Geo-Location with Wireless Sensor Networks using Non-linear Optimization

Abdellah Chehri[†, ††], Paul Fortier[†], and Pierre-Martin Tardif[†]

[†]Department of Electrical Engineering, Laval University, Québec, Qc, G1K 7P4, Canada.
††LRCS Laboratory, 450, 3 Av, Val-d'Or, J9P 1S2, Qc, Canada

Summary
In this paper we propose a Reduced-Complexity Differential Evolution (DE) for localization of multi-hop sensor networks. In relatively smaller population size of DE algorithm (Np<50). We study scenarios where the nodes do not receive signal from anchors directly (i.e., multi-hop). The nodes with unknown positions use a new algorithm based on non-linear least squares (NLLS) method for deriving their estimated location. We consider the problem of minimizing a cost function based on distance measurements between these nodes and some definite anchors. We use Reduced Complexity Differential Evolution (RCDE) to find the global minimum of this cost function which corresponds to the nodes estimated coordinates. The main characteristic of the DE algorithm is that it can get close to the optimal solution with low complexity, if the steps of the algorithm are designed appropriately. In this paper we show that the DE optimization method can be used in ad hoc sensor networks to estimate the location of nodes accurately. The performance in terms of mean localization squared error is evaluated by simulations for different scenarios. The sensitivity of the estimated position to the uncertainty on distance measurements, radio transmission, and irregular area are also evaluated.

Key words: *Wireless Sensor Networks, Localization, Non-Linear Least Squares, Optimization, Differential Evolution (DE).*

1. Introduction

Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and our lives. Many novel applications are emerging: environmental monitoring, precision agriculture, health monitoring, smart building failure detection, target tracking and personal localization, etc. In these applications, it is necessary to accurately orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful [1]. Since gathering physical measurements without attaching coordinates to the collected data might be of little use, "sensing data without knowing the sensor location is meaningless" [2]. Further, knowledge of sensor location can be used to facilitate network functions such as packet routing [3], and collaborative signal processing [4].

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

This paper presents a new technique for sensor localization. We propose to use Differential Evolution algorithm technique to optimize a cost function in order to estimate the location of the sensor nodes. Initially all nodes, except the anchors, are assigned a random estimate of the location. A cost function, which represents the quantitative measure of the "goodness" of the coordinate estimate, was formulated using the estimated coordinates of the nodes and their measured distances with respect to anchors. Differential Evolution is used to optimize this cost function which corresponds to the estimation of the node's coordinate.

Differential Evolution (DE) is one of the recent population-based techniques. DE was invented by K. Price and R. Storn in 1995 as an heuristic method for minimizing nonlinear and non differentiable continuous space functions. DE algorithm is considered as a stochastic optimization method minimizing an objective function that can model the problem's objectives while incorporating constraints. The algorithm mainly has three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using a few control parameters.

The proposed algorithm for sensor localization is implemented in a centralized architecture, where all nodes send their measurements to a central station for localization.

The rest of the paper is organized as follows: Section 2 presents prior work in positioning algorithm for sensor networks. Section 3 discusses the design goals of the positioning protocol developed, we described in detail the Positioning algorithms associated to optimality criteria. Differential Evolution for Non-Linear Optimization and the used parameter for DE algorithms are given in Section 4. Section 5 presents simulation; we performed and analyze the results for different scenarios, while Section 6 concludes the paper.

2. Related Work

Many positioning systems for sensor networks have been proposed in the literature [5] - [8]. In this section, we briefly present the most important ones and focus more specially on the solutions that use centralized architectures that use hop-counts as distances, as they are the most closely related to our proposal. All the solutions can be broadly classified into two categories: connectivity based approaches and measurement based approaches.

2.1 Connectivity based multi-hop localization approach's

Connectivity based (called also "range free") localization algorithms use the connectivity information [9]. The principle of these algorithms is: a sensor being in the transmission range of another sensor defines a proximity constraint between both sensors, which can be exploited for localization. The GPS-less low cost localization system described in [10] is an example of a connectivity based system. In this system, a set of pre-deployed, location aware reference nodes transmit spatially overlapped beacon signals. Other nodes with unknown locations can localize themselves at the centroid of the reference nodes from which they can receive beacon signals. The best results are obtained when the nodes are arranged in a mesh pattern.

He et al in [11], proposed a new technique called APIT for large sensor localization. The main idea of APIT is to divide the whole network into triangular regions among anchors, and then to determine the possible position where a sensor locates via the aggregation of the two distinct triangular regions. Consequently, the position of a sensor can be estimated by calculating the center of gravity of the intersections of the triangles where a sensor resides.

A method inspired by a data analysis technique, multidimensional scaling (MDS), is proposed in [12]. The algorithm, called MDS-MAP starts with rough estimations of distance between each pair of sensors by means of the all-pairs shortest-paths algorithm. Classical MDS is then used to derive node locations.

2.2 Distance-based multi-hop localization approaches

The core of distance-based localization algorithms is the use of inter-sensor distance measurements in a sensor network to locate the entire based on the approach of processing the individual inter-sensor distance data, distance-based localization algorithms can be considered in two main classes: centralized algorithms and distributed algorithms. Distributed algorithms rely on self localization of each node in the sensor network, while centralized algorithms use a single central processor to collect all the individual inter-sensor distance data and produce a map of the entire sensor network.

2.2.1 Distributed algorithms

To estimate the node-anchor distance, many researchers have found "hop count" to be a useful way to compute inter-node distances1. In the DV-hop method developed by Niculescu and Nath [13], the anchor propagates their location information inside the network. Each node forwards the anchor information to its neighbours and maintains a table with the anchor ID, location, and hop distance. When an anchor receives one of the propagated packets with the position of a different anchors. It uses that information to calculate the average hop-distance between the two anchors. The computed average hop distance is broadcasted back into the network as a correction to previously know hop distances. The nodes that receive this message use the average hop distances to each of the anchors to estimate their distances to the anchors.

