

OPTIMIZATION SCHEMES FOR WIRELESS SENSOR NETWORK LOCALIZATION

EWA NIEWIADOMSKA-SZYNKIEWICZ^{*,**}, MICHAŁ MARKS^{*,**}

^{*} Research and Academic Computer Network
ul. Wąwózowa 18, 02–796 Warsaw, Poland

^{**} Institute of Control and Computation Engineering
Warsaw University of Technology, ul. Nowowiejska 15/19, 00–665 Warsaw, Poland
e-mail: {ens, mmarks}@ia.pw.edu.pl

Many applications of wireless sensor networks (WSN) require information about the geographical location of each sensor node. Self-organization and localization capabilities are one of the most important requirements in sensor networks. This paper provides an overview of centralized distance-based algorithms for estimating the positions of nodes in a sensor network. We discuss and compare three approaches: semidefinite programming, simulated annealing and two-phase stochastic optimization—a hybrid scheme that we have proposed. We analyze the properties of all listed methods and report the results of numerical tests. Particular attention is paid to our technique—the two-phase method—that uses a combination of trilateration, and stochastic optimization for performing sensor localization. We describe its performance in the case of centralized and distributed implementations.

Keywords: wireless sensor networks, localization, stochastic optimization, simulated annealing.

1. Introduction to localization techniques

Location awareness is required for many wireless sensor network applications, but it is often too expensive to include a GPS adapter in each sensor node. An approximated geographical location is needed for acquiring and managing data, geographic routing, geographic hash tables, energy conservation algorithms, and others. Hence, various localization schemes for assigning geographical coordinates to each node in a sensor network system are proposed. Localization methods should give a solution in a short time, achieve good accuracy even in the case of unevenly distributed nodes, and scale to large networks.

A large number of research and commercial location systems have been developed over the past two decades. A general survey is found in (Hightower and Borriello, 2001; de Brito and Peralta, 2007). Recently, some localization techniques have been proposed to allow estimating node location using information transmitted by a set of anchor nodes. An anchor is defined as a node that is aware of its own location, either through a GPS adapter or manual deployment at a point with known position.

The approaches to the localization differ in their as-

sumptions about network configuration and mobility, the distribution of the calculation process and, finally, hardware capabilities (Hu and Evans, 2004). We can distinguish four types of networks: static nodes, static anchor nodes and mobile non-anchors, mobile anchors and static non-anchors, mobile anchors and non-anchors. Location calculation can be conducted using one machine or calculations can be distributed. Centralized localization algorithms depend on sensor nodes transmitting data to a central computer, where calculation is performed to determine estimated location of each node. Distributed algorithms rely on self-organization of nodes in a sensor network. Each non-anchor node estimates its position based on the local data gathered from its neighbors. Requiring central computation involves high communication costs but, on the other hand, centralized methods are less complicated than distributed ones and they are likely to provide more accurate results.

Considering hardware capabilities, we can distinguish two classes of available localization methods: distance-based (also called range-based) and connectivity-based (also called range-free). Distance-based techniques use inter-sensor distance or angle measurements in loca-

tion calculation. In accordance with the available hardware, they exploit received signal strength, time difference between arrival of two different signals, the time of arrival or the angle of arrival. Generally, the proposed solutions are based on signal processing and algorithms transforming measurements into the coordinates of the nodes in the network. The approaches for designing centralized distance-based algorithms provided in the literature include multidimensional scaling (MDS) (Shang *et al.*, 2004; Ji and Zha, 2004), semidefinite programming (SDP) (Biswas and Ye, 2004) and simulated annealing (Kannan *et al.*, 2005; 2006). Connectivity-based algorithms use only the contents of the received messages to locate the entire sensor network. Popular connectivity-based algorithms are hop-counting techniques. Examples of connectivity-based methods are presented in (Niculescu and Nath, 2001; Doherty *et al.*, 2001) and (Shang *et al.*, 2004). Both distance-based and connectivity-based approaches have some weaknesses. Distance-based algorithms require additional equipment but through that we can reach fine resolution. Connectivity-based ones are cost effective but their performance is usually worse.

In this paper we compare four centralized distance-based localization algorithms: utilizing semidefinite programming (SDP), simulated annealing (SA), and our two schemes—trilateration and simulated annealing (TSA), and trilateration and genetic algorithm (TGA). TSA and TGA are hybrid techniques that use a combination of the geometry of triangles, and stochastic optimization. We discuss the properties of all listed approaches based on the results of simulation experiments. We present a localization performance as a function of several sensor network characteristics, such as network size, nodes density and deployment. In the final part of the paper we compare the performance of centralized and distributed versions of the TSA method.

