-anchor node i and j and $d_{kj}^{\wedge}$.

In order to evaluate the performance of the proposed algorithm, we use the following metric that calculates the distance between the estimated and the real positions of non- anchor nodes in the network.

$$LE = \frac{1}{n} \sum_{i=1}^{n} \frac{(|| x_i^{\wedge} - x_i ||)^2}{r^2} \times 100 \qquad (3)$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad (4)$$

Where α and β are complex numbers, which represent the corresponding state appearance probability, following below constraint: is measured distance between anchor node k and non-anchor node j. As mentioned, the distances among nodes are measured through Formula 1.

$$|\alpha|^2 + |\beta|^2 = 1 \qquad (5)$$

One of the advantages of using probabilistic representation is to simply represent 2m states simultaneously by using m q-bits. At each observation, a q-bits quantum state col- lapses to a single state as determined by its corresponding probabilities. Consider i^{th} individual in t^{th} generation, for instance, which is defined as an m-q-bit as below: of ith non-anchor node.

QEA

QEA is a problem-solving technique, using a set of probabilistic individuals to discover promising regions in the search i=1, 2, ..., n, n is the number of possible solutions in population, and t is the current generation number of the evolution space. Even if using a small number of individuals, the QEA preserves the diversity longer in the population. It is inspired from quantum computation, and its superposition of states is based on q-bit, the 'brick' or the smallest unit of information saved in a two-state quantum computer. A q-bit can be rep-initialized with 1. This implies that each q-bit individual 2q0 represents the linear superposition of based on q-bit, the 'brick' or the smallest unit of information saved in a two-state quantum computer. A q-bit can be rep initialized with 1. This implies that each q-bit individual 2q0 represents the linear superposition of described below:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad (4)$$

Where α and β are complex numbers, which represent the corresponding state appearance probability, following below constraint:

$$|\alpha|^2 + |\beta|^2 = 1 \qquad (5)$$

One of the advantages of using probabilistic representation is to simply represent 2m states simultaneously by using m q-bits. At

each observation, a q-bits quantum state collapses to a single state as determined by its corresponding probabilities. Consider i^{th} individual in t^{th} generation, for instance, which is defined as an m-q-bit as below:

$$\begin{bmatrix} \alpha_{i1}^{t} \; \alpha_{i2}^{t} \ldots \ldots \alpha_{ij}^{t} \ldots \ldots \alpha_{im}^{t} \\ \beta_{i1}^{t} \; \beta_{i2}^{t} \ldots \ldots \ldots \beta_{ij}^{t} \ldots \ldots \beta_{im}^{t} \end{bmatrix} \qquad (6)$$

QEA structure

In the initialization step of QEA, [αt, βt] T of all q0 are i j i j with equal probability. The next step makes a set of binary instants; xt by observing Q(t)={q^t , q^t , ..., q^t } states, where Table 1: Lookup Table of Δθ, the rotation gate. xi is the ith bit of the observed binary solution and bi is the ith bit of of q-bit population. Each binary instant, xt of length m, is formed by selecting each bit using the probability of q-bit, give some measure of its fitness. The initial best solution {f(x^t)} is then selected and stored from among the binary instants of X(t). Then, in 'update' Q(t), quantum gates U update this set of q-bit individuals Q(t) as discussed below. This process is repeated in a while loop until convert- j^{th} q-bit value of ith quantum individual in generation t, [αtt gence is achieved. The appropriate quantum gate is usually designed in accordance with problems under consideration.

Quantum gates assignment

Several quantum perturbation operators [2,11,20,23] have been proposed for steering the quantum individuals during the optimization. These operators act like the movement operator in the particle swarm optimization algorithm (PSO). Like the PSO algorithm, previously identified good solutions are selected as a guideline for the current individuals to adjust their positions in the search space. However, unlike the PSO algorithm in that the best individual in the population directly leads the other individuals, in the QEA the recently explored good solutions steer the individuals through increasing or decreasing their probabilities. More specifically, first, in the migration operator, the values of all individuals are replaced by those of the best individual, and then in the update operator each individual set its probability through these recently good obtained values. Here, we use the rotation gate as an updating-quantum individual procedure. Specifically, a q-bit individual q_i^t is as follows. The Where Δθ is rotation angle controlling the speed of convergence, which is determined from Table 1. Reference [10] shows that these values for Δθ have better performance.

The Proposed Algorithm

The proposed algorithm is a memetic algorithm that is a hybridization of the quantum evolutionary algorithm with the multitrilateration [17] and the baldwinian local search procedures for solving the fine grained localization problem in WSNs. It also uses a mapping procedure to convert the binary solutions obtained by the

proposed algorithm to non-anchor node positions. First, we take look the solution representation in the proposed algorithm and then describe the algorithm in detail.

Solution representation

Sensor nodes locations in the proposed network model are encoded in the real value scheme, so because of the binary representation of the solutions of the proposed algorithm, we cannot directly apply the proposed algorithm to solve the localization problem, and we need to convert the binary solutions of the proposed algorithm to the real-coded solutions. To do this, in this paper we propose a binary-to real mapping procedure that converts the solutions obtained by the proposed algorithm to their corresponding real-coded solutions. Using this, we can evaluate the proposed algorithm performance. In particular, let us consider x_1, x_2, ..., x_n as solutions of the proposed algorithm where x_1 is denoted in Figure 1.

Figure 2a shows how the position of a node converts to its corresponding binary form where its position is divided into 8 integers, and each of which is then transformed to the binary string of length 4 by iteratively being divided by 2. To convert the binary form of solutions to the corresponding real-number form, Figure 2b illustrates an example of binary- to-real conversion. As shown in Figure 2a, we ignore the values in the left side of the point of the position values because the positions of sensor nodes are in the range of (0-1).

In general, the algorithm is composed of four specific parts: ML, real-to-binary and binary-to-real, quantum evolutionary and local search procedures. First, in order to give the quantum individuals a good location guideline that can be used during the search process, the algorithm employs the ML procedure, which is applied to the best personal observed solutions called B. The ML procedure is an approximation process attempting to provide good initial locations for non-anchor nodes. Second, to make the binary observed solutions suitable for the real-domain localization problem in WSNs, the algorithm applies the real-to-binary procedure to convert the binary solutions to the real solutions and the binary-to-real procedure to do the opposite. The algorithm also utilizes the Solis Wets' local search (SW-LS) [26] in the Baldwinian scheme for the best observed solution at each specific generation. We called it the Baldwinian evolution because similar to the Baldwin effect in genetic algorithms, it does not have a direct effect on the genotypes of the individuals. Instead, it improves the observed solutions indirectly by sending back the recently local searching-improved values to the population, then steering them through the rotation gate. The algorithm, finally, uses the rotation gate as a Q-gate for updating quantum individuals in Q^t. The framework of the proposed algorithm is represented in Figure 3.