This information is then used to triangulate the node location [5]. The DV-distance approach is similar to DV-hop but uses the measured distances between two adjacents nodes [13].

The Euclidean propagation method uses the true distance measurement to an anchor. In this case, nodes that have at least two distance measurements to nodes that have distance estimates to an anchor can use simple trigonometric relationships to estimate their locations. Another approach described in [14] uses an algorithm similar to DV-Hop called Hop-TERRAIN in combination with a least squares refinement.

2.2.2 Centralized algorithms

Various techniques have been developed for designing centralized localization. Most of these methods are based on minimizing some global error functions (called objective function), which can be different when the model of uncertainty changes. Depending on the kind of optimization problem being formulated, the characteristic and computation complexity varies. For example, the maximum likelihood estimation for sensor network localization problem is a non-convex optimization problem [15]. In fact, most previous approaches adopt global optimization techniques such as non-linear least square methods.

An alternate approach, called the semi-definite programming (SDP) method has been proposed for sensor localization [16]. In [16] the distance-based sensor network localization problem is formulated in a quadratic form and solved using SDP; and in [15] the result in [16]

[1] The transmission range of each node is limited by energy constraint.

is improved using a gradient search procedure to fine-tune the initial solution obtained using SDP.

The stochastic optimization approach suggests an alternative formulation and solution of the distance-based localization problem using combinatorial optimization notions and tools. The main tool used in this approach is the simulated annealing (SA) technique [17], which is a generalization of the well known Monte Carlo method in combinatorial optimization. One particular property of the SA method is its robustness against converging to a false local minimum.

In this paper we show that the reduced-complexity Differential Evolution optimization method can be used in ad hoc sensor networks to estimate the location of nodes accurately.

3 Positioning algorithms associated to optimality criteria

The algorithm presented here creates a virtual coordinate system for sensor networks. We consider a static[2], random network with no isolated nodes[3] (Fig. 2). For each node, we call neighbours the set of nodes within its transmission range. To improve the accuracy of estimation, we use different number of anchor percentage.

In practice a popular method for estimating node location is using the method of non-linear least squares (NLLS). This approach assumes that the node located at (x, y) calculate a relative distance form different anchors by multi-hop. These M anchors are located at $(x_1, y_1), (x_2, y_2), \cdots (x_M, y_M)$ As a performance measure we consider the function:

$$f_i(z) = d_i - \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \,. \quad (1)$$

where d_i distance from i^{th} anchor and $z = (\hat{x}, \hat{y})$. Basically, $f_i(z)$ is an estimation of the error between the measured distance d_i of node to i^{th} anchor, and the estimated distance $\sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2}$, where $(\hat{x}, \hat{y})$ represent the estimated location of the node. Hence to obtain a location estimate the following cost function is used:

$$F(z) = \sum_{i=1}^{M} \alpha_i^2 f_i^2(z) \cdot \quad (2)$$

where is the weight which may reflect the i^{th} measurement distance accuracy. For our analysis we propose that this parameter is inversely proportional to the number of hop (N_{hop}) between node and any anchor. In other word, we penalize the distance measurement where the number of hop is great, the evident explication that the error measurement propagate with N_{hop}.

Other objective function $F(z)$ can be formed replacing, for example, $f_i^2(z)$ with $|f_i(z)|$. However, these methods usually do not perform as well as minimizing the sum of squares [18].

The estimated location of node is obtained by minimizing the cost function $F(z)$ given in (2). To determinate the estimated location many algorithms have been developed using others objectives functions. For example, Cheng et al [19] use Gauss-Newton method for sensor positioning. Each sensor updates its estimated location by computing the Gauss-Newton step for a local cost function and choosing a proper step length. Then it transmits the updated estimate to all the neighbouring sensors. In [20], an iterative non-linear least-squares method is proposed to estimate sensor locations.

The gradient-based algorithms can also be employed for position estimation. One is the Davidon-Fletcher-Powell (DFP) Quasi-Newton algorithm [21] which has been used in the UWB precision assets location system developed by Multispectral Solution, Inc [22].

In this paper we use stochastic research based on Reduced-Complexity Differential Evolution (will be described in the next section) to find global minima of model (2) (Fig. 1). It should be noted that this least-squares approach is statistically more justifiable that the usual ad hoc procedure of finding intersection of multiple circles or ellipse and then using least-squares method on the reduced intersection data (see Fig. 2). This is because the estimated distances using DV-distance are always greater than the true distance; also an additive error caused by inter-nodes error distance measurements is collapsed the estimated distance.

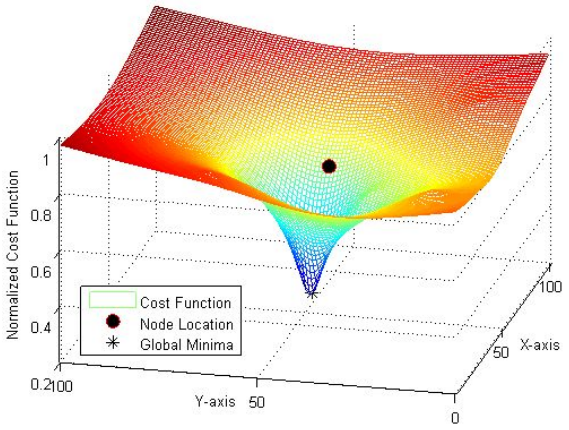

Figure1: Plot of Cost Function F(z) for Radiolocation Illustrating Global Minima (Normalized log(F(z))).