2. Localization problem formulation

The mathematical model of the WSN localization problem can be described as follows: Assume we have a set of m sensors (anchor nodes), each with known position expressed as l -dimensional coordinates $a_k \in \mathbb{R}^l$, $k = 1, \dots, m$, and n sensors (non-anchor nodes) $x_j \in \mathbb{R}^l$, $j = 1, \dots, n$ with unknown locations. For each pair of two nodes in the network, we introduce the Euclidean distance, i.e., $d_{kj} = \|a_k - x_j\|$ between anchors and non-anchors, and $d_{ij} = \|x_i - x_j\|$ between two non-anchors, $j = 1, \dots, n$ and $i \neq j$.

In practical applications of distance-based methods, each sensor emits a known signal that allows the neighboring nodes to estimate their distances. We define sets of neighboring nodes as collections of nodes located within transmission ranges (radio ranges) of given nodes. Hence, for all anchors and non-anchors we introduce the follow-

ing sets:

$$\begin{aligned} N_k &= \{(k, j) : d_{kj} \leq r_k\}, \quad j = 1, \dots, n, \\ N_i &= \{(i, j) : d_{ij} \leq r_i\}, \quad j = 1, \dots, n, \end{aligned} \quad (1)$$

where r_k and r_i are parameters representing maximal transmission ranges of the k -th anchor node and the i -th non-anchor node, respectively.

The measured values \tilde{d}_{kj} and \tilde{d}_{ij} of true physical distances d_{kj} and d_{ij} are produced by measurement methods described in the literature (Anderson *et al.*, 2007). These methods involve measurement uncertainty, hence each distance value \tilde{d}_{kj} and \tilde{d}_{ij} represents the true distance corrupted with noise describing the uncertainty of the distance measurement,

$$\tilde{d}_{kj} = d_{kj} + \xi_{kj}, \quad \tilde{d}_{ij} = d_{ij} + \xi_{ij}, \quad (2)$$

where ξ_{kj} and ξ_{ij} denote measurement errors.

Hence, the general localization problem statement is as follows: Given noisy distance measurements \tilde{d}_{kj} , \tilde{d}_{ij} and positions of anchor nodes $a_k \in \mathbb{R}^l$, $k = 1, \dots, m$, estimate the locations of all nodes $x_j \in \mathbb{R}^l$, $j = 1, \dots, n$ with unknown positions. We can formulate the model of the optimization problem that minimizes the sum of errors in sensor positions for fitting the distance measurements. In the next two sections, we present two approaches to the problem solution. For simplicity, we assume that all sensors are placed on a plane.

2.1. Formulation for semidefinite programming.

The distance-based localization problem can be formulated as a quadratic optimization problem, and then transformed to a standard semidefinite programming problem (Boyd *et al.*, 1994). Semidefinite programs are a generalization of linear programs, so to solve the transformed problem we can use the existing linear solvers. Such formulation of the localization task is presented in (Biswas and Ye, 2004).

Let us consider the network of n non-anchors and m anchors. For each pair of two nodes, we introduce (based on measurements) the upper bound \tilde{d}_{kj}^{\max} and lower bound \tilde{d}_{kj}^{\min} to the Euclidean distance between a_k and x_j , and upper bound \tilde{d}_{ij}^{\max} and lower bound \tilde{d}_{ij}^{\min} to the Euclidean distance between x_i and x_j . Then, the model of the localization problem can be defined as

$$\min_{\hat{x}} \left\{ J_{SDP} = \sum_{k=1}^m \sum_{j \in N_k} e_{kj} + \sum_{i=1}^n \sum_{j \in N_i} e_{ij} \right\} \quad (3)$$

subject to

$$(\tilde{d}_{ij}^{\min})^2 - e_{ij} \leq \|\hat{x}_i - \hat{x}_j\|^2 \leq (\tilde{d}_{ij}^{\max})^2 + e_{ij}, \quad \forall i \neq j, \quad j \in N_i, \quad (4)$$

$$(\tilde{d}_{kj}^{\min})^2 - e_{kj} \leq \|a_k - \hat{x}_j\|^2 \leq (\tilde{d}_{kj}^{\max})^2 + e_{kj}, \quad \forall k, j, \quad j \in N_k \quad (5)$$

where $e_{kj} \geq 0$ and $e_{ij} \geq 0$ denote errors in sensor position estimates, \hat{x}_i and \hat{x}_j are estimated positions of nodes i and j , respectively, and N_i, N_k are sets of neighboring nodes defined in (1).