Algorithm 1

The Proposed Algorithm

Proposed Algorithm

Begin

t=0

1. initialize Q^0

2. make X0 by observing the states of Q_0

3. Make C0 using MT procedure

4. Initialize B0 using real-to-binary procedure

x_i	b_i	f (x)>f (b)	Δθ
0	0	FALSE	0
0	0	TRUE	0
0	1	FALSE	0.01π
0	1	TRUE	0
1	0	FALSE	-0.01π
1	0	TRUE	0
1	1	FALSE	0
1	1	TRUE	0

Table 1: Lookup Table of Δθ, the rotation gate. x_i is the *i*-th bit of the observed binary solution and b_i is the ith bit of the best found binary solution.
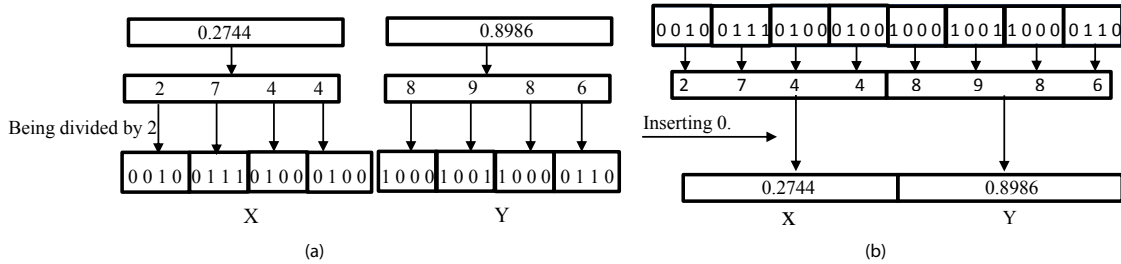
Figure 2: An example of conversion scheme. a) Converting the position of a node to its corresponding binary form. b) Converting the binary form of solutions to its corresponding real-coded one.
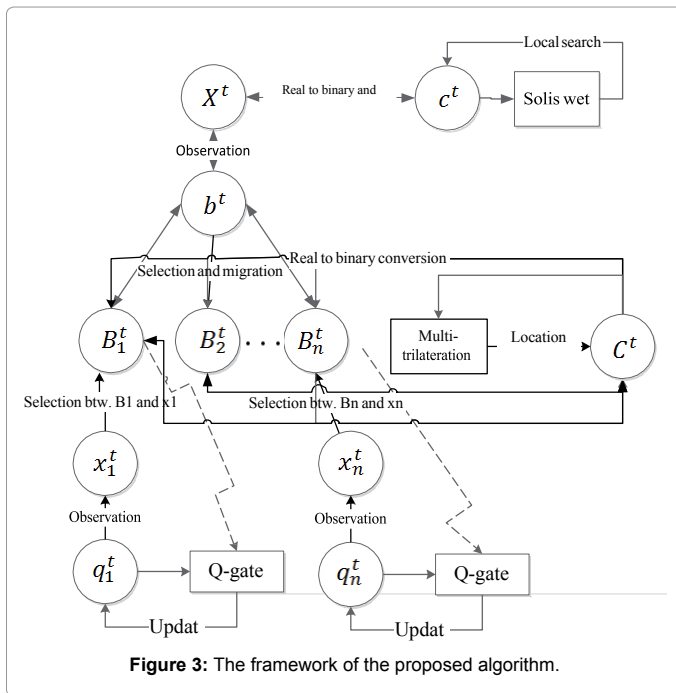

Figure 3: The framework of the proposed algorithm.

5. while not termination condition do

begin t=t+1

6. make X^t by observing the states of Q^{t-1}

7. make real-code solutions E^t using the binary-to-real procedure

8. evaluate E^t

9. among X^t and B^{t-1}, the best individuals are stored in B^t

10. store the best individual among B^t in b^{t-1} in b^t

Note that the proposed algorithm possesses a population of two-dimensional quantum individuals

$Q^t = \{ q_{11}^t, q_{12}^t, \ldots q_{1n}^t, q_{21}^t, q_{22}^t, \ldots q_{2n}^t \}$ Where t is the current iteration and n is the size of population. It also uses two other populations: bi In order to apply the estimated locations of non-anchor in Q0 is initialized without as, nary population Xt, real-coded population Et. The description of the proposed algorithm is in the following.

1. This process is carried nodes given by the multi-trilateration procedure to B^t, we need to convert the positions of sensor nodes to its corresponding binary form. To do this, we use a real-to-binary procedure simply turning the positions of sensor

nodes into the binary solutions in B^t. Note that since k is the dimensions of the area in which the nodes are located.

$\alpha_{kij}^0 = \sqrt{\dfrac{2}{2}}$, for k=1,2, i=1, ..., n, and j=1, ..., m (8)

2. In this step, the binary solutions $X^0 = \{ x_{11}^0, x_{12}^0, \ldots x_{1n}^0, \ldots x_{21}^t, x_{22}^t, \ldots x_{2n}^t \}$ each 4-digit real number in C^t representing the x or y- position of a node is turned into the binary string of length 16, the size of the binary population B^t is 16 times larger than that of C^t. So the size of C^t is $16 \times 2 \times n \times m$.

$x_{kij}^t = \begin{cases} 0 & if \ R(0,1) < |\alpha_{kij}^t|^2, \\ 1 & otherwise \end{cases}$

3. In order to find good initial locations for the non-anchor nodes, the algorithm first uses the ML procedure. It then copies the results into C^t, the real-coded form of B^t. The MT procedure is performed as follows. First, all nodes are divided into two sets, the set of anchor nodes, A, and the set of non-anchor nodes, F. Then, the non- anchor nodes in F are localized by the trilateration technique [17]. To do this, we require at least three anchor nodes. If a non-anchor node has at least three neighbours converting the binary solutions in Bt to real-coded solutions in Ct is used. In this process we do the opposite actions performed in the real-to-binary procedure.

4. In order to apply the estimated locations of non-anchor nodes given by the multi-trilateration procedure to B^t, we need to convert the positions of sensor nodes to its corresponding binary form. To do this, we use a real to binary procedure simply turning the positions of sensor nodes into the binary solutions in B^t. Note that since each 4-digit real number in C^t representing the x or y position of a node is turned into the binary string of Length 16, the size of the binary population B^t is 16 times larger than that of C^t. So the size of Ct is $16 \times 2 \times n \times m$.

5. The while loop is finished until the maximum number of iterations MI is reached.

6. This step is carried out like Step 2.

7. In order to make the observed solutions suitable for the localization problems in WSNs, a mapping procedure converting the binary solutions in B^t to real-coded solutions in Ct is used. In this process we do the opposite actions performed in the real-to-binary procedure.

8. To evaluate the individuals, they must be turned into real-coded form. So, we convert X^t to real-coded form E^t which is then evaluated using Formula 2.

9. For each individual the best place in the search space the

individual has reached during the optimization is stored in B^t. For example, if Ct is better than $E(t-1)_i$ in terms of the CX value, its corresponding binary-formed individual in X^t is stored in B_i^t.