2 The nodes can be mobile; in this case the process of localization must be brought up to date in a regular way.

3 If node is isolated from network it is impossible to locate it.

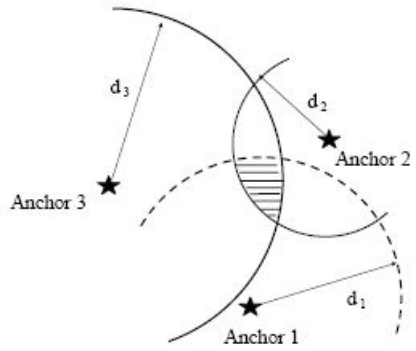

Fig. 2: Ambiguity in localization, as all the circles do not intersect at one point (shaded region).

4 Differential Evolution for Non-Linear Optimization

Differential Evolution (DE) is one of the recent population-based techniques. DE was invented by K. Price and R. Storn in 1995 as an heuristic method for minimizing nonlinear and non differentiable continuous space functions. DE algorithm is considered as a stochastic optimization method minimizing an objective function that can model the problem's objectives while incorporating constraints. The algorithm mainly has three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using a few control parameters.

The DE algorithm is a population based algorithm like genetic algorithms using the similar operators; *crossover*, *mutation* and *selection*. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operation. This main operation is based on the differences of randomly sampled pairs of solutions in the population. Unlike the binary chromosomes typical of genetic algorithms, an individual in DE is generally comprised of a real-valued chromosome.

DE is a simple evolutionary algorithm that creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness. This is a rather greedy selection scheme that often outperforms traditional EAs.

The algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search space. The DE algorithm also uses a non-uniform crossover that can take child vector parameters from one parent more often than it does from others. By using the components of the existing population members to construct trial vectors, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space.

An optimization task consisting of D parameters can be represented by a D-dimensional vector. In DE, a population of *Np* solution vectors is randomly created at the start. This population is successfully improved by applying *mutation*, *crossover* and *selection* operators. The main steps of the DE algorithm are given in algorithm 1.

Algorithm. 1: Outline of DE's main procedure

%Differential Evolution
1: Evaluate the initial population *P* of random individuals.
2: **While** stopping criterion not met, **do**:
  2.1. **For** each individual *Pi* (*i* = 1, . . ., *popSize*) **repeat**:
    (a) Create candidate *C* from parent *Pi*.
    (b) Evaluate the candidate.
    (c) **If** the candidate is better than the parent, the candidate replaces the parent.
      Otherwise, the candidate is discarded.
  2.2. Randomly enumerate the individuals in *P*.

Although several DE algorithms exist we only describe one version of the algorithm based on the DE/rand/1/bin scheme [23]. The different variants of the DE algorithm are described using the shorthand DE/x/y/z, where x specifies how the base vector to be perturbed is chosen (rand if it is randomly selected or best if the best individual is selected), y is the number of difference vectors used, and z denotes the crossover scheme used (bin for crossover based on independent binominal experiments, and exp for exponential crossover). DE/rand/1/bin version is used in our simulation (described in Algorithm. 2).

Algorithm. 2: Outline of the candidate creation in scheme *DE/rand/1/bin*

%Candidate creation
Input: Parent Pi
1: Randomly select three individuals *Pi1, Pi2, Pi3* from *P*,
2: Calculate candidate *C* as $C = Pi1 + F.\ (Pi2 - Pi3)$, where F is a scaling factor.
3: Modify the candidate by binary crossover with the parent using crossover probability CR.
Output: Candidate C

At the start of this algorithm, a population of N, d-dimensional vectors $X_j = (x_{i1}, x_{i2}, ..., x_{id})$, $j = 1, ..., n$, is randomly initialised and evaluated using a fitness function *f*. During the search process, each individual (j) is iteratively refined. The modification process has three steps:

1) Create a variant solution, using randomly selected members of the population.
2) Create a trial solution, by combining the variant solution with *j* (crossover step).
3) Perform a selection process to determine whether the trial solution replaces *j* in the population.

Under the mutation operator, for each vector $X_j(t)$, a variant solution $V_j(t+1)$ is obtained using equation:

$$V_j(t+1) = X_m(t) + F(X_k(t) - X_l(t)) \quad (3)$$

where, $k, l, m \in 1,...,N$ are mutually different, randomly selected indices, and all the indices $\neq j$ ($X_m(t)$ is referred to as the base vector, and $X_k(t) - X_l(t)$ is referred to as a difference vector). Variants on this step include the use of more than three individuals from the population, and/or the inclusion of the highest-fitness point in the population as one of these individuals [21]. The difference between vectors $X_k(t)$ and $X_l(t)$ is multiplied by a scaling parameter F. The scaling factor controls the amplification of the difference between $X_k(t)$ and $X_l(t)$, and is used to avoid stagnation of the search process.

Following the creation of the variant solution, a trial solution $U_j(t+1) = (u_{j1}, u_{j2},..., u_{jd})$ is obtained using:

$$U_{jn}(t+1) = \begin{cases} V_{jn}, \text{if } (rand \le CR) \text{ or } (j = rnbr(i)); \\ X_{jn}, \text{if } (rand > CR) \text{ and } (j \ne rnbr(i)) \end{cases} \quad (4)$$

where *n=1,2,...,d, rand* is drawn from a uniform random number generator in the range (0,1), *CR* is the user-specified crossover constant from the range (0,1), and *rnbr*(*i*) is a randomly chosen index chosen from the range (1,2,…, *n*).