Let $X = [x_1, x_2, \dots, x_n]$ be the $2 \times n$ matrix that needs to be determined. Our problem may be transformed and formulated in matrix form. The following approach converts quadratic distance constraints into linear constraints by introducing a relaxation to remove the quadratic term in the formulation.

Let $Y = X^T X$. This constraint can be relaxed to $Y \succeq X^T X$, which is equivalent to a linear matrix inequality (e.g., (Boyd *et al.*, 1994)):

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0,$$

where the relation $Z \succeq 0$ means that Z is positive semidefinite, i.e., $y^T Z y \geq 0$ for all $y \in \mathbb{R}^n$. Then, the problem (3)–(5) can be formulated as a standard SDP problem:

$$\min_{\hat{x}} \left\{ J_{SDP} = \sum_{k=1}^m \sum_{j \in N_k} e_{kj} + \sum_{i=1}^n \sum_{j \in N_i} e_{ij} \right\} \quad (6)$$

subject to

$$(1; 0; \mathbf{0})^T Z (1; 0; \mathbf{0}) = 1, \quad (7)$$

$$(0; 1; \mathbf{0})^T Z (0; 1; \mathbf{0}) = 1, \quad (8)$$

$$(1; 1; \mathbf{0})^T Z (1; 1; \mathbf{0}) = 2, \quad (9)$$

$$(\tilde{d}_{ij}^{\min})^2 - e_{ij} \leq (\mathbf{0}; v_{ij})^T Z (\mathbf{0}; v_{ij}) \leq (\tilde{d}_{ij}^{\max})^2 + e_{ij}, \quad \forall i \neq j, \quad j \in N_i, \quad (10)$$

$$(\tilde{d}_{kj}^{\min})^2 - e_{kj} \leq (a_k; v_j)^T Z (a_k; v_j) \leq (\tilde{d}_{kj}^{\max})^2 + e_{kj}, \quad \forall k, j, \quad j \in N_k, \quad (11)$$

$$Z \succeq 0, \quad (12)$$

where v_{ij} is the vector with 1 in the i -th position, -1 in the j -th position and zero everywhere else; v_j is the vector of zeros except -1 in the j -th position, $e_{ij} \geq 0$ and $e_{kj} \geq 0$.

The optimization problem (6)–(12) can be effectively solved by interior point method based SDP solvers available for optimization research community, such as, for example, SeDuMI (Sturm, 1999) or CSDP (Borchers, 1999).

2.2. Formulation for stochastic optimization. Another approach is to formulate the optimization problem (13) with a nonlinear performance function and apply stochastic methods to solve it:

$$\min_{\hat{x}} \left\{ J_{SO} = \sum_{k=1}^m \sum_{j \in N_k} (\|a_k - \hat{x}_j\| - \tilde{d}_{kj})^2 + \sum_{i=1}^n \sum_{j \in N_i} (\|\hat{x}_i - \hat{x}_j\| - \tilde{d}_{ij})^2 \right\}, \quad (13)$$

where \hat{x}_i and \hat{x}_j denote, respectively, estimated positions of nodes i and j , \tilde{d}_{kj} and \tilde{d}_{ij} are measured distances between the pairs of nodes (k, j) and (i, j) , and N_i, N_k are sets of neighboring nodes defined in (1).

Three stochastic methods, SA, TSA and TGA, were used to solve the optimization problem (13).

2.2.1. Simulated annealing method. Simulated annealing is a well-known heuristic used to solve the localization problem (13) (Anderson *et al.*, 2007; Kannan *et al.*, 2005; 2006). It is implemented as a computer simulation of a stochastic process. It performs point-to-point transformation. For our research, we implemented the SA method similar to the algorithm presented in (Kannan *et al.*, 2005). It is a classical version of SA with one modification—the cooling process is slowed down. At each value of the coordinating parameter T (temperature), not one but $P \cdot n$ non-anchor nodes are randomly selected for modification (where n denotes the number of sensors with unknown positions in the network and P is a reasonably large number to lead the system into thermal equilibrium). The SA algorithm is presented Algorithm 1 as below.

In each iteration of the algorithm, a new solution, i.e