Estimate the location of the non-anchor node. By iteratively using the trilateration technique, each non-anchor sensor node in F is localized, and moved to the set A. This technique is iterated until the non-anchor nodes in F, having at least three known node neighbors, are found.

10. Among all individuals in B^t, the best is selected and stored in b^t. In addition, c^t, the best corresponding real- coded solutions, are all replaced by the best individual in E^t.

11. Update Q^t.

12. The SW-LS is performed periodically in the proposed algorithm. If the pre-specified periodic time is reached (that is 100 generations), the SW-LS process is initiated.

13. To perform the SW-LS on the best solution, we need to convert b^t to its corresponding real-coded form ct and then after performing the local search, we reconvert and replace the best found solution to the binary form if reaching better CX value. The SW-LS is a stochastic hill- climber, using an adaptive step size to discover promising area of the search space. It starts from c^t and then making several randomized moves toward the better nearby solutions in order to find a better area of the search space. After several successful and unsuccessful moves, the algorithm adjusts its step size, trying to find better position in the search space. After specific amount of iterations (equal to 100). If five successful steps are taken, ρ is multiplied by 2; if three unsuccessful steps are taken, it is multiplied by 2. Figure 4 shows how the SW-LS work. As seen, Success, Fail, Eval are variables, counting the number of success and failure obtained during the SW-LS process and the number of function evaluations spent on the LS process, respectively; s′ and s″ are 2-dimensional arrays, holding the position of s after positive and negative perturbations, respectively; bias is an array holding the history of search during the optimization and max Eval is the maximum number of FEs assigned to the SW-LS at each local-search period. Note that, after performing the local search, we need replace c^t with s, providing it has reached better CX value. Then, b^t is re- placed by c^t obtained from the real-to-binary procedure.

14. In the proposed algorithm, migration is performed glob- ally, tending to accelerate the convergence rate of the population. In the migration process, the binary and real values of the best solution (b^t and c^t) are copied into all the best binary and real solutions (B^t and C^t), respectively.

Simulation Results

The aim of this section is to evaluate the performance of the proposed algorithm when solving the node localization problem in WSNs. In our simulations, first, we have constructed 10 random network topologies named as TOP0-TOP9 with 4 various transmission ranges of the nodes. Then, we perform the proposed algorithm on these proposed net- works. More specifically, we first describe the randomly generated network topologies in terms of the number of nodes versus neighbourhood cardinality and the number of anchor nodes versus the number of non-anchor nodes. Then, we investigate how applying

the BL and MT procedures can help the proposed algorithm boost its performance. Finally, we perform a comparison between the proposed algorithms with 6 existing optimization approaches on the proposed network topologies.

In the network topologies the noise factor NF is %0.1 and the transmission range of the nodes, which controls the connectivity of the nodes is r=(0.13, 0.15, 0.18, 0.22), and the number of anchor and non-anchor nodes are 20 and 180, respectively. The nodes are uniformly placed in a square region of $[0, 1] \times [0, 1] \subset R2$.

Topology Setup: Figure 5 demonstrates the mean percentage of all nodes (anchor and non-anchor nodes) on 10 random network topologies against the neighbourhood cardinality of the nodes for 4 different connectivity ranges.

As shown in Figure 5, the more communication ranges the more neighbourhood cardinality. For instance, for r=0.13, about 13% of nodes have 10 adjacent nodes, and for r=0.15, 0.18 and 0.22, roughly 8, 4 and 1 percent of nodes have 10 neighbouring nodes. Furthermore, it can also be observed that the highest node cardinality for r=0.13, 0.15, 0.18 and 0.22 are 17, 20, 27 and 38, respectively, and no node for r=0.13, 0.15, 0.18 and 0.22 have more than 9, 12, 17 and 29 neighbouring nodes, respectively.

The impact of applying the local search to the performance of the algorithm

The local search procedure has great impact upon the pro- posed algorithm whether it is performed as a pre-processor procedure (the MT procedure) in the location initialization process or as an interleaved procedure (the BL procedure) in the evolutionary process. In this section, we investigate its effect on the proposed algorithm. First, we look at the effect of the ML procedure on the CX values. Second, we examine the effect of the BL local search on the CX values. To this end, we first compare the QEA+MT with QEA on ten randomly created network topologies described in the previous subsection, and then we compare the proposed algorithm (QEA+MT+BL) with the QEA+MT on the network topologies. Figure 6 shows the CX trends of the QEA+MT and QEA on the first topology (TOP0) and r=0.13.

As shown in Figure 7, for r=0.13, about 55% of non- anchor nodes have only one adjacent anchor node, and for r=0.18 and r=0.22 roughly 40 and 15 have one neighbouring anchor node. Furthermore, about 27%, 5% and 1% of non- anchor nodes have no nearby anchor nodes. This clearly indicates that solving the localization problem on the proposed network topologies is very demanding.

As shown in Figure 6 the QEA+MT reaches better CX values much faster than QEA. As a result, we can suggest that using the MT procedure can boost the ability of the algorithm to find better results. Table 2 summarize the results of the QEA+MT and QEA on 10 network topologies with 4 connectivity ranges. The best results are typed in bold rest cases (7 cases), for r=0.15, 0.18 and 0.22, interestingly, the QEA+MT+BL performs best on 7 cases, and the rest (3 cases) are gained by the proposed algorithm without the BL. This may suggest that the BL procedure has a positive effect on the performance of the algorithm. As represented in Table 2, the QEA+MT offer best results for all cases.