The random index is used to ensure that the trial solution differs by at least one component from $X_i(t)$. The resulting trial solution replaces its predecessor, if it has higher fitness (a form of selection), otherwise the predecessor survives unchanged into the next iteration of the algorithm (equation 5).

$$X_i(t+1) = \begin{cases} U_i(t+1), \text{if } f(U_i(t+1)) < f(X_i(t+1)); \\ X_i(t), \qquad \text{otherwise} \end{cases} \quad (5)$$

Sometimes, the newly created candidate falls out of bounds of the variable space. In such cases, many approaches of constraint handling are possible. We address this problem by simply replacing the candidate value violating the boundary constraints with the closest boundary value. In this way, the candidate becomes feasible by making as few alterations to it as possible. Moreover, this approach does not require the construction of a new candidate.

Generally, the DE algorithm has three parameters, the population size (*Np*), the crossover rate (*CR*), and the scaling factor (*F*). Higher values of *CR* tend to produce faster convergence of the population of solutions. For our simulation *Np=50*, *F=0.8*, and *CR=0.8* were taken.

5 Simulations

In this section we state and solve (via simulations) the location problem for the two-dimensional case. We assume that the sensors are deployed randomly over a two-dimensional monitored area (i.e., on the ground). However, we can be easily extended to three-dimensional space. The N nodes are randomly placed monitored area. However the M anchors are placed at known location. The anchors are necessary prerequisite to locate a network in a global coordinate system. Their placement can often have a significant impact on localization [12].

In most cases the nodes do not receive signal from anchors directly (hence no direct distance estimation). However, many groups have found hop count to be a useful way to compute inter-node distances. Hence, we will use multi-hop to estimate the distance between any node and anchors.

The local connectivity information provided by the radio defines an unweighted graph, where the vertices are sensor nodes, and edges represent direct radio links between nodes. The hop count (h_{ij}) between sensor nodes s_i and s_j is then defined as the length of the shortest path in the graph between s_i and s_j. Naively, if the hop count between s_i and s_j is h_{ij} then the distance between s_i and s_j, d_{ij}, is less than $R \times h_{ij}$, where R is again the maximum radio range. If the distance between two nodes is less than radio range R, a noisy measurement of the distance is given by [15]:

$$\hat{d}_{ij} = d_{ij}(1 + randn \times \eta). \quad (6)$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (7)$$

where (x_i, y_i), and (x_j, y_j) are the coordinates of two (adjacents) different nodes, d_{ij}, $\hat{d}_{ij}$ are the true and measured distance respectively, η (called noise factor) is a given number related to the accuracy of the distance measurement and *randn* is a standard normal random variable with zero mean and unit variance.

For estimate distance between node and anchor we use Dijkstra's algorithm to find this shortest path [24]. Hence, the estimated distance between any node and the all anchors is the sum of estimated distance d_{ij} between all inter node (i and j) along the shortest path.

The proposed algorithm should be run in centralized architecture. We suppose that the central node is equipped with large resources (i.e., memory, computation, power...)

to be able to make all computation. Since the proposed algorithm is centralized, each node needs to communicate the list of its radio neighbours to the central unit in charge of the computation. This information is necessary first to build the distance matrix D_C from node i to anchor j with elements given by $\{d_c\}_{i,j} = dist(i,j)$, and the hop -count between each nodes of network and anchors, Then for each node the central node sort the shortest path to the nearest anchors (four for our simulation) and it's the distance corresponding to this path. The matrix D_C and number of hop are the only input parameter required by the localization algorithm. Algorithm 3 contains the pseudo-code of the localization scheme.

Algorithm. 3: DE Based Localization

% Initialization
1: *noOfNodes, noOfAnchors, NoiseFactor, Radio,*
2: *DE parameters (NP , Iter, F, C).*
% Main Loop
3: **for** all nodes *noOfNodes* **do**
4: $(x_n, y_n) =$ random ()
5: **end for**
6: **for** all Anchors *noOfAnchors* **do**
7: $(x_m, y_m) =$ random ()
8: **end for**
9: **Input:** matrix D_C Distance matrix $\{d_c\}_{i,j} = dist(i,j)$, hop count (α_i) to nearest four Anchors % Using *Dijkstra's* algorithm
10: **for** *i* = 1: to *noOfNode* **do**
11: **for** Anchors *j* to four nearest Anchors of *i (M=4)* **do**
12: $f_i(\hat{x},\hat{y}) = d_j - \sqrt{(x_j-\hat{x})^2 + (y_j-\hat{y})^2}$.
13: $F_i(\hat{x},\hat{y}) = \sum_{i=1}^{M=4} \alpha_i^2 f_i^2(\hat{x},\hat{y})$
14: $[\hat{x},\hat{y}] = \arg\min(F_i(\hat{x},\hat{y}))$ % Using DE algorithms
15: **end for**
16: **end for**

The estimation location error (%) of positioning algorithm was computed as:

$$err = \frac{100}{N \times R} \sum_{i=1}^{N} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}\,\% \qquad (8)$$

First, in Fig. 3, we show an example of 400 nodes randomly deployed in a regular square area 100×100. The anchors are represented by a blue square. First, when no noise is introduced and only one-hop is used to estimate the distance, which is far from reality (an ideal situation). It can be seen that the proposed algorithms converge to the nearest position of each node (the estimated positions are represented by a red star). However an exact localization is impossible because of measurement noise and imperfection of multi-hop distance estimations.

