



Title	Evolutionary approach on connectivity-based sensor network localization
Author(s)	Qiao, D; Pang, GKH
Citation	Applied Soft Computing, 2014, v. 22, p. 36-46
Issued Date	2014
URL	http://hdl.handle.net/10722/202819
Rights	NOTICE: this is the author's version of a work that was accepted for publication in Applied Soft Computing. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Applied Soft Computing, 2014, v. 22, p. 36-46. DOI: 10.1016/j.asoc.2014.04.019

Evolutionary approach on Connectivity-based Sensor Network Localization

Dapeng Qiao, Grantham K.H. Pang

Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

Email: {dpqiao, gpang}@eee.hku.hk

Abstract

The sensor network localization based on connectivity can be modeled as a non-convex optimization problem in mathematics. It can be argued that the actual problem should be represented as an optimization problem with both convex and non-convex constraints. A two-objective evolutionary algorithm is proposed which utilizes the result of all convex constraints to provide a starting point on the location of the unknown nodes and then searches for a solution to satisfy all the convex and non-convex constraints of the problem. The final solution can reach the most suitable configuration of the unknown nodes because all the information on the constraints (convex and non-convex) related to connectivity have been used. Compared with current models that only consider the nodes that have connections, this method considers not only the connection constraints, but also the disconnection constraints. As a MOEA (multi-objective evolution algorithm), PAES (pareto archived evolution strategy) is used to solve the problem. Simulation results have shown that better solution can be obtained through the use of this method when compared with those produced by other methods.

Keywords: wireless sensor network, localization, connectivity, evolutionary algorithm, genetic algorithm, non-convex constraints

1. Introduction

Position estimation is necessary in many applications such as remote patient monitoring, package and personnel tracking, environment monitoring and wildlife habitat monitoring. In these systems, there could be hundreds or even thousands of low-cost sensor nodes, which can take some simple measurements. Based on either the signal strength or the connectivity among the nodes, we would like to estimate the location of these nodes in a wireless sensor network. It is necessary to accurately localize the sensors in order to measure data which is geographically meaningful. This localization issue has been studied by many researchers and there are many different methods and algorithms [1–4] dealing with this situation.

For applications like automatic guidance, and wildlife habitat tracking, GPS-like devices are widely used. However, GPS devices are expensive and inefficient on power consumption [1]. Moreover, GPS devices are useless in indoor conditions such as patient monitoring, package and personnel tracking. Thus, in sensor networks with a large number of sensor nodes, attaching a GPS device to each node is not practical. In most cases, there are only a few nodes with known positions in the whole sensor network, while others are unknown. The only information between the known nodes and the unknown nodes is the communication among them, which can also include information on connectivity.

In a typical sensor network, a few nodes have known positions, and they are called the anchors. However, the positions of the majority of the nodes need to be estimated using their

relationships to the anchors and other unknown nodes. Based on whether the distances between nodes in a sensor network are known or not, the localization methods can be grouped into two categories: range-based and range-free. Range-based methods are for situations where distances between each pair of nodes are estimated or measured. The information is then communicated to a centralized station in the sensor network and algorithms such as MDS [3] compute the location of each sensor in the network. Usually, the distance between each pair of nodes is estimated by the signal strength received between them, and this information is very noisy in practice. On the other hand, range-free methods, which are also called connectivity-based methods, assume that the distances between any two nodes are unknown. However, connectivity information between them is known. If the distance between any two nodes in the network is within a range, connectivity between the two nodes is said to be established. Although the actual distance is not known, this would provide many connection-imposed proximity constraints to the problem. These connectivity-based methods only require very simple and low-cost hardware. Yet, they can give adequate position estimation based on just connectivity information among the nodes. This paper will focus on the various connectivity-based methods for localization and propose how improvements can be made based on the previous methods.

2. Background and motivation

The solutions of the connectivity-based localization problem can fall into two categories. The first class of methods tries to find the number of direct connections between two nodes. In other words, the number of hops from one node to another node needs to be calculated. Hence, they use the hop count to roughly represent the distance between two nodes. The centroid method [2], the approximate point in triangulation (APIT) [3], the multidimensional scaling–MAP (MDS–MAP) [4], DV-Hop [5] all belong to this class. The other class of methods models the connectivity-based localization problem as a constrained optimization problem. The connectivity information becomes the constraints that the optimization result must satisfy. For example, convex position estimation (CPE) [6] selects the convex constraints and formulates the problem as a convex optimization.

Terwilliger et. al. [7] and Zhang et. al. [8] both use evolutionary algorithm to tackle the localization problem in which the distances among nodes are known. The target of the evolutionary algorithm is to minimize the difference between the known distances and the distances based on estimated nodes position. The variables of the evolutionary algorithm are the coordinates of all the unknown nodes. Tam et. al.[9, 10] use genetic (GA) /evolutionary algorithm in triangulation to estimate the position of a node based on the hop counts to its three nearest anchors. This may be considered as a kind of connectivity-based localization method. However, in their method, there is a GA for estimating the position of each unknown node. The computation complexity is therefore small because the scale of the GA is small (population<30). In each GA, only part of the population with better performance is used in computation, which also decreases the computation cost. Hence, the evolutionary algorithm on connectivity-based localization by Tam is only on the triangulation process, and not on the whole problem.

Actually, the connectivity information includes whether any two nodes are connected or not. If we just consider the connections between nodes, the problem is just a convex optimization problem. Nevertheless, the non-connection situations which indicate two nodes are not within a certain range lead to non-convexity. Therefore, for a complete solution, the problem should be

modeled as a non-convex constraint satisfaction problem if all the connectivity constraints are considered.

In this paper, the connectivity information includes the connection and non-connection constraints among the nodes. An optimization algorithm is used to decrease the number of violated connectivity (i.e. the connection which is mistaken as non-connection, and the non-connection which is mistaken as connection) to be zero. Two objectives are implemented. The first is related to the number of violated connectivity, which is an integer. The second objective relates to the summation of differences of all the violated connectivities. The second objective aims to supply the direction when the first objective is being minimized. Without the second objective, the problem will become an integer optimization problem, which is difficult to converge without a suitable direction in searching for a minimum. On the other hand, the first objective helps to avoid the local minimum of the second objective.

Multi-objective evolutionary algorithms (MOEAs) can be utilized to minimize the two objectives during the optimization process. MOEA is first used in sensor network localization by Massimo et. al. [11]. Massimo et. al. assume the distances among the nodes can be obtained by some measurement such as time of arrival (TOA), received signal strength (RSS), etc. An optimization problem is then established based on distance measurement and the anchors positions. It is then tackled by the Pareto Archived Evolution Strategy (PAES), which is a well-known MOEA and often used as benchmarks of other MOEAs. In their paper, the first objective is the squared error between the inter-node distances corresponding to the candidate geometry and the measured data. The second objective counts the violated connectivity. Their result shows that on distance-based localization PEAS outperform SAL (simulated annealing) in accuracy.

A two-objective optimization problem is also proposed in this paper, but the scenario is changed to connectivity-based localization. To the best of our knowledge, there has been no work focused on handling the convex and non-convex constraints in a connectivity-based localization problem. Current algorithms such as the centroid method or multidimensional scaling (MDS) also consider the connections among the nodes, and the semi-definite programming (SDP) method handles only the convex constraints of the problem. The accuracy of the solutions obtained by these methods is usually low. MOEA has the potential to show its advantage in accuracy when compared with the current connectivity-based algorithms. This paper will model the complete connectivity information and demonstrate a method which applies MOEA for tackling this kind of problem.

3. Related work

Current connectivity-based localization algorithms on sensor networks include the centroid method [2], the approximate point in triangulation (APIT) [3], the multidimensional scaling–MAP (MDS–MAP) [4], DV-Hop [5] and the convex position estimation (CPE) [6]. They all only consider the convex constraints but ignore the non-convex constraints. This section briefly describes those algorithms, which will be used to compare with the result of our evolutionary algorithm.