Comparison against existing optimization techniques

In order to evaluate the performance of the proposed algorithm on randomly generated networks, we compare it with the SAL [15], GAL
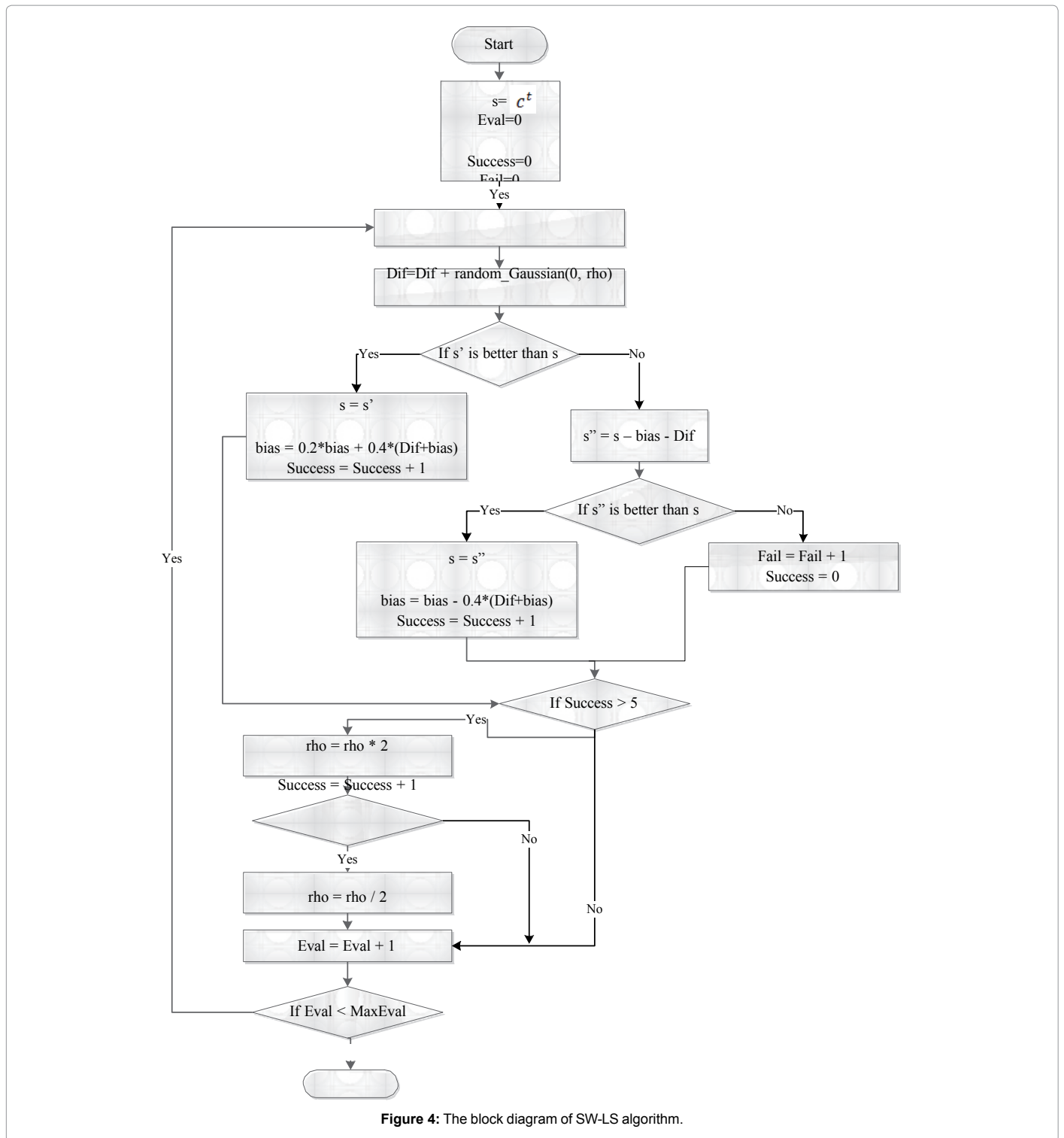
Figure 4: The block diagram of SW-LS algorithm.

[43], ICA [30], PSO [5], TSA [28] and PAES [34] on the ten randomly generated network topologies. We use the best parameter values for all the algorithms recommended in [5,15,28,30,34] and represented in Table 3.

As shown in Figure 8, the proposed algorithm (QEA+MT+BL) reaches better CX values than QEA+MT. We can also see the CX values of the proposed algorithm start to rapidly decrease after each 100 generation in a step-shaped fashion. The reason behind such behaviour is that the BL procedure in the algorithm is reactivated after each 100 generation. Table 4 summarizes the proposed algorithm's performance with/without the BL procedure on the ten network topologies using four communication ranges of nodes. The best results are typed in bold. According to Table 3, for r=0.13 the proposed algorithm with the BL

Connectivity range	r=0.13				r=0.15				r=0.18				r=0.22			
Algorithm	QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
TOP0	20	13	16	9.53	2	0.59	2.13	0	1.5	0.51	2.03	0	0.01	0	0.02	0
TOP1	1.2	0.1	1.9	0.43	0.4	0.08	0.54	0.01	0.5	0	0.52	0	0.06	0	0.11	0
TOP2	30	10	24	12.1	0.03	0.01	0.05	0	0.7	0.4	0.22	0	0.22	0.1	0.1	0
TOP3	5.6	1.4	5.1	0.36	5.93	1.61	3.38	0.87	1.9	1.05	1.13	0	0.34	0.3	0.46	0
TOP4	162	70	160	65.6	0.18	0.07	0.19	0.03	0.3	0.06	0.7	0	0.12	0	0.14	0
TOP5	133	32	107	28	79.3	9.51	66.8	15	0	0.01	0.04	0	0.06	0	0.11	0
TOP6	134	27	165	47.9	0.06	0	0.06	0	0.2	0	0.22	0	0.34	0.3	0.31	0
TOP7	147	38	112	28.2	0.15	0.04	0.29	0.05	0.7	0.02	1.31	0	0.76	0.1	0.16	0
TOP8	23	12	26	22.4	0.17	0.03	0.22	0.04	0.1	0.01	0.08	0	0.05	0	0.09	0
TOP9	21	25	2.8	1.65	0.03	0.01	0.05	0	0.2	0.16	0.17	0	1.74	0.7	2.62	0.7

Table 2: The mean and standard deviation of the LE values obtained by the QEA and QEA+MT on the ten network topologies and for 4 different communication ranges over 20 runs.
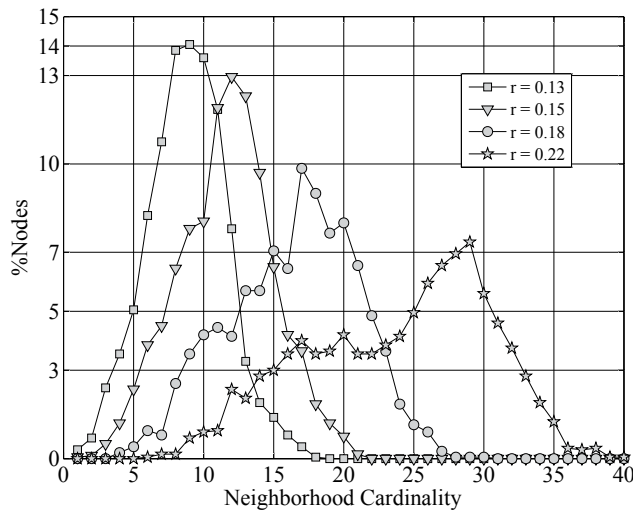
Figure 5: The mean percentage of sensor nodes versus neighborhood cardinality for 4 communication ranges.
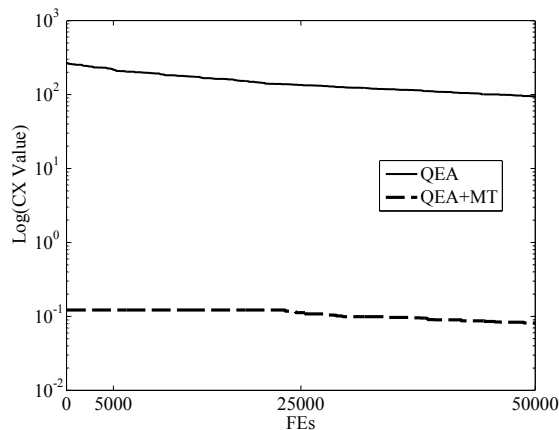
Figure 6: CX trends of the QEA and QEA+MT where *TOP*0 and *r*=0.13.