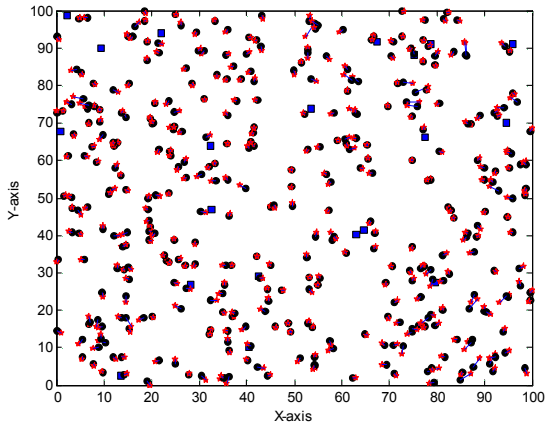

Fig. 3: Estimation results based on RCDE using distances between neighbors for noise factor = 0%, Ratio transmission =60. The 20 blue nodes are anchors; remaining nodes are non-anchors.

Now for the same problem we suppose that the measured distance between to adjacent node is inaccurate (which is more realistic) , Hence for each d_{ij} below the radio range is collapsed by a Gaussian noise with mean zero and variance equal to 5 % of the actual distance ($\eta = 0.05$). The transmission range is fixed to 10. In this case, the estimation positions are represented by a red star and an error offset line has been drawn between the true and the estimated locations (Fig. 4. (a)).

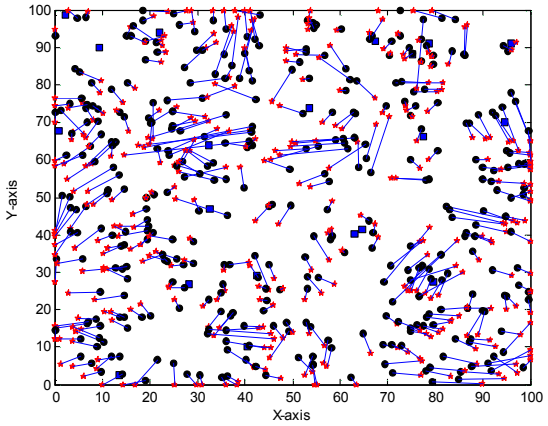

Fig. 4: Estimation results based on RCGA using distances between neighbors for N=200, Noise factor = 5%, Transmission Range =10. The 20 blue nodes are anchors; the original location of node is in black circle, the estimate locations are presented by red.

5.2 Performance under different transmission ranges

The connectivity (witch mean the number of neighbors per node) depends on the number of nodes per area (node density) and their radio transmission range.

In the sensor network if the transmission power of a node is increased, it will typically achieve a higher transmission range and therefore reach more other nodes via a direct link. On the other hand, if we make the transmission power of a node very low, the node may become isolated without any link to other nodes. Higher transmission range will cause more routing overhead and more interference to other nodes and therefore reduces the overall capacity of the network.

In practice, increasing the transmission range make the localization result more satisfactory. A relationship of transmission range versus connectivity is given in Table .1. These values are the calculated from 100 random configurations based on deployment of 200 nodes.

Table 1: Radio transmission versus connectivity averaged from 100 random configurations, with N=200 nodes.

Transmission Range	12	14	16	18	20
Min	8	11	14	17	21
Average	9.11	11.85	14.89	18.26	22.07
Max	10	13	16	20	24

Figure. 5 (a) shows the results for different transmission ranges, for different noise factor. As we would expect, the accuracy is higher when the transmission range is larger, which is higher network density.

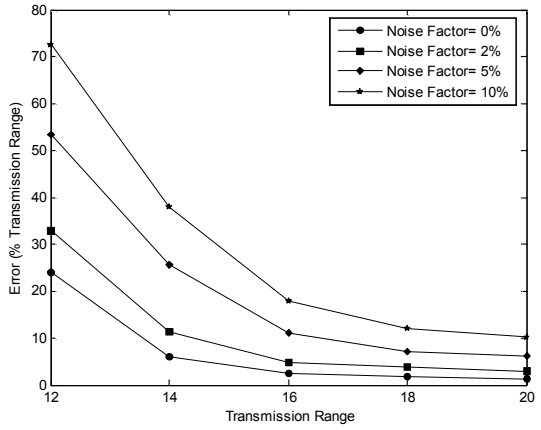

Fig. 5 (a): Variation of estimation error (RMSE) with transmission range for N=200 node, and 5 % of anchors and with 100 random realizations.

As the average connectivity level increases, the variation error decreases (Fig 5. (b)). Indeed, this shows that dense networks can provide more consistent average error values. This is due to the fact that dense networks have smaller multi-hop regions (two and more), which in turn lead to more accurate shortest path distances. These distances therefore improve the RCDE positing algorithm results, since more accurate distances translate into more accurate position estimation. However, since the transmission range is determined by transmission power, there is a tradeoff between energy-efficiency and localization accuracy.

5.3 Performance under different percentage of anchors

Given a sensor network with a few anchors; most nodes with unknown position should eventually drive good estimates of their positions. Locations near anchor and /or more near form many anchors should be most accurate, but the positioning accuracy should decline gracefully for more remote nodes.

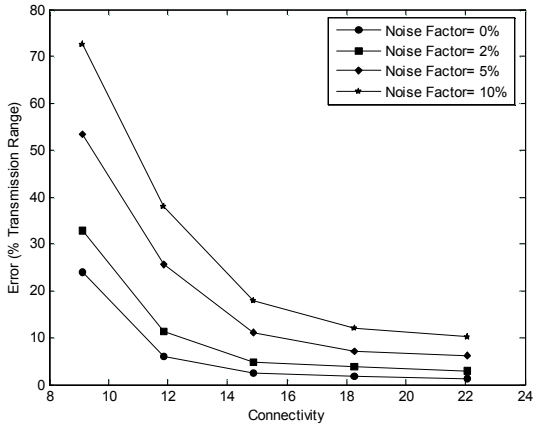

Fig. 5 (b): Variation of estimation error (RMSE) with connectivity for N=200 node, and 5 % of anchors and with 100 random realizations.