3.1. The centroid method

Centroid localization is probably the earliest and simplest approach. A proximity-based and coarse approach is proposed by Bulusu and Heidemann [2]. Every unknown node receives several nearby anchors' information. The location information of the anchors is used, and the

estimated location of the unknown node is assumed to be the average of the location of all the nearby anchors. The following formula is used:

$$(X_{est}, Y_{est}) = \left(\frac{X_{i1} + \dots + X_{ik}}{k}, \frac{Y_{i1} + \dots + Y_{ik}}{k} \right)$$

where $(X_{i1}, Y_{i1}) \dots (X_{ik}, Y_{ik})$ is the location of the k anchors that the unknown node i can contact; (X_{est}, Y_{est}) is the estimated location of node i .

3.2. Approximate Point-In Triangulation test (APIT)

APIT is another method first proposed by He et al. [3]. The area around an unknown node is split into several pieces by some triangles with their vertexes being anchors it can hear. A method is developed to check whether the unknown node is in a particular triangle or not. For each unknown node, all the triangles formed by the anchors are investigated to check whether the unknown node is within. The possible position of the unknown node must be in the common area of the triangles that contain the node, but not in the area of any triangle that does not contain the node. Therefore, the estimated location of the unknown node is set as the center of gravity (COG) of this possible area.

In the process of checking whether a node is in a triangle, the unknown node utilizes information obtained by its nearby unknown nodes. The strength of the signal transmitted by the anchors is also used in this process. In order to improve accuracy, APIT requires the anchors to have a larger communication range than normal nodes. In their simulation, they have used anchors that have a communication range that is ten times larger.

3.3. Multi-dimensional scaling (MDS)

The basic MDS method [4, 12] can estimate the positions of all the unknown nodes by using the distance information between any two nodes. An extension of MDS [4, 13] for the connectivity-based localization problem has also been developed. First, a rough estimate of the relative node distance is obtained based on hop count information [14]. One hop is one direct connection between two nodes. The hop count between two nodes roughly represents the distance. Then, the relative positions are calculated by Singular Value Decomposition on the estimated distance information matrix. Finally, absolute positions of the unknown nodes are estimated based on the relative positions and the positions of the anchors. The computation complexity of this method is about $O(n^3)$ time for a sensor network of n nodes.

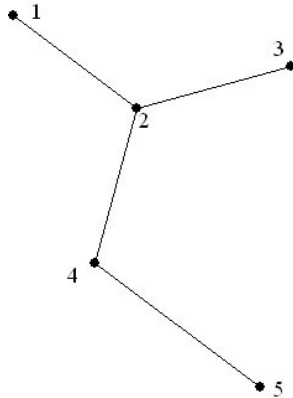


Figure 1. Hop count is used in modified MDS

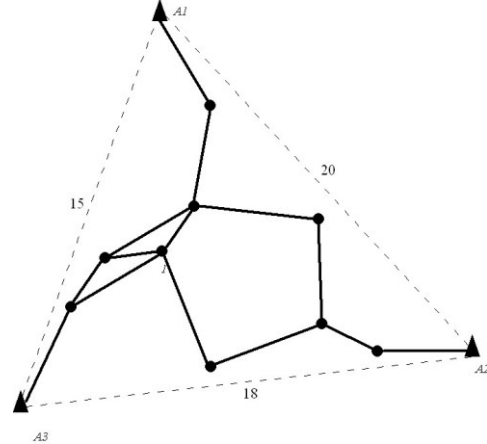


Figure 2. A sensor network example for DV-hop

An example is shown in Figure 1. The connections are represented by lines between two nodes. The hop count between node 1 and node 5 is the number of connections between them, which is 3. Hence, the hop count between any two nodes can be obtained and used in MDS to estimate the locations of the nodes.

3.4. DV-HOP

Another well-known localization algorithm is DV-Hop (distance vector-hop) [5]. The idea of DV-Hop is to transform the distance to all anchors from hops to units of length measurement using the average size of a hop. DV-Hop was first proposed by Niculescu [15], and improved by many researchers. Anchors broadcast their location information to other anchors, and such information will be flooded with the hop count increment. Every anchor knows the hop count from any other anchor, and uses this information to estimate the average hop size. The distance between an anchor and an unknown node is computed by the hop size and the hop count between them. Finally, triangulation is used when an unknown node knows the distances to at least three anchors.

In the example in Figure 2, anchors broadcast their position information to other anchors, and hop counting will be carried out in this process. The minimal hop count from $A3$ to $A1$ is 5. The minimal hop count from $A3$ to $A2$ is 6. Then the average hop distance (AHD) of anchor $A3$ is $AHD_3 = (15+18)/(5+6) = 3$.

$A3$ is the anchor nearest to node 1. Hence, the AHD of $A3$ is used to estimate the distance between node 1 and all the anchors.

$$\begin{aligned} d_{1,A1} &= 3 \times AHD_3 = 9 \\ d_{1,A2} &= 4 \times AHD_3 = 12 \\ d_{1,A3} &= 2 \times AHD_3 = 6 \end{aligned}$$

Finally, triangulation will be used to locate node 1.

3.5. Convex Constraints in Localization[6]

The connectivity between two nodes relates to whether the distance between these two nodes is less than a certain communication range. The convex position estimation (CPE) uses this information in convex optimization and narrows the possible area by the solutions from optimization.

Many researchers have formulated the connectivity-based localization problem as an optimization problem with some convex constraints. When two nodes are connected, the distance between them must be within a range distance. All the connections are then expressed by semi-definite inequalities. As all the constraints are convex, this method is called semi-definite programming or convex programming (SDP)[6].

For examples, the convex constraints that represent the connectivity among nodes are

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R \quad \text{for node } i (x_i, y_i) \text{ in connection with node } j (x_j, y_j).$$

Here, R denotes the communication range defined for a pair of nodes.

4. Problem formulation

A formal definition of the connectivity-based localization problem is given next. Let $G=(V,E)$ be a given network, where V denotes the nodes of the network and E denotes the edge of the network. Let V be partitioned into two sets:

$V_a=\{1, \dots, m\}$ of anchors;

$V_b=\{m+1, \dots, m+n\}$ of sensors (unknown nodes).

E is also partitioned into two sets:

$E_{ab} = \{(i, j) \in E : i \in V_a, j \in V_b\}$ which are the edges between a sensor and an anchor;

$E_{bb} = \{(i, j) \in E : i, j \in V_b\}$ which are the edges between two sensors.

For each anchor $i \in V_a$, the position $a_i \in \mathfrak{R}^2$ is assumed to be known. For each sensor $i \in V_b$, the position $b_i \in \mathfrak{R}^2$ is assumed to be unknown.

Let $C_{ab} = \{(i, j, k_{ij}) : i \in V_a, j \in V_b, k_{ij} \in \{0, 1\}\}$ be the connectivity information between a sensor and an anchor.

Also let $C_{bb} = \{(i, j, k_{ij}) : i, j \in V_b, k_{ij} \in \{0, 1\}\}$ be the connectivity information between two sensors.

The value k_{ij} in C_{ab} or C_{bb} is binary (either 0 or 1):

$k_{ij}=0$ if there is no connection between node i and j .

$k_{ij}=1$ if there is connection between node i and j .

Let a be a vector containing the positions of the anchors

$$a = (a_i)_{i \in V_a} \in \mathfrak{R}^{2m}$$

The goal of the network localization problem is to determine the coordinates of all the sensors (unknown nodes)

$$b = (b_i)_{i \in V_b} \in \mathfrak{R}^{2n}$$

such that b satisfies the following constraints:

Let R be the maximum distance (called the range) within which connectivity can be established.

$$\text{If } k_{ij}=1 \begin{cases} \|a_i - b_j\|_2^2 \leq R^2 \text{ for } (i, j) \in E_{ab} \\ \|b_i - b_j\|_2^2 \leq R^2 \text{ for } (i, j) \in E_{bb} \end{cases}$$

$$\text{else } k_{ij}=0 \begin{cases} \|a_i - b_j\|_2^2 > R^2 \text{ for } (i, j) \in E_{ab} \\ \|b_i - b_j\|_2^2 > R^2 \text{ for } (i, j) \in E_{bb} \end{cases}$$

From the inequalities based on whether any two sensors (including one sensor and one anchor) are in connection or not, the constraints behind those inequalities can be classified into convex constraints and non-convex constraints as shown in Figure 3 and Figure 4.