procedure performs best only on 3 cases and the proposed algorithm without the BL procedure on the To make a fair comparison, the termination condition for all the algorithms is set to 50000 function evaluations (FEs); that is, for the population-based algorithms (GAL, PSOL, ICA and the proposed algorithm), the size of the population is

set to 50, the maximum number of iterations for GAL and PSOL is set to 1000 and for the single-solution algorithms (SAL, TSA and PAES), on the other hand, the maximum number of iterations is set to 50000. Note that because of involvement of the BL local search, the maximum number of iterations for the proposed algorithm is set to 980 so the

	Parameter	Value	Description
	T0	0.1	The initial temperature
	Nf	50,000	The number of function evaluations
	P	10	The number of iterations in the inner loop
SAL	Q	2 ∗ n	The number of iterations in the outer loop
	D0	0.1	The initial move distance
	α	0.8	The cooling factor
	β	0.94	The distance-declining factor
	Mr	0.85	The mutation rate
GAL	Ir Sp	0.75	The crossover rate
		50	The size of population
	MI	1,000	The maximum number of iterations
	W	0.7	The mutation rate
PSOL	C	1.494	The crossover rate The size of population
	Sp	50	
	MI	1,000	The maximum number of iterations
	NI	8	The number of empires
ICA	NC	50	The number of countries
	MI	1,000	The maximum number of iterations
	As	20	Archive size
	Nr	5	Number of regions
PAES	Nf	50,000	The number of fitness evaluations
	PM	0.9	Node mutation probability
	PN	0.3	Node rigid translation probability
	T0	0.1	The initial temperature
	Nf	50,000	The number of function evaluations
TSA	P	4	The number of iterations in the inner loop
	Q	2 ∗ n	The number of iterations in the outer loop
	D0	0.1	The initial move distance
	α	0.8	The cooling factor
	β	0.94	The distance-reducing factor
	μ	0.2	The parameter used for threshold value calculation
	λ	0.1	The parameter used for threshold value calculation
	γ	0.05	The parameter used for threshold value calculation
	Sp	50	The size of population
	Δθ	0.01	Rotation angle of each Q-bit
QEA+MT+BL	MI	960	The maximum number of iterations
	Lp	100	The local search period
	Mp	100	The migration period
	IL	100	The intensity of local search

Table 3: Parameter configuration of the SAL, GAL, PSOL, ICA, TSA, PAES and QEA+MT+BL.

overall FEs for the proposed algorithm is 50000. Moreover, due to the new termination condition (50000 FCs); we have to ignore the other termination conditions of the algorithms. For instance, for the SAL, we ignore the Tf (the final temperature of the SAL algorithm). Randomly generated network topologies and four different communication ranges. The best results are typed in bold.

It can be observed in Table 5 for r=0.13, 0.15 the pro- posed algorithm achieves best for all network topologies (except TOP4 and TOP5); after that, the PAES, TSA and SAL are the second to forth best algorithms. Furthermore, for r=0.18 and 0.22 the QEA+MT+BL maintains its superiority over the other algorithms, and it achieves the best for all network topologies. It is also shown that as the connectivity range increases, the performance of the PAES and TSA significantly changes. For r=0.13 the PAES is superior over the TSA on 9 cases and for r=0.15 on 7 cases. However, for r=0.18, 0.22, the TSA is better than PAES on 8 and 10 cases, respectively.

Intuitively, a combination of an approximation procedure such as the multi-trilateration technique with an optimization process such as the SA or QEA induces better performance (see the results of the proposed algorithm and TSA). It is also observed that the pure optimization algorithms could not yield well results (see the results of the GAL, ICA and PSOL). To shed more light on the LE values, Figure 9 demonstrates the estimated coordinates of network nodes, obtained by the proposed algorithm and its spins-off as well as the other 6 optimization algorithms on *TOP*0, r=0.13 where black solid stars, rectangles, multiplication signs, straight lines represent the coordinates of anchor nodes, the real positions of non-anchor nodes, the estimated positions of non-anchor nodes and the Euclidean distance between the real and estimated positions of non- anchor nodes, respectively. As shown in Figure 9, the proposed algorithm as well as QEA+MT estimates the positions of non-anchor nodes with
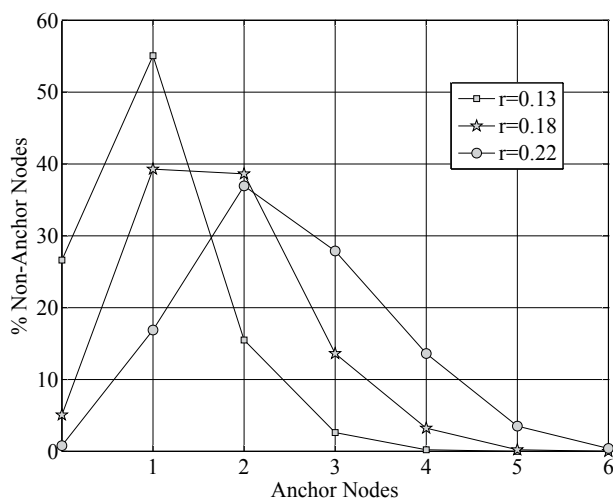
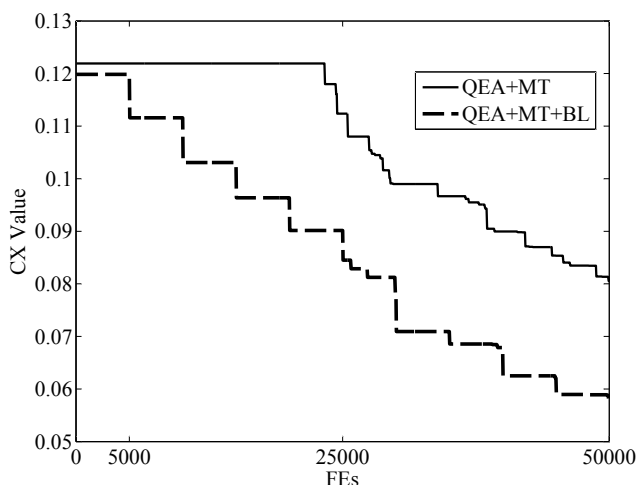**Figure 7:** The mean percentage of non-anchor nodes versus anchor nodes for different communication ranges.

Figure 8: A CX trend of the proposed algorithm with/without the BL procedure where *TOP*0 and *r*=0.13.