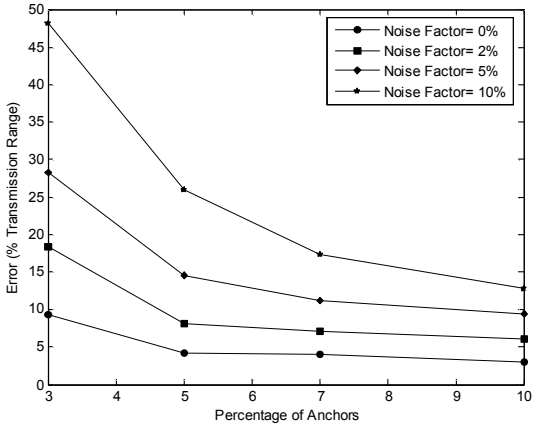

Fig. 6: Variation of estimation error (RMSE) with number of anchor randomly placed for N=200 node, Radio range R=15 for different noise factor and with 100 random realizations.

In practice, increasing the number of anchors makes localization result more satisfactory, but this also implies growing of the network cost. However it should be noted

that the localization depends closely with the type of desired application.

Figure 6 presents the average of estimation error as function of percentage anchor in networks. We see that whatever the noise error, increasing the percentage of anchors reduces the RMSD of error location. This reduction is, however, more important for high anchor percentage.

5.4 Performance under irregular area

In order to validate the performance of the proposed algorithm, we evaluate the localization algorithms under two irregular areas resulting from the presence of large obstacles in the region of the deployment. The first one is random C-shaped networks the second one is more complicated W-shaped maps, the both area are called soft and hard irregular area (Fig. 7 (a), and (b)).

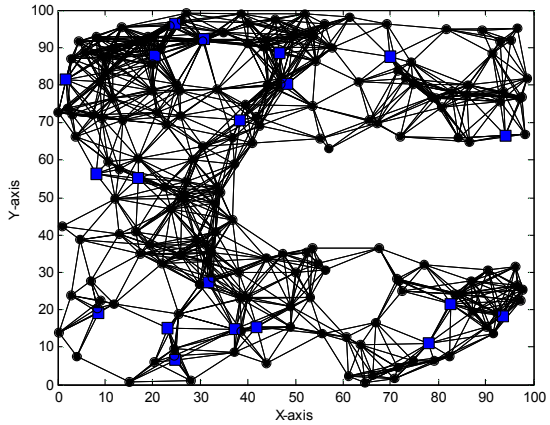

Fig. 7 (a): Example of random nodes deployment in "soft" irregular area (C-shaped maps), 20 blue nodes are anchors; remaining nodes (200) non-anchors.

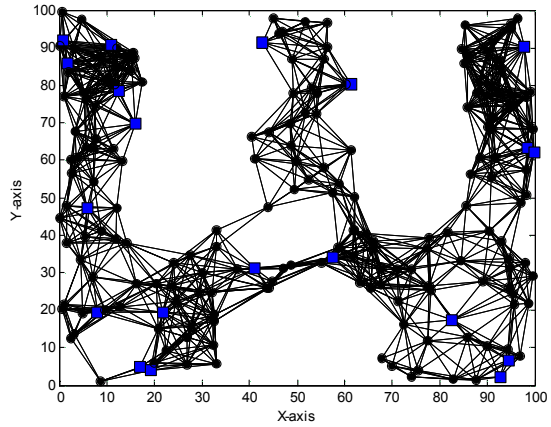

Fig. 7 (b): Example of random nodes deployment in "hard" irregular area, (W-shaped maps), 20 blue nodes are anchors; remaining nodes (200) non-anchors.

An example of estimation results based on RCGA in "soft" irregular area (C-shaped maps) for one configuration is shown in Fig. 8. Twenty blue nodes are anchors; remaining nodes (i.e., 200) are non-anchors, hence 10% anchor fraction is considered. The noise factor is equal to 5% and transmission range is fixed at 15. Discrepancies between the original and estimated nodes are indicated by red solid lines.

Figure. 9 shows the variation of estimation error averaged from 100 random realizations by increasing the measurement noise while varying the number of anchor 5 % and 10 % for both C-shaped maps and W-shaped maps irregular areas, where the radio range is fixed at 15. We can see that in both case the variation of estimation error increase while the of range error (noise factor) increase, however, the variation of estimation error for W-shaped maps is more sensitive to error range than C-shaped maps.

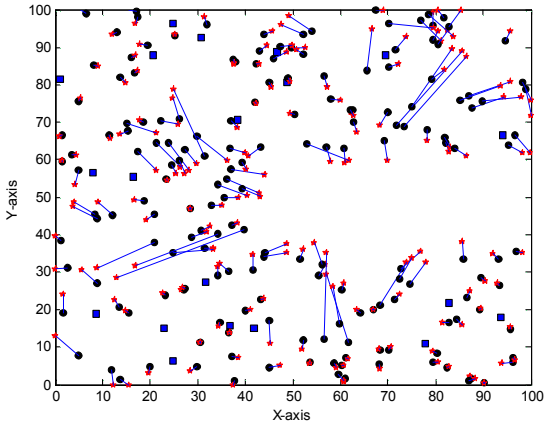

Fig. 8: Estimation results based on RCGA in "soft" irregular area (C-shaped maps) for one configuration.