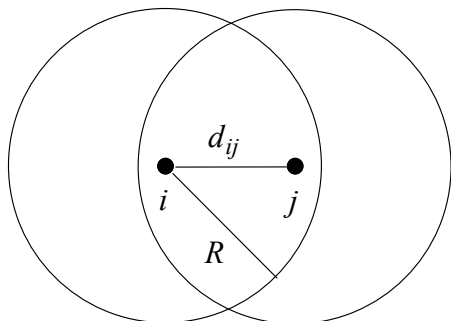


Figure 3. A convex constraint is established when $k_{ij}=1$, d_{ij} is the distance between node i and node j

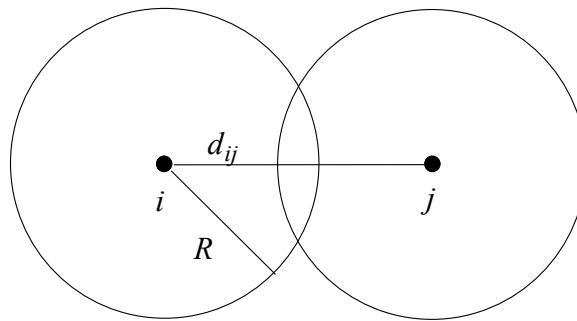


Figure 4. A non-convex constraint is established when $k_{ij}=0$, d_{ij} is the distance between node i and node j

5. The two-objective optimization

5.1 Nature of the problem

From the last section, whether the methods use hop count such as MDS and DV-hop, or the methods utilize neighbors' information such as centroid and APIT, they only use the information when the two nodes are in connection, which is the convex constraint information. A more obvious example is the SDP method which only formulates the optimization problem based on the convex constraints. This method only considers the convex information which then leads to the estimated positions of the unknown nodes. The drawback of these methods is that the estimated nodes tend to crowd together and cannot 'keep distance' from each other.

One example can be shown in Figure 5. In the upper left sub-figure, an example of a sensor network is given. The anchors are node 1-4 (green squares) and the unknown nodes are node 5-7 (small circles). Let the communication range among the nodes be 5. Note that here are 5 connections among the seven nodes, which are displayed as green lines. Based on the 5 connection constraints and the locations of the anchors, a solution is obtained by SDP which is shown by triangles 5'-7' in the upper right sub-figure. The green lines are the expected correct connections between the estimated nodes. However, there are FOUR more connections among the nodes, which are shown by the red lines. These connections are not supposed to exist among the nodes, but the solution obtained by SDP only aims to find a node configuration that satisfy the given five convex constraints, and pay no attention to other consequences.

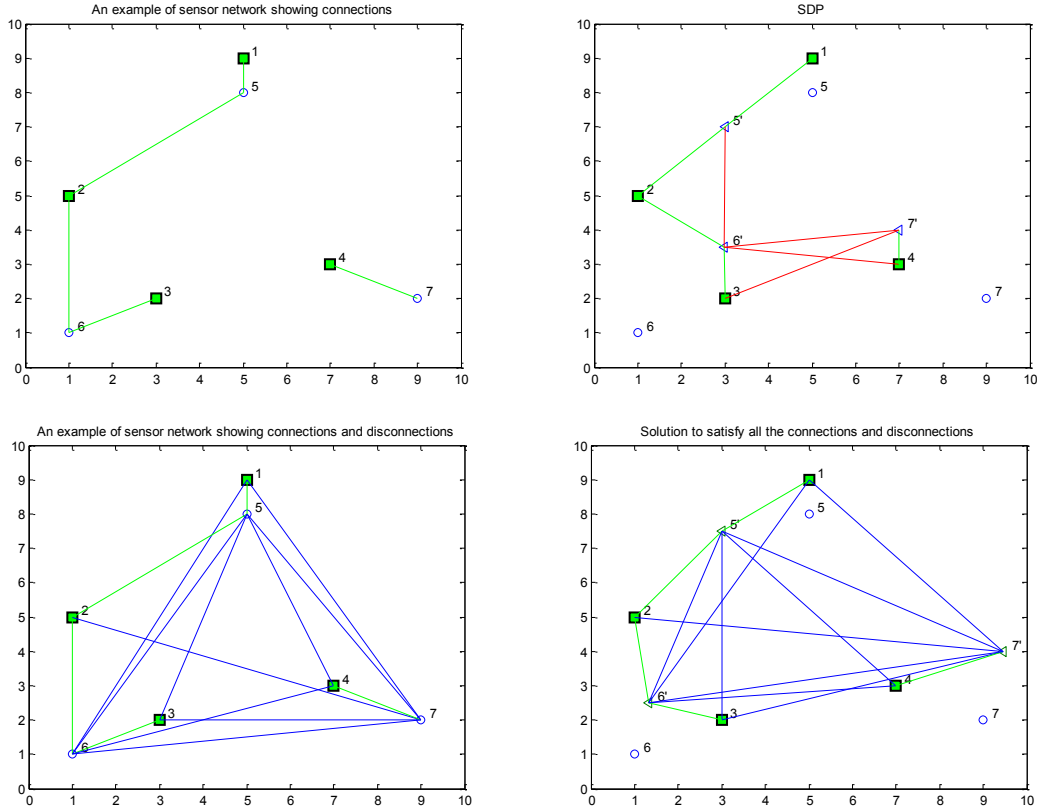


Figure 5. An example showing both the connection and disconnection constraints

The accuracy of a solution to the above example can be improved if the ‘disconnection’ constraints are also taken into consideration. The original localization example is also shown in the lower left sub-figure of Figure 5. In this figure, TEN disconnections among the nodes are shown as the blue lines. In order words, a valid node configuration should satisfy the 5 connections constraints as well as the 10 disconnections constraints. Such a solution is obtained by our developed method and it is shown by the small triangles 5’-7’ in the lower right sub-figure.

Our solution is obtained based on a multi-objective evolution algorithm described in the next section. The result obtained is close to the original node locations for setting up the problem (in the upper left sub-figure). In the search for a solution, the addition of the disconnection information will mean that the connectivity-based localization problem is harder to solve. Once adding these non-convex constraints, the localization problem will become a non-convex optimization problem. Sometimes, non-convex optimization can be reduced into a number of convex minimization sub-problems. However, classical algorithms for this reduction only work in some special cases. In this paper, we have developed an evolutionary algorithm which has the potential to deal with general non-convex problems and to avoid local minima in a stochastic, noncontrollable search [16, 17]. Furthermore, evolutionary algorithms are easy to use in distributed or parallel computing environment, which is appropriate for distributed sensor network. The rest of this section will introduce an evolutionary algorithm for tackling the connectivity-based localization problem.

5.2 Variables

The decision vector (or the chromosome) encodes the locations of all unknown nodes. For 2-Dimension sensor network situation with n unknown nodes and m anchors, the decision vector (named as T) is a $2n$ dimensional row vector, e.g. the first two elements in each decision vector are the coordinates of the 1st unknown sensor (node: $m+1$); the i th pair of elements represent the coordinates of the i th unknown sensor (node $m+i$) as shown in Figure 6. The area boundary of the sensor network bounds the positions of the sensors. It is assumed that each sensor's position has the same upper and lower limit which is the boundary of the sensor network area.

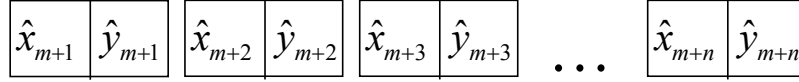


Figure 6. The decision vector T

5.3 Objectives

In this optimization, the solution should satisfy both the convex and non-convex constraints. The situation is formulated as an optimization problem as follows:

- If the connectivity information between a pair of node is known ($k_{ij}=1$), which means the two nodes are in connection, then there is a convex constraint required in the solution to the problem. That is,

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R \quad \text{for node } i \text{ in connection with node } j.$$

- If a pair of nodes is disconnected ($k_{ij}=0$), then there is a non-convex constraint required in the solution to the problem.