Connectivity range	r=0.13				r=0.15				r=0.18				r=0.22			
Algorithm	QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT		QEA+MT+BL		QEA+MT	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std
TOP0	20.2	13	16	9.5	2	0.6	2.1	0	1.5	0.5	2.03	0	0	0	0	0
TOP1	1.17	0.1	1.9	0.4	0.4	0.1	0.5	0	0.5	0	0.52	0	0.1	0.02	0.1	0
TOP2	30.1	10	24	12	0.03	0	0.1	0	0.7	0.4	0.22	0.01	0.2	0.11	0.1	0
TOP3	5.56	1.4	5.1	0.4	5.93	1.6	3.4	0.9	1.9	1.1	1.13	0.01	0.3	0.28	0.5	0
TOP4	162	70	160	66	0.18	0.1	0.2	0	0.3	0.1	0.7	0	0.1	0.03	0.1	0
TOP5	133	32	107	28	79.3	9.5	67	15	0	0	0.04	0	0.1	0.02	0.1	0
TOP6	134	27	165	48	0.06	0	0.1	0	0.2	0	0.22	0	0.3	0.31	0.3	0
TOP7	147	38	112	28	0.15	0	0.3	0.1	0.7	0	1.31	0	0.8	0.11	0.2	0
TOP8	23.2	12	26	22	0.17	0	0.2	0	0.1	0	0.08	0	0.1	0.01	0.1	0
TOP9	21.1	25	2.8	1.7	0.03	0	0.1	0	0.2	0.2	0.17	0	1.7	0.66	2.6	0.7

Table 4: The mean and standard deviation of the LE values obtained by the proposed algorithm with/without the BL procedure on the ten network topologies using 4 different communication ranges.

		TOP0	TOP1	TOP2	TOP3	TOP4	TOP5	TOP6	TOP7	TOP8	TOP9
r=0.13											
QEA+MT+BL	Mean	20.2	1.17	30.14	5.56	161.53	132.57	133.66	147.33	23.16	21.11
	Std.	13.01	0.05	10.34	1.37	69.71	32.2	27.04	37.66	11.83	24.51
SAL	Mean	529.74	621.51	733.39	525.49	667.18	417.63	690.75	602.64	641.03	526.26
	Std.	313.58	180.33	140.37	178.25	231.81	186.79	199.41	156.53	239.66	201.88
PSOL	Mean	1072.78	1331.88	1332.75	1096.95	1202.64	1202.77	1266.55	1385.59	1241.31	1185.91
	Std.	42.6	37.76	32.66	48.87	54.03	50.98	27.79	19.89	40.97	53.16
ICA	Mean	933.71	1058.67	1005.29	1022.28	989.04	1015.56	1007.29	1022.25	1013.05	1016.04
	Std.	29.78	21.41	32.22	29.51	29.61	28.93	30.76	29.75	33.22	32.47
GAL	Mean	2372.32	2392.19	2354.78	2453.45	2336.21	2362.7	2315.94	2302.5	2345.77	2419.54
	Std.	88.97	57.97	54.06	58.34	66.55	45.79	57.26	67.2	67.4	62.97
PAES	Mean	146.63	169.31	268.27	159.96	142.89	97.51	136.97	175.48	205.46	130.19
	Std.	9.07	16.1	48.12	27.19	26.13	15.28	8.36	43.26	44.8	15.76
TSA	Mean	214.78	225.85	380.98	175.4	604.56	350.93	318.41	308.42	191.59	212.49
	Std.	59.43	43.05	48.13	55.49	298.71	158.06	62.97	101.77	44.32	67.53
r=0.15											
QEA+MT+BL	Mean	20.2	1.17	30.14	5.56	161.53	132.57	133.66	147.33	23.16	21.11
	Std.	13.01	0.05	10.34	1.37	69.71	32.2	27.04	37.66	11.83	24.51
SAL	Mean	121.47	208.65	361.92	380.07	182.85	110.04	219.81	294.49	141.35	180.77
	Std.	39.8	57.99	82.85	240.92	56.68	15.89	77.94	21.49	55.22	43.83
PSOL	Mean	770.81	928.91	914.97	924.37	822.08	898.08	926.36	870.32	875.72	909.76
	Std.	28.82	45.24	33.34	29.57	40.33	48.59	28.39	57.77	29.35	42.01
ICA	Mean	680.07	758.84	691.11	700.86	714.67	741.46	715.71	721.18	716.28	715.89
	Std.	18.41	28.83	15.19	12.13	3.93	12.19	22.44	9.44	15.41	17.4
GAL	Mean	1761.14	1790.55	1749.62	1816.37	1738.83	1768.51	1749.29	1711.6	1787.15	1795.84
	Std.	35.73	33.77	42.87	28.33	45.04	43.68	25.88	50	51	41.93
PAES	Mean	75.26	80.54	125.99	68.78	81.32	51.5	73.14	120.89	56.25	84.88
	Std.	18.48	2.09	38.57	3.87	10.32	5.21	2.92	33.5	3.45	8.1
TSA	Mean	105.48	95.78	129.17	88.92	79.62	97.86	103.1	120.5	56.42	60.46
	Std.	19.67	16.62	33.32	31.97	20.83	44.8	12.99	24.12	9.84	22.56
r=0.18											
QEA+MT+BL	Mean	1.51	0.52	0.66	1.91	0.28	0.03	0.22	0.65	0.07	0.17
	Std.	0.51	0	0.4	1.05	0.06	0.01	0	0.02	0.01	0.16
SAL	Mean	44.71	97.05	203.59	89.96	100.58	78.27	121.48	126.6	91.93	87.57
	Std.	26.49	46.47	42.61	72.84	52.15	54.78	59.16	50.74	60.18	70.54
PSOL	Mean	522.94	485.33	484.5	542.44	539.93	556.44	511.28	417.78	508.89	498.82
	Std.	29.4	25.92	23.79	28.08	56.06	34.04	38.11	24.7	38.79	41.48
ICA	Mean	492.67	549.68	518.43	533.92	520.9	535	525.87	532.26	534.25	533.05
	Std.	18.7	16.87	17.79	17.35	14.19	21.92	15.84	18.22	13.17	12.32
GAL	Mean	1200.45	1216.18	1216.11	1271.44	1222.1	1222.11	1206.39	1191.57	1219.66	1251.77
	Std.	36.26	33.63	36.31	35.14	26.6	34.08	38.99	30.72	32.03	20.74
PAES	Mean	53.94	109.39	46.02	42.88	38.32	37.91	39.59	59.98	39.03	41.33
	Std.	3.91	21.08	2.6	2.41	3.02	4.53	1.49	8.29	4.16	3.6
TSA	Mean	24.07	21.84	65.76	14.5	17.64	15.21	41.2	14.9	21.66	18.46
	Std.	18.39	10.24	35.03	9.11	14.18	15.53	10.66	9.32	17.29	9.96
r=0.22											
QEA+MT+BL	Mean	0.01	0.06	0.22	0.34	0.12	0.06	0.34	0.76	0.05	1.74
	Std.	0	0.02	0.11	0.28	0.03	0.02	0.31	0.11	0.01	0.66
SAL	Mean	14.69	46.19	60.46	27.06	31.83	31.87	52.36	43.49	42.78	28.84
	Std.	9.03	35.88	37.36	26.88	27.57	32.39	37.74	30.21	37.42	28.42
PSOL	Mean	281.64	258.66	250.63	336.45	270.81	259.98	294.04	288.52	268.91	376.3
	Std.	17.3	53	30.79	35.3	24.18	32.34	37.36	35.33	23.95	15.95
ICA	Mean	314.27	362.78	349.62	353.97	348.15	358.65	349.76	351.79	360.02	357.22
	Std.	4.8	11.45	10.61	13.02	10.76	13.1	11.46	10.98	11.48	8.36
GAL	Mean	809.4	820.99	808.85	842.4	809.48	826.5	815.05	809.66	822.53	826.95
	Std.	19.55	25.37	16.29	15.43	23.69	20.95	10.96	18.6	11.99	25.95