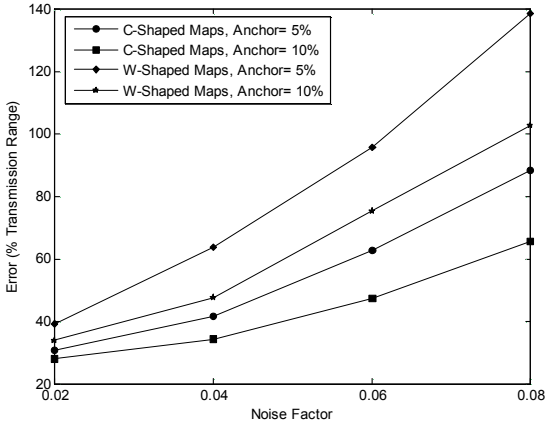

Fig. 9: Variation of RM estimation error (RMSE) with noise factor (from 2 to 8 %) for different number of anchor randomly placed, and for N=200 nodes, and R=15, 100 random realizations.

5.7 Comparison with others centralized methods

As mentioned above, the centralized algorithms for localization in wireless sensor has been already investigated in literature. However, is quite interesting to compare the performances of the proposed RCDE based localization algorithm with other centralized algorithms. The most cited centralized algorithm is Semidefinite Programming. Thus we compare the performance of RCDE algorithm with SDP results.

The simulations were performed in a sensor network of 200 nodes randomly distributed in a square of size. the noise factor is taken to 10%. The results of these simulations were compared with the ones obtained using the SDP approach with gradient search improvement [15], [16].

In Fig. 10, where the location estimation error is normalized by the transmission range while varying the percentage of anchor form 5 to 10%. All four combinations are quite sensitive to the radio range (connectivity). The sensitivity to the anchor fraction is quite similar. More anchors ease the localization task, especially for SDP. However RCDE algorithm has better accuracy than the SDP algorithm with gradient search particularly for low radio transmission and/or few anchors. For example, RCDE gives 72.66% of estimation error with 5% of anchor and a minimum connectivity of 8 nodes, while SDP gives 185.44%, this error was reduced to 30.2% for 10% of anchor fraction for RCDE versus 55.44% for SDP. This is an expected result of robustness of DE against convergence to false local minima compared to SDP; however increasing transmission ranges (and/or anchor fraction) the difference of error between both techniques decreases to gives approximately the same performances.

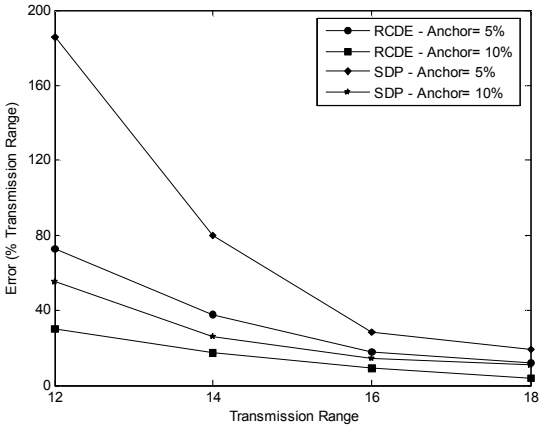

Fig. 10: Variation of estimation error (RMSE) results of RCDE and SDP algorithms, N= 200-node *random uniform* networks and 10 % of Noise Factor.

Finally, we compare RCDE with DV-hop and DV-distance using a standard least-squares approach. These last two algorithms are generally executed in distributed manner. However, for a legal comparison between the all techniques, we suppose that the DV-hop and DV-distance algorithms are also executed in centralised architecture.

The results of DV-hop and DV-distance are quite similar to the ones in [13]. While RCDE and DV-distance using local distance measures with 5% errors, DV-hop use only connectivity information. Since it (i.e., DV-hop) does not use range measurements, it is completely insensitive to this source of errors (see Section 2).

The results are shown in Figure 11 the RCDE algorithm is consistently much better than DV-hop and DV-distance. The RCDE's variations of estimation errors are just smaller of those estimated using a classical DV-distance when the transmission range is low (i.e., connectivity less than 8 nodes), RCDE's algorithm provides 55% of estimated error versus at least 150% for DV-hop and DV-distance. However it is worth noting that the computational cost of the DE approach is relatively higher than both algorithms.

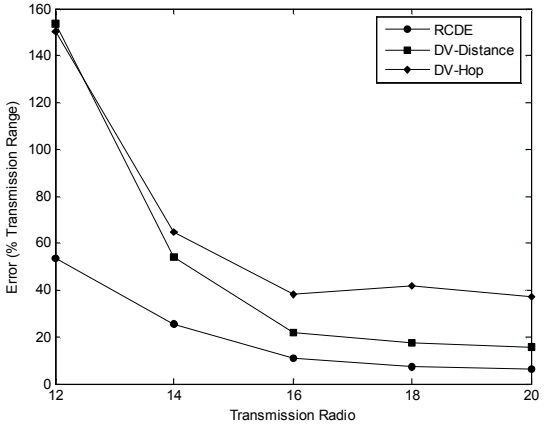

Fig. 11: Comparison of RCGA, DV-hop, and DV-distance on 200-node random uniform networks with 5% errors. 5% of anchor fraction.

6. Conclusion and future works

Wireless sensor network localization has attracted significant research interest. This interest is expected to grow further with the proliferation of wireless sensor network application. This paper describes a new centralized algorithm for WSN localization based on centralized Differential Evolution optimization.

Indeed, using the approach based a nonlinear least-squares optimization (RCDE) is statistically more justifiable that the usual procedure of finding intersection circles or hyperbolas (i.e., DV-distance). This approach becomes difficult if the hyperbolas or circles do not intersect at a point (i.e., due to measurement errors). Starting from this point, we have derivate an algorithm for sensor localization based on measurement approach between two adjacent nodes, multi-hop distance estimation

and number of hop between any node and some anchors. We consider the problem of minimizing a cost function based on above parameters.