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > R \quad \text{for node } i \text{ not in connection with node } j.$$

The feasible solution should satisfy all the above constraints. Therefore, the first objective is for all the estimated node positions to satisfy all the convex and non-convex constraints. The first objective function is called *Count_wrong_connection* with a target value of 0. If a set of variables with the estimated node positions has the function value equals 0, the set of variables is our answer. Pseudocode of the first objective function *Count_wrong_connection* is shown in Figure 7.

```

Count_wrong_connection=0;
for all pair of (i,j) in Eab and Ebb
    if (  $k_{ij}=1$  and  $(R - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) < 0$  ) or (  $k_{ij}=0$  and  $(R - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) >= 0$  )
        Count_wrong_connection++;
    end if
end for
return Count_wrong_connection;

```

Figure.7 Pseudocode for the first objective function *Count_wrong_connection*(T)

The second objective is a summation of all the differences in distances for the node pairs that have violated the connectivity constraints. It is used to help with the convergence of the first objective. Obviously, if a variable vector has function value of the first objective to be zero, the second objective of this variable must also be zero. Figure 8 gives the pseudocode to compute the second objective: *Error*.

```

Error=0;
for all pair of (i,j) in Eab and Ebb
    if (  $k_{ij}=1$  and  $(R-\sqrt{(x_i-x_j)^2+(y_i-y_j)^2})<0$  ) or (  $k_{ij}=0$  and  $(R-\sqrt{(x_i-x_j)^2+(y_i-y_j)^2})\geq 0$  )
        Error= Error +abs( $R-\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$  );
    end if
end for
return Error;

```

Figure 8. Pseudocode for the second objective function $Error(T)$

5.4 PAES

The problem has been formulated as a two-objective optimization problem, which belongs to Multi-objective optimization. The two objectives are consistent in most cases. But sometimes they are conflicted and it is not easy to identify a single solution that simultaneously optimizes each objective. An example is shown in Figure 9. For node i and node j , a node with connection of node i , and non-connection of node j need to be estimated. The solution region is the shadowed area. If point p is the estimation, the first objective $Count_wrong_connection(p)$ is 2, and the second objective $Error(p)$ is $(a-R)+(R-b)$. If point p' is a new estimation, the first objective $Count_wrong_connection(p')$ is 1, and the second objective $Error(p')$ is c . Obviously, p' has smaller $Count_wrong_connection$, but $Error(p')$ is larger than $Error(p)$. In that case, local minima will appear and the algorithm will spend more time to jump out of it.

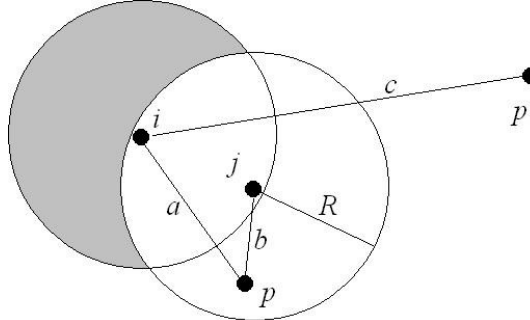


Figure 9. The situation when the two objectives are conflicted

There are already some evolutionary algorithms developed to solve multi-objective optimization. They are called Multi-Objective Evolutionary Algorithms (MOEAs) which include the Non-dominated Sorting Genetic Algorithm (NSGA) and its evolution (NSGA-II), Strength Pareto Evolutionary Approach (SPEA) and its evolution (SPEAS2), and the Pareto Archived Evolution Strategy (PAES) [18]. Any of above algorithms can show us the how accurate and effective MOEAs are in the application of connectivity-based localization problem. In this paper, the performance of PAES on this problem is examined since PAES is simple and always set as benchmarks of other MOEAs, as discussed in [11].

The two-objective optimization evolutionary algorithm used in this framework is the Pareto Archived Evolution Strategy (PAES) algorithm. The implementation of PAES used in our optimization can be found from jMetal [19]. PAES may represent the simplest possible nontrivial algorithm capable of generating diverse solutions in the Pareto optimal set [20]. The simplest form: (1+1) evolution strategy, which is applied in this paper, employs local search but using a reference archive of previously found solutions in order to identify the approximate dominance

ranking of the current and candidate solution vectors. PAES comprises three parts: the candidate solution generator, the candidate solution acceptance function, and the non-dominated-solution (NDS) archive [20]. The candidate solution generator is similar to simple random mutation hillclimbing, but prefers the less crowding solutions to keep the diversity preservation; it maintains a single current solution and, at each iteration, produces a single new candidate via random mutation. The (1+1)-PAES algorithm is outlined in Figure 11. in which the the crowding area is judged by an algorithm called adaptive grid algorithm [20].

Following the well known Pareto dominance, an objective vector $[Count_wrong_connection(T1); Error(T1)]$, is said to dominate another objective vector $[Count_wrong_connection(T2); Error(T2)]$, if the two elements of the first objective vector is smaller than or equal to the corresponding elements of the second objective vector, except the case that the two objective vectors are identical [21]. Symbolically, $T1$ dominates $T2$ is denoted by ' $T1 \succ T2$ '.

Accordingly, $T1$ is a better solution than $T2$ in the above inequality. As discussed in section 5.1, the smallest value of the objective vector is $[0;0]$. Therefore, if all the local minima can be successfully avoided, there should be only one optimal objective vector: $[0;0]$, but there may exist several different solutions to reach this objective vector. To illustrate the mapping, Figure 10 shows the mapping process in which T is assumed to have only two components: $T = [T_1, T_2]$. Note that the difference between Figure 10 and the actual localization problem is that T has $2n$ components in the actual problem.