PAES	Mean	28.49	28.17	33.57	28.84	22.27	23.56	28.75	27.87	30.89	32.02
	Std.	5.81	1.56	10.44	0.81	0.37	0.75	5.3	1.69	3.08	6.21
TSA	Mean	3.65	4.16	9.42	1.19	3.3	0.99	5.9	3.85	8.92	2.41
	Std.	4.37	4.18	9.19	0.78	2.95	1.48	9.07	3.33	7.09	2.43

Table 5: The mean and standard deviation of the LE values obtained by 7 optimization techniques on the ten network topologies and for 4 different communication ranges. The best results are typed in bold.

Figure 9: Location estimates using different optimization techniques on *TOP*0 and r=0.13.
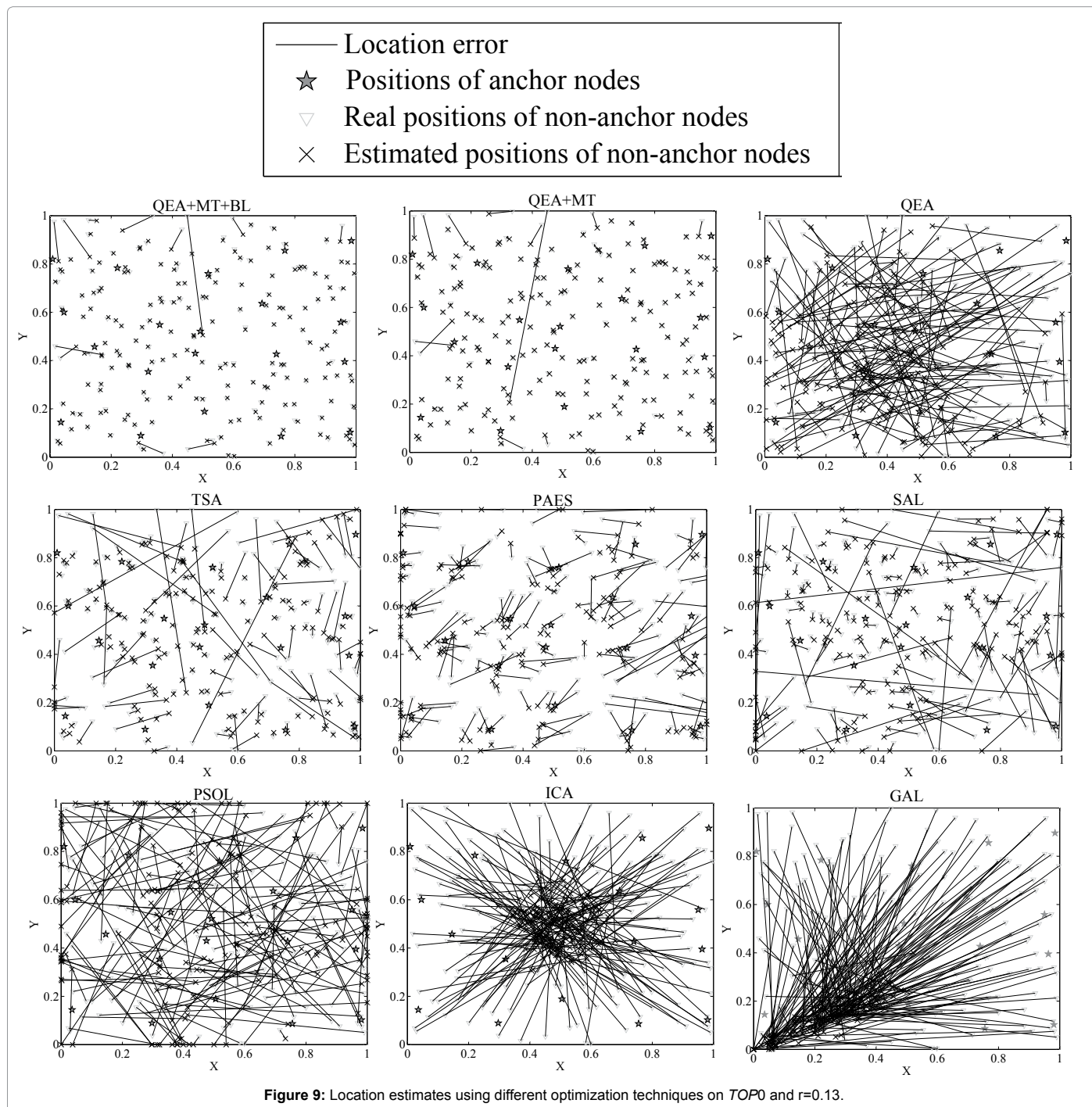
higher accuracy than the other algorithms. After that the PAES and TSA offer good accuracy.

Conclusion

This paper proposed a new memetic algorithm for managing the fine-grained localization problem in WSNs. The memetic algorithm is based on the QEA and a local search procedure in the form of Baldwinian scheme. The QEA in the proposed algorithm improves the explorative

ability and the algorithm, and the local search procedure enhances the exploitation ability of the proposed algorithm, finding local optima more quickly. In particular, the proposed algorithm can be summarized as following aspects. First, to make good initial locations for sensor nodes, the algorithm uses the multi-trilateration procedure iteratively. Second, to enhance the exploitation ability of the algorithm, it also utilizes Solis Wet's local search in the form of Baldwinian scheme. Third, to make the proposed algorithm appropriate for the localization problem a conversion procedure, which converts the real-coded solutions to the binary solutions, is used. The proposed algorithm was compared with six existing optimization techniques on ten randomly created network topologies. The simulation results and comparisons demonstrate superiority of the proposed algorithm in terms of localization error and robustness.

References

1. Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: A survey. Computer Networks 38: 393-422.

2. Baldwin J (1896) A new factor in evolution. Am Nat 30: 536-553.

3. Banimelhem O, Khasawneh S (2012) Gmcar: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks. Ad Hoc Networks 10: 1346-1361.