We use reduced complexity Differential Evolution form number of population and iteration point of view, to find the global minimum of this cost function which corresponds to the nodes estimated coordinates.

The performance in terms of mean localization squared error is evaluated by simulations for different scenarios. The sensitivity of the estimated position to the uncertainty on distance measurements, radio transmission, number of anchors, tested in regular and irregular areas are evaluated. We have confirmed via simulations that DE based algorithm gives better accuracy for sensor localization in particular for low noise, high radio range and with more anchors.

Several directions for future work present themselves. First, we would like to make analysis of the performance under irregular radio transmission caused by devices and the propagation media (sending power, antenna gains, receiver sensitivity…). This assumption is more realistic since each node can use its own power transmission range. This analysis is also important since the relationship between radio range and accuracy of localization is highly correlated. We try also to analyze in more detail the impact of network connectivity for the localization accuracy and the impact of irregular shape area.

References

[1] K. Zhao, L. Guibas, *"Wireless sensor networks, an information processing approach"*, Morgan-Kaufman 2004.

[2] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, R. J. O'Dea, *"Relative location estimation in wireless sensor networks",* IEEE Transactions on Signal Processing, Vol. 51, Issue 8, pp. 2137 - 2148, August 2003.

[3] S. De, C. Qiao, H. Wu, *"Meshed multipath routing: an efficient strategy in wireless sensor networks",* IEEE Wireless Communications and Networking Conference (WCNC'03), New Orleans, March 2003.

[4] J. Heidemann, N. Bulusu, *"Using geospatial information in sensor networks"*, Proceedings of the CSTB Workshop on Intersection of Geospatial Information and Information Technology, 2001.

[5] A. Savvides, M. Srivastava, L. Girod, D. Estrin, *"Localization in Sensor Networks"*, Chapter 15 of Wireless Sensor Networks, Kluwer Academic Publishers, May 2004.

[6] G. Mao, B. Fidan, B. D. O. Anderson, *"Wireless sensor network localization techniques"*, Computer Network (2007), doi:10.1016/j.comnet.2006.11.018.

[7] S. S. Wang, K. P. Shih, C. Y. Chang, *"Distributed direction-based localization in wireless sensor networks",* Computer Communication (2007), doi:10.1016/ j.comcom. 2007.01.002.

[8] K. Langendoen, N. Reijers, *"Distributes localization in wireless sensor networks: a quantitative comparison*", Computer Networks, Vol. 43, pp. 499 – 518, 2003.

[9] Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, *"Localization from connectivity in sensor networks"*, IEEE Transactions on Parallel and Distributed Systems, Vol. 15, Issue. 11, pp. 961 - 974, 2004.

[10] N. Bulusu, J. Heidemann, D. Estrin, *"GPS-less low-cost outdoor localization for very small devices",* IEEE Personal Communications, Vol. 7, Issue. 5, pp. 28 - 34, 2000.

[11] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, *"Range free localization schemes for large scale sensor networks",* ACM International Conference on Mobile Computing and Networking (MOBICOM), pp. 81 - 95, September 2003.

[12] Y. Shang, W. Ruml, Y. Zhang, *"Localization from mere connectivity"*, International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), pp. 201 - 212, June 2003.

[13] D. Niculescu, B. Nath, *"Ad-hoc positioning systems (APS)",* IEEE Globecom '01, Vol. 5, p. 2926 - 2931, 25 - 29 November 2001.

[14] C. Savarese, J. Rabay, K. Langendoen, *"Robust Positioning algorithms for distributed ad hoc wireless sensor networks",* USENIX Technical Annual Conference, June 2002.

[15] T. C. Liang, T. C. Wang, Y. Ye, *"A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization",* formal report 5, Sanford University, 2004.

[16] P. Biswas, Y. Ye, *"Semidefinite programming for ad hoc wireless sensor network localization"*, in Proceedings of the third international symposium on Information processing in sensor networks, ACM Press, pp. 46 - 54, 2004.

[17] A. A. Kannan, G. Mao, B. Vucetic, *"Simulated annealing based localization in wireless sensor network",* IEEE Conference on Local Computer Networks, pp. 513 - 514, 15 - 17 November 2005.

[18] J. J. Ca_ery, G. L. Stüber, *"Subscriber Location in CDMA Cellular Networks",* IEEE Transactions on. Vehicular Technology, Vol. 47, No. 2, pp. 406 - 416, May 1998.

[19] B. H. Cheng, R. E. Hudson, F. Lorenzelli, L. Vandenberghe, K. Yao, *"Distributed Gauss-Newton method for node localization in wireless sensor networks",* IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, pp. 915 - 919, 5 - 8 June 2005.

[20] R. Moses, D. Krishnamurth, *R. Patterson,* "Self-localization for Wireless Networks", Eurasip Journal on Applied Signal Processing, pp. 348 - 358, 2003.

[21] R. Fletcher and M. J. D. Powsll, "A rapidly convergent descent method far minimization," Camputer Joumal, vol. 6, pp. 163-168, 1963.

[22] R. I. Fantana, E. Richley, and J. Bamey. "Commercialization of an ultrawideband precision asset location system" in Proc. IEEE Con on UWB systems and Technologies, 2003.

[23] R. Storn, "Differential Evolution - A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces". Journal of Global Optimization, Vol. 11, Dordrecht, pp. 341-359, 1997.

[24] P. W. Eklund, S. Kirkby, S. Pollitt, "A Dynamic Multi-source Dijkstra's Algorithm for Vehicle Routing", Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS '96), pp. 329 - 333, 1996.