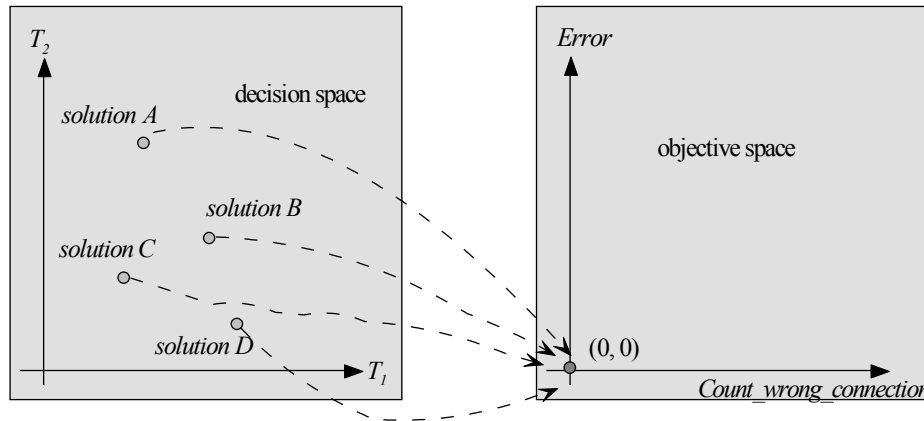
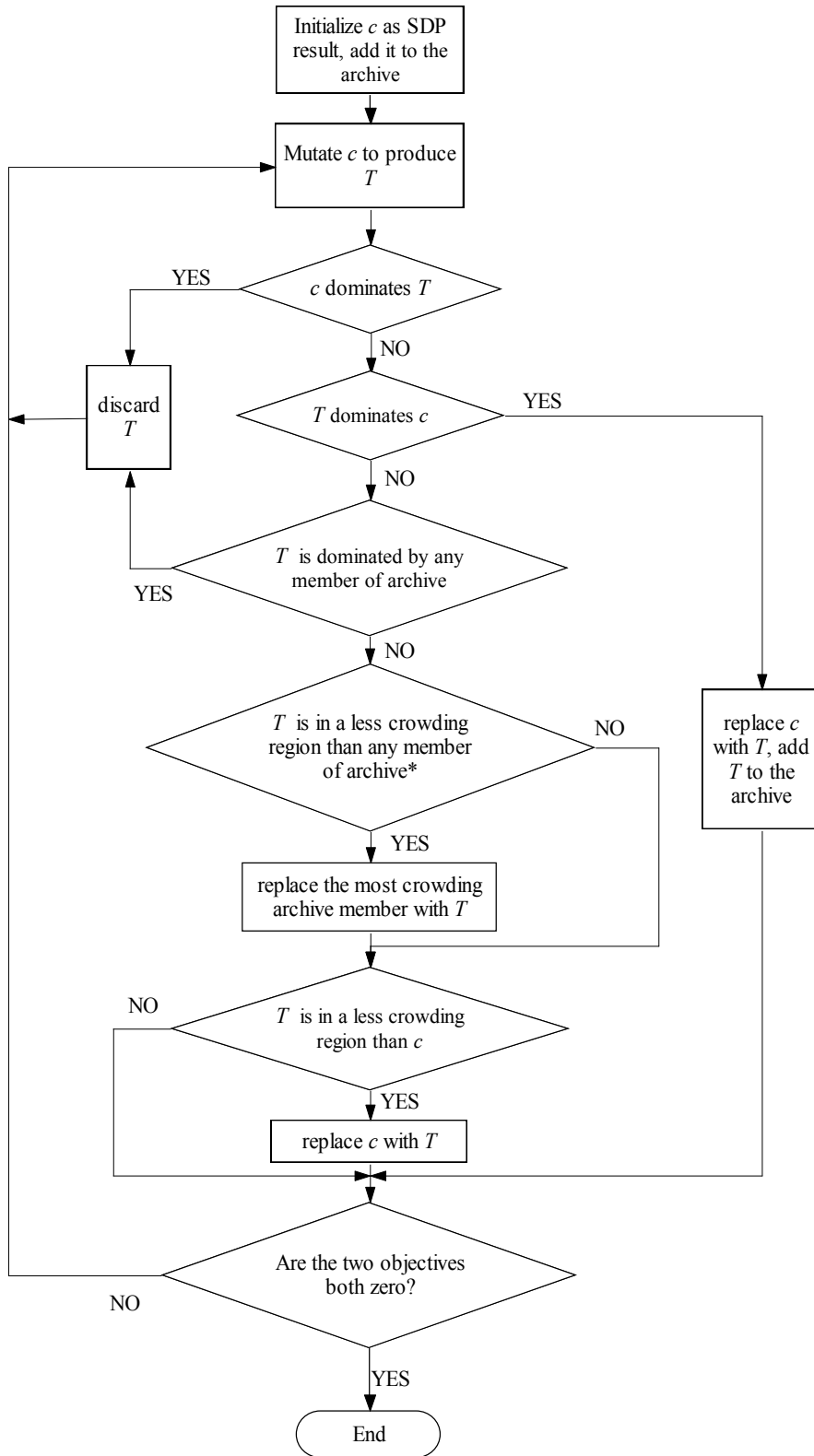


Figure 10. Possible solutions (A, B, C, D)

5.5 The initial decision vector (variables)

The initial decision vector (also called initial solution) may be obtained based on the result of convex optimization, which aims to satisfy all the convex constraints. SDP is one method that can be used to obtain a convex optimization result. Actually, other methods such as DV-hop and MDS may also give an estimation of the initial coordinates. However, the estimation by MDS or DV-hop usually cannot give a solution that satisfy all the convex constraints, whereas SDP can provide a solution that satisfy all the convex constraints by the nature of its algorithm. Hence, the estimation by the SDP method is used as the initial estimates.

The evolutionary algorithm based on PAES can be illustrated by Figure 11.



* if the archive is not full, this procedure is skipped, and add T to the archive

Figure 11. Flowchart for the optimization based on PAES

6. Evolutionary algorithm for connectivity-based sensor network localization

The performance of the proposed two-objective evolutionary algorithm on connectivity-based sensor network localization is evaluated in this section. The scenario considered in our simulation is described below. As shown in Table 1, there are 100 nodes randomly placed in an area of $[0,10]$ by $[0,10]$, among which, 20% are anchors, while the rest are with unknown positions. The range of the nodes is 1.5, which means two nodes within a distance of 1.5 can communicate with each other. The connectivity information between any pair of nodes is assumed to be known. For more convincing result, 10 different network topologies with different node locations are built, and each network topology is used by our algorithm for 30 trials.

Table 1. Parameters of the sensor network

Parameter	Value
m	20
n	80
R	1.5
Boundary	$[0,10]*[0,10]$
Number of topology	10
Trial number	30

Among the 10 topologies, which is called No.1, No.2, ..., No.10, No.1 is chosen as an example and it is shown in Figure 12. 100 nodes are randomly placed in a square area of 10 by 10. The 20 anchors are marked as the small squares, while the other nodes are marked as small circles. The connections between any two nodes are represented by a straight line. Obviously, the length of any lines should be less than or equal to R , which is 1.5. Any node can communicate with other nodes directly or indirectly, in other words, there must be some path between any two nodes in the network. Otherwise, the unconnected nodes will be not recognized by the network.

For each network topology (No.1 to No.10), the unknown nodes can be divided into two groups. Some unknown nodes are in direct connection with at least one anchor, and they are collected into the first group. The rest of the unknown nodes are grouped into the second one. The nodes in the first group of the No.1 network is drawn with lines connected to corresponding anchors as Figure 13. The percentage of the number of the nodes in the first group over the total number of unknown nodes is shown as Figure 14. The percentages of the 10 different topologies also called the density of the anchors which vary from 63% to 85%.

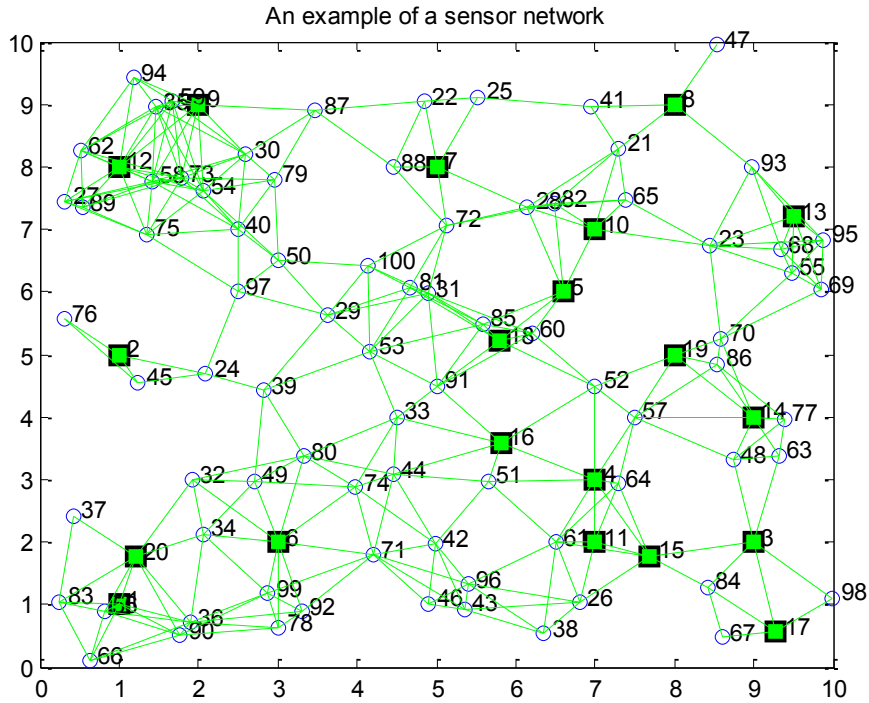


Figure 12. The topology of the No.1 network. The anchors are marked as small squares; the real positions of the unknown nodes are marked as small circles; and the connections between any two nodes are shown as the green lines.