4. Chuang PJ, Wu CP (2008) An effective pso-based node localization scheme for wireless sensor networks. In: Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008. Ninth International Conference, pp: 187-194.

5. Chung C, Yu H, Wong KP (2011) An advanced quantum-inspired evolutionary algorithm for unit commitment. Power Systems, IEEE Transactions on Evolutionary Computation 26: 847-854.

6. Doherty L, Pister KSJ, El Ghaoui L (2001) Convex position estimation in wireless sensor networks. In: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE 3: 1655-1663.

7. Ganarski P, Blansch A (2008) Darwinian, Lamarckian and Baldwinian (co) evolutionary approaches for feature weighting in k-meansbased algorithms, p: 12.

8. Gong M, Jiao L, Zhang L (2010) Baldwinian learning in clonal selection algorithm for optimization. Information Sciences.

9. Han HK, Kim HJ (2002) Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation 6: 580-593.

10. Han KH, Kim JH (2004) Quantum-inspired evolutionary algorithms with a new termination criterion, epsilon gate and two-phase scheme. IEEE Transactions on Evolutionary Computation 8: 156-169.

11. Hofmann-Wellenhof B, Lichtenegger H, Collins J (2001) Global positioning system: Theory and Practice, 5th edn. Springer.

12. Hong Y, Pen K (2010) Optimal var planning considering intermittent wind power using Markov model and quantum evolutionary algorithm. IEEE Transactions on Power Delivery 25: 2987–2996.

13. Hu F, Wu B (2009) Quantum evolutionary algorithm for vehicle routing problem with simultaneous delivery and pickup. In: Proceedings of the 48th IEEE Conference on Decision and Control.

14. Kannan A, Mao G, Vucetic B (2005) Simulated annealing based localization in wireless sensor network. In: Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on 2: 514.

15. Kannan A, Mao G, Vucetic B (2006) Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In: Vehicular Technology Conference, 2006. VTC 2006 Spring. IEEE 2: 1022 –1026.

16. Karl H, Willig A (2005) Protocols and architectures for wireless sensor networks. John Wiley & Sons.

17. Kim JH, Han JH, Kim YH (2012) Preference-based solution selection algorithm for evolutionary multi-objective optimization.

18. Kim K (2012) A clustering algorithm based on geographical sensor position in wireless sensor networks. In: Sobh T, Elleithy K, Mahmood A, Karim M (eds.) Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications. Springer Netherlands pp: 245-249.

19. Leon F (2012) Real-valued quantum-inspired evolutionary algorithm for multi-issue multi-lateral negotiation. In: Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference, pp: 41-48.

20. Li B, Zhou Z, Zou W, Li D (2012) Quantum memetic evolutionary algorithm-based low-complexity signal detection for underwater acoustic sensor networks. Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on Evolutionary Computation 42: 626-640.

21. Liu B, Wang L, Jin YH (2007) An effective pso-based memetic algorithm for flow shop scheduling. Systems, Man and Cybernetics, Part B: Cybernetics, IEEE Transactions on Evolutionary Computation 37: 18-27.

22. Liu W, Chen H, Yan Q, Liu Z, Xu J, et al. (2010) A novel quantum-inspired evolutionary algorithm based on variable angle-distance rotation. IEEE Congress, pp: 1-7.

23. Liu Y, Yang Z (2010) Location, localization, and localizability. J Comput Sci Technol 25: 274-297.

24. Mao G, Fidan B (2009) Localization algorithms and strategies for wireless sensor networks. Premier Reference Source. Information Science Reference.

25. Molina D, Lozano M, Snchez A, Herrera F (2011) Memetic algorithms based on local search chains for large scale continuous optimisation problems: Ma-ssw-chains. Soft Computing 15: 2201-2220.

26. Nicolau AS, Schirrua R, de Moura Meneses AA (2011) Quantum evolutionary algorithm applied to transient identification of a nuclear power plant. Progress in Nuclear Energy 53: 86-91.

27. Niewiadomska-Szynkiewicz E, Marks M (2009) Optimization schemes for wireless sensor network localization. Int J Appl Math Comput Sci 19: 291-302.

28. Patwari N (2005) Location estimation in sensor networks. Ph.D. thesis, Citeseer.

29. Sayadnavard M, Haghighat A, Abdechiri M (2010) Wireless sensor network localization using imperialist competitive algorithm. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference 9: 818-822.

30. Sinha N, Hazarika K, Paul S, Shekhar H, Karmakar A (2010) Improved real quantum evolutionary algorithm for optimum economic load dispatch with non-convex loads. Springer Lecture Notes in Computer Science 6466: 689-700.

31. Sun L, Guo J, Lu K, Wang R (2007) Topology control based on quantum genetic algorithm in sensor networks. Frontiers of Electrical and Electronic Engineering in China 2: 326-329.

32. Tayarani MH, Akbarzadeh T (2008) A sinusoid size ring structure quantum evolutionary algorithm. In: IEEE Conference on Cybernetics and Intelligent Systems.

33. Vecchio M, Lpez-Valcarce R, Marcelloni F (2012) A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks. Applied Soft Computing 12: 1891-1901.

34. Wang Z, Li S, Cai Q, Su S, Liu M (2009) A fast watermarking algorithm based on quantum evolutionary algorithm. In: IEEE International Conference on Intelligent Computing and Intelligent System.

35. Wanga Y, Fenga X, Huanga Y, Pub D, Zhoua W, et al. (2007) A novel quantum swarm evolutionary algorithm and its applications. J Neurocomputing 7: 633-640.

36. Wyatt D, Bull L (2008) A memetic learning classifier system for describing continuous-valued problem spaces. Springer Berlin Heidelberg 166: 355-395.

37. Xiao J (2009) Improved quantum evolutionary algorithm combined with chaos and its application. Advances in Neural Networks 5553: 704-713.

38. Xue F, Kumar PR (2006) On the coverage and connectivity of large random networks. IEEE/ACM Trans Net 14: 2289-2299.

39. Yao X, Wang F, Padmanabhan K, Salcedo-Sanz S (2005) Hybrid evolutionary approaches to terminal assignment in communications networks. Springer Berlin Heidelberg 166: 129-159.

40. Yuan Q, Qian F, Du W (2010) A hybrid genetic algorithm with the Baldwin effect. Information Sciences 180: 640-652.

41. Yun S, Lee J, Chung W, Kim E, Kim S (2009) A soft computing approach to localization in wireless sensor networks. Expert Syst Appl 36: 7552-7561.

42. Zhang C, Zhang Y, Fang Y (2009) Localized algorithms for coverage boundary detection in wireless sensor networks. Wireless Networks 15: 3-20.

43. Zhang Q, Wang J, Jin C, Ye J, Ma C, et al. (2008) Genetic algorithm based wireless sensor network localization. Natural Computation, pp: 608-613.