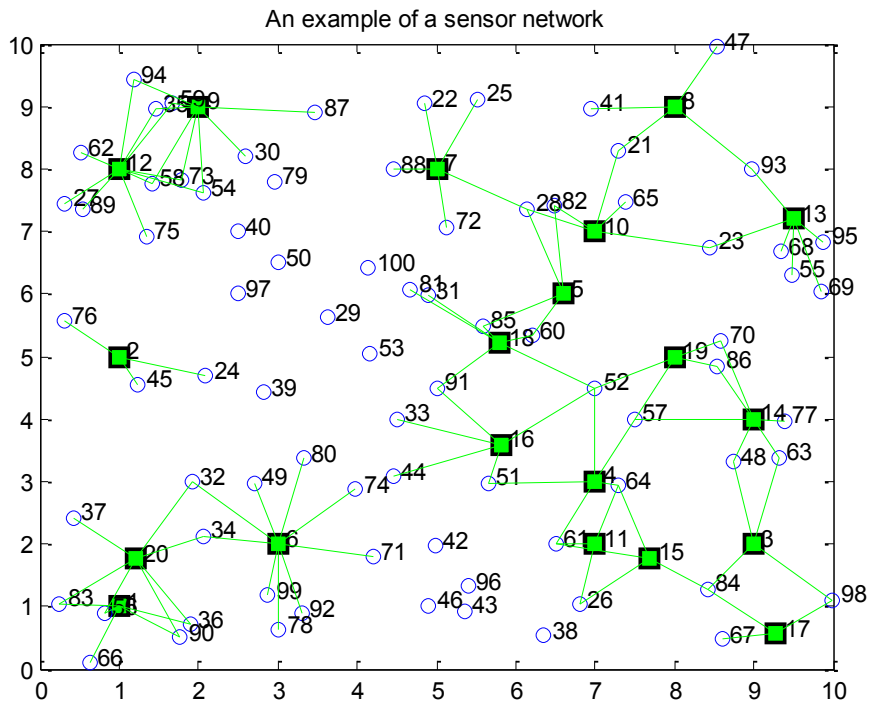


Figure 13. The connections between the nodes in the first group and the anchors are represented by the green lines, while the anchors are marked as the small squares; the unknown nodes are marked as small circles.

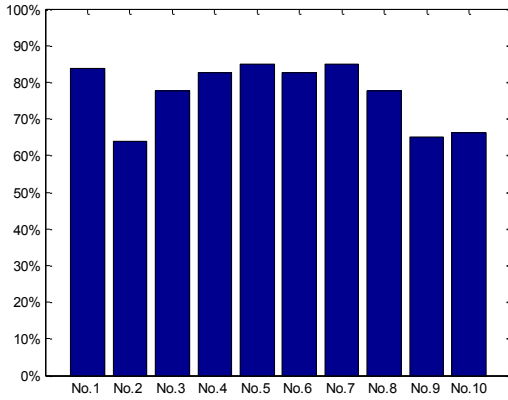


Figure 14. The percentage of the unknown nodes in the first group over all unknown nodes

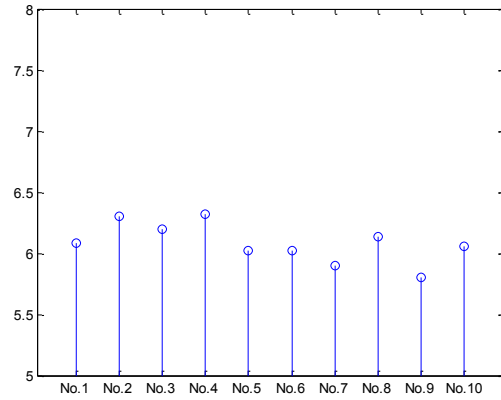


Figure 15. The average connectivity degree of each topology

After showing the density of the anchors, the density of the nodes will be discussed below. Some researchers define the ‘degree’ of nodes to represent the density of the nodes. The degree may be understood as the average number of connections for a node in the network, which is defined as below. In our simulation, the degree of the 10 networks varies from 5.8 to 6.4 in Figure 15.

$$Degree = (2 * \text{number of connections}) / \text{number of nodes}.$$

The parameters used in PAES are shown in Table 2. In this simulation, the archive size does not affect the result greatly. Its value can vary from 20 to 100. There are two independent conditions that the iteration would stop. The first is when the two targets: *Count_wrong_connection*, *Error* both become zero. The second is the number of iteration equals the maximum number of iteration defined.

Table 2. Parameters used in PAES

Parameter	value
Archive size	100
Number of regions	5
Maximum number of fitness evaluations	1,000,000

It is repetitive to show the detail information of all the 10 topologies. Instead, only the information for No.1 is given in Figure 16 and Figure 17. Figure 16 shows the value of the two objectives when the iteration in No.1 network finishes. There are 30 trials of the simulation. Therefore, there should be 30 points in Figure 16, each point is with x-axis representing value of *count_wrong_connection* and y-axis representing value of *Error*. However, in the figure, there are only six points because some points are overlapped on same points. To show the number of overlapped trials, Figure 17 gives how many trials in different value of *count_wrong_connection*. There are 23 trials ended when *Count_wrong_connection*=0, and there is only one point in Figure 16 when *Count_wrong_connection*=0. That means 23 trials are overlapped as one point in Figure 16 at the position [0,0].

The accuracy of the estimation is evaluated by the difference between the estimated positions of unknown nodes and the positions of the corresponding unknown nodes when the problem is setup. The average error per unknown node is calculated by the formula below.

$$error\ per\ node = \frac{\sum_{i=m+1}^{m+n} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{n}$$

where (\hat{x}_i, \hat{y}_i) is the estimated position of node i , (x_i, y_i) is the real position of the node i .

Boxplots of the average error for unknown nodes for the 10 topologies are shown in Figure 18. On each box, the central mark (red) is the median of the 30 value of error per node for the 30 trials, the edges (both blue) of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points the algorithm considers to be not outliers, and the outliers are plotted individually as red crosses ‘+’.

While the evaluation stops at two independent conditions, the number of iteration required for each trial may be different. Boxplot in Figure 19 shows the 30 numbers of iteration for each topology. No.1, 3, 5, 6, 7, 8 are with most trials being stopped by reaching $[Count_wrong_connection; Error]=[0;0]$. The corresponding topologies in Figure 18 give more accurate estimation than others. Figure 20 and 21 show the boxplots of the two targets: $Count_wrong_connection; Error$ respectively. It is shown that the two targets are not contrary in most cases. They have the similar trend under different topologies.

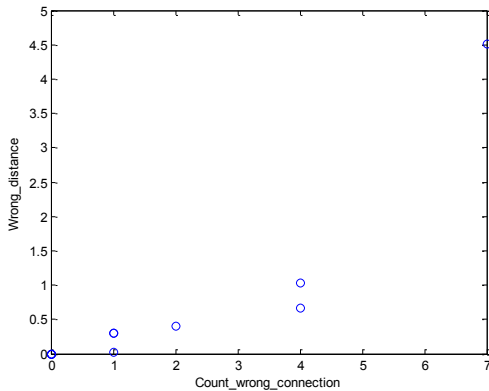


Figure 16. The values of the two objectives when the iteration of the network No.1 is finished

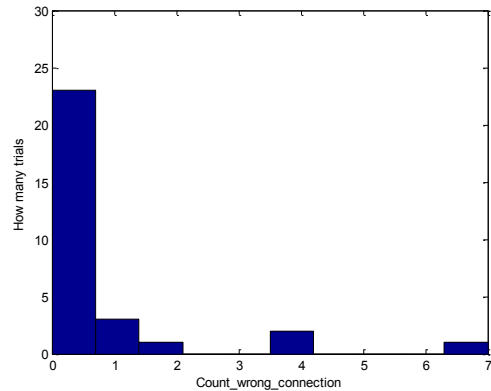


Figure 17. The number of the trials under different values of $Count_wrong_connection$ for the network No.1

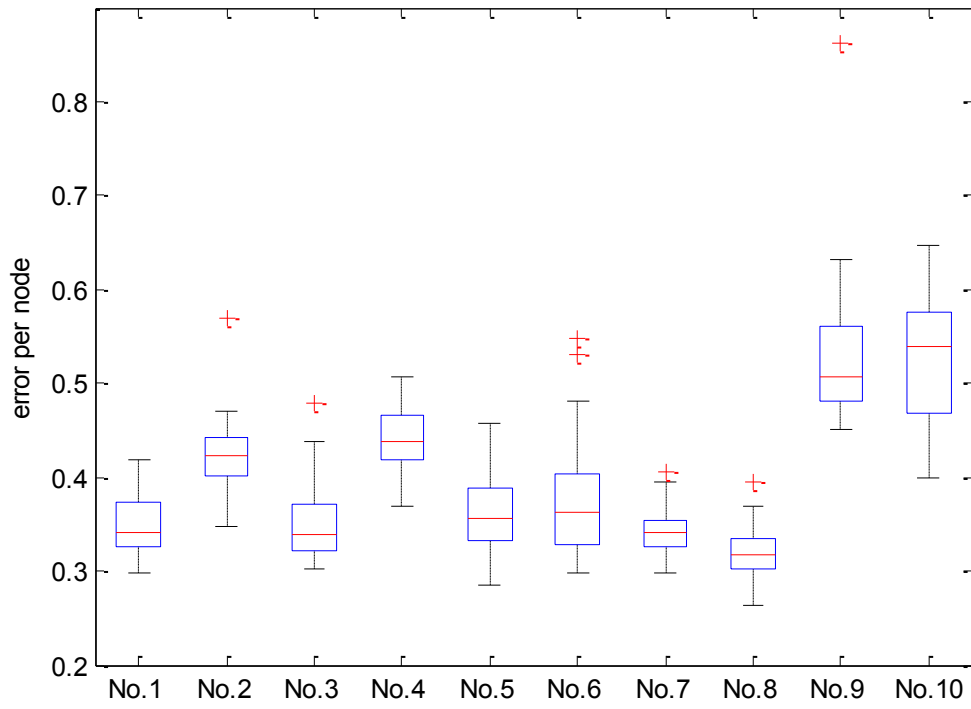


Figure 18. Boxplot of the error per node for the 10 topology networks

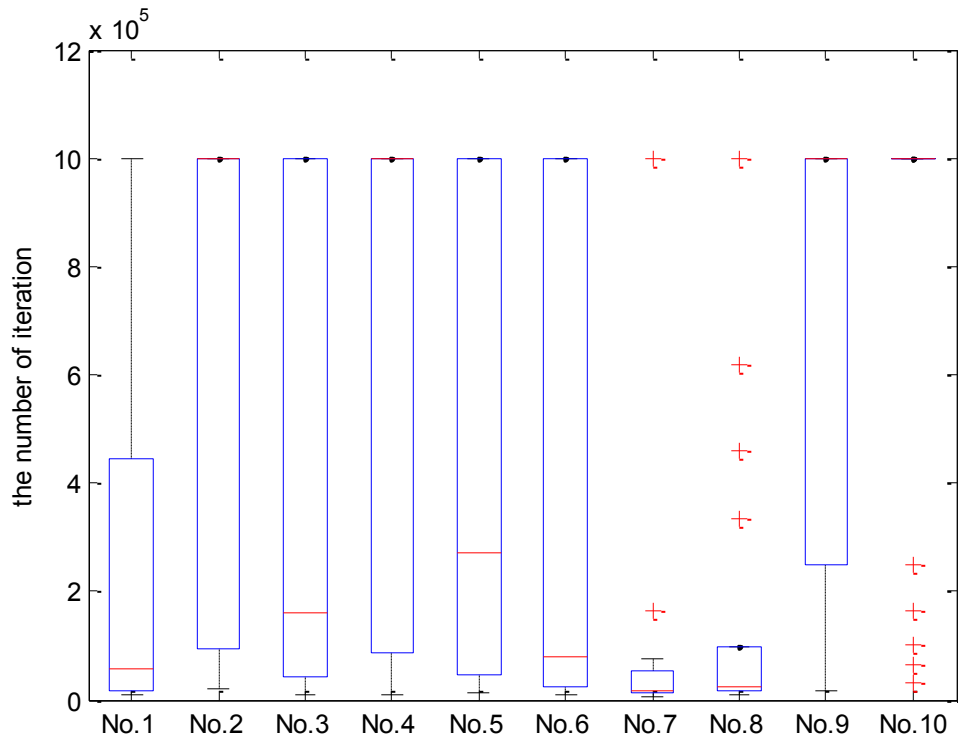


Figure 19. Boxplot of the number of iteration for the 10 topology networks

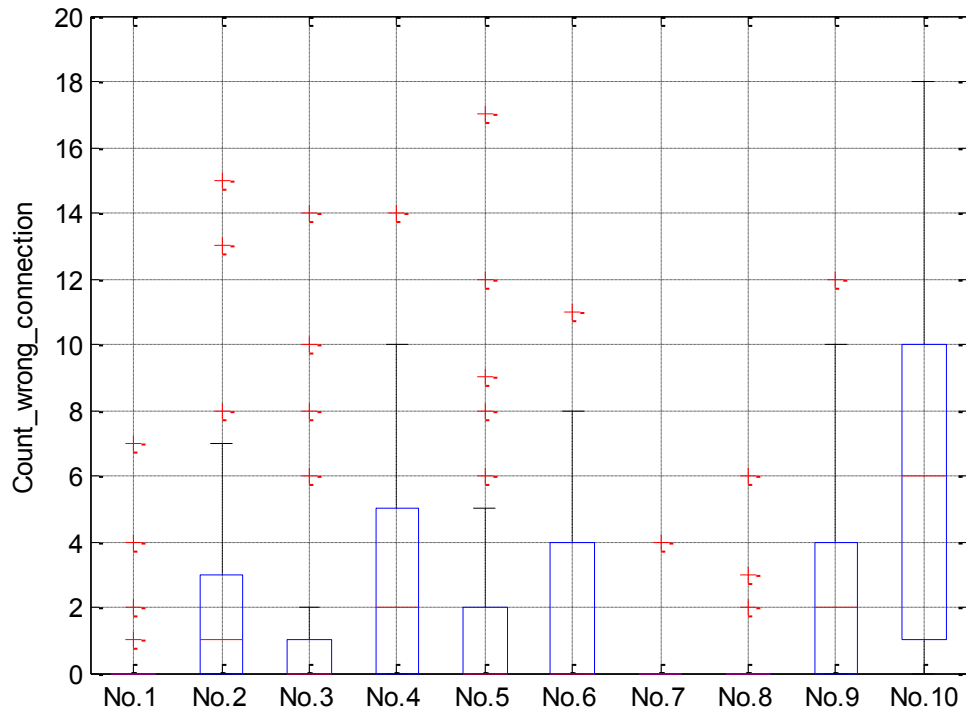


Figure 20. Boxplot of the first target: *Count_wrong_connection* for the 10 topology networks

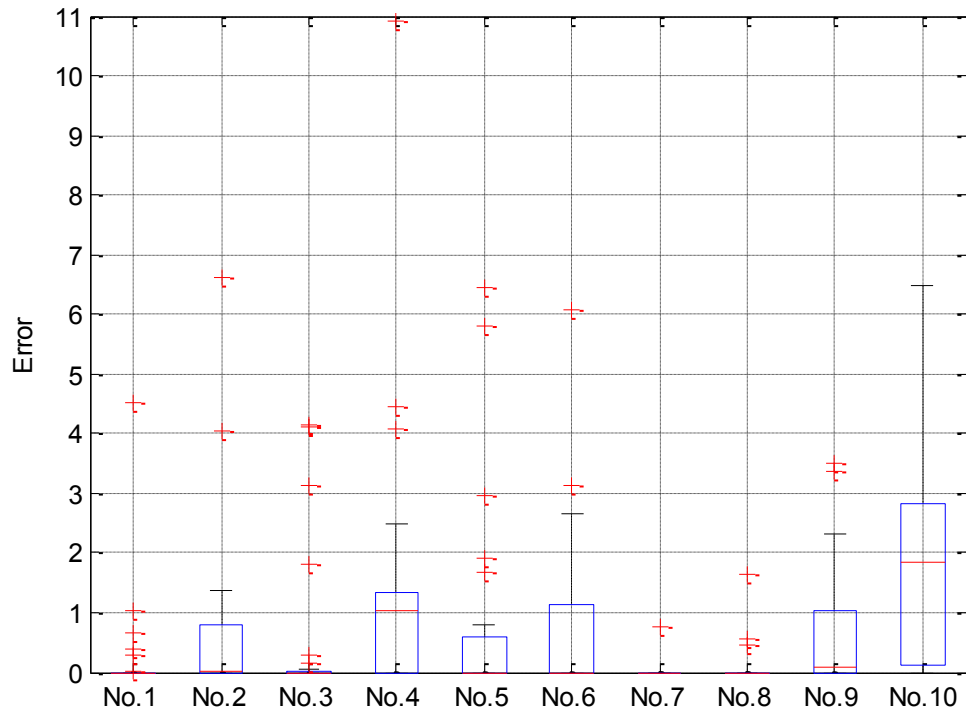


Figure 21. Boxplot of the first target: Error for the 10 topology networks

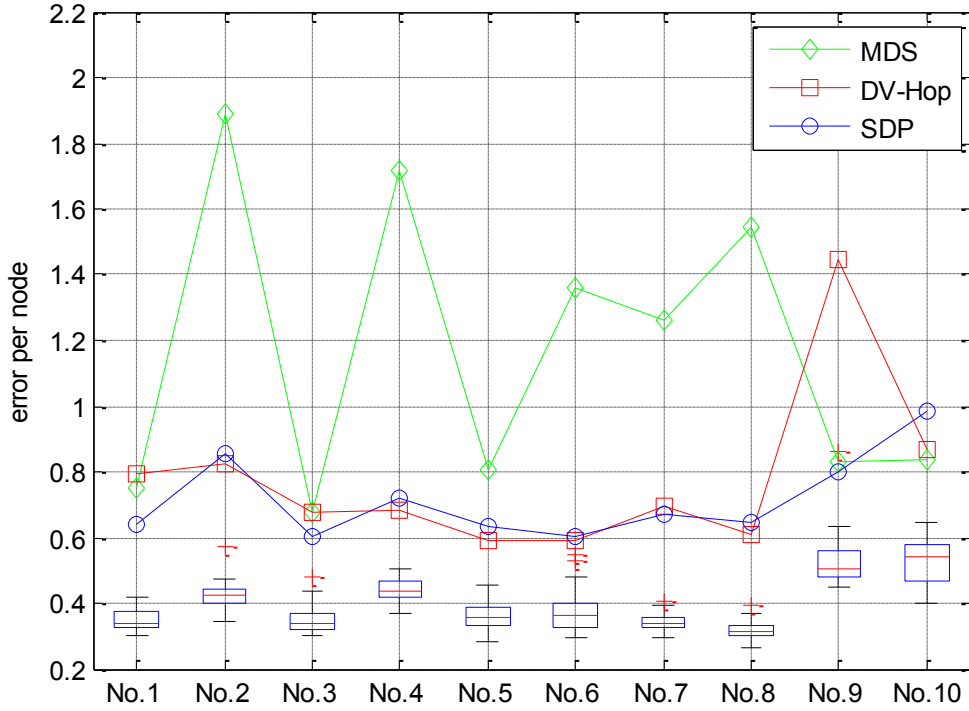


Figure 22. Our result compared with other methods (MDS, DV-Hop and SDP)

Simulation results of our method compared with other methods (MDS, DV-hop and SDP) are shown in Figure 22. It is clearly shown that all the results of PAES is better than convex methods for each topology. Error per node for DV-Hop and SDP are mostly between 0.6 and 0.8. The error for MDS is not very stable, but the minimum is above 0.7. PAES can give much more accurate estimation. The error per node mainly distributed around 0.4. If we concentrate on each single topology, the median of error in PAES is nearly half of error in the most accurate convex method. The whisker range is around 0.1, which is a relative small disturbance. Even the outliers are underneath the convex methods, while only one exception appears in the No.9, in which one outlier is bigger than error per node of DV-Hop and MDS. In the No.9 No.10, the error of PAES is a little bigger, but the most of the 30 trials are 0.2 less than the error of convex methods.

7. Conclusion

The constraints in a connectivity-based sensor network localization problem can be divided into two categories: convex and non-convex, which indicate that the problem is actually a non-convex optimization problem. However, current connectivity-based localization methods only utilize the convex constraints. Motivated by searching for good accurate solution based on the connectivity information, a two-objective evolutionary algorithm framework has been developed. The first objective is to count the wrong connectivity including the violated connections and violated disconnections between the node pairs. The second objective is to quantitatively evaluate the degree of seriousness in the violation of the connectivity. A popular MOEA method, PAES, is used to tackle this optimization problem. To obtain an appropriate initial decision vector for this

optimization, convex optimization result from SDP is used. The estimation of SDP satisfies all the convex constraints and makes the number of the wrong connectivity to be small enough to converge quickly. In a comparison of its accuracy with other methods such as MDS, DV-Hop, and SDP, the developed method can decrease the estimated position error by nearly 50%.

References

- [1] T. Eren, O. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *IEEE INFOCOM 2004*, pp. 2673-2684.
- [2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE*, vol. 7, pp. 28-34, 2000.
- [3] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking* San Diego, CA, USA: ACM, 2003.
- [4] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking* Annapolis, Maryland, USA: ACM, 2003.
- [5] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Kluwer journal of Telecommunication Systems*, pp. 267–280, 2003.
- [6] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, pp. 1655-1663 vol.3.
- [7] M. Terwilliger, A. Gupta, A. Khokhar, and G. Greenwood, "Localization using evolution strategies in sensornets," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, pp. 322-327 Vol.1.
- [8] Q. G. Zhang, J. H. Wang, C. Jin, J. M. Ye, C. L. Ma, and W. Zhang, "Genetic Algorithm Based Wireless Sensor Network Localization," in *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, 2008, pp. 608-613.
- [9] V. Tam, K. Y. Cheng, and K. S. Lui, "Using Micro-Genetic Algorithms to Improve Localization in Wireless Sensor Networks," *Journal of Communications, North America*, vol. 1, 2006.
- [10] V. Tam, K. Y. Cheng, and K. S. Lui, "A Descend-Based Evolutionary Approach to Enhance Position Estimation in Wireless Sensor Networks," in *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, 2006, pp. 568-574.
- [11] M. Vecchio, R. Lopez-Valcarce, and F. Marcelloni, "A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks," *Applied Soft Computing*, vol. In Press, Corrected Proof, 2011.
- [12] "http://en.wikipedia.org/wiki/Multidimensional_scaling."

- [13] S. Yi and W. Ruml, "Improved MDS-based localization," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 2640-2651 vol.4.
- [14] S. Minhan, C. Wook, and C. Hyunseung, "A Cluster-Based MDS Scheme for Range-Free Localization in Wireless Sensor Networks," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference on*, pp. 42-47.
- [15] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, 2001, pp. 2926-2931 vol.5.
- [16] E. S. Mistakidis and G. E. Stavroulakis, *Nonconvex Optimization in Mechanics*: Kluwer Academic Publishers, 1998.
- [17] R. H. Byrd and R. A. Waltz, "An active-set algorithm for nonlinear programming using linear programming and equality constrained subproblems," Optimization Technology Center, Northwestern University 2002.
- [18] "http://en.wikipedia.org/wiki/Multiobjective_optimization," 2011.
- [19] "<http://jmetal.sourceforge.net/>," 2011.
- [20] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evol. Comput.*, vol. 8, pp. 149-172, 2000.
- [21] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," in *Metaheuristics for Multiobjective Optimisation*: Springer-Verlag, 2003, pp. 3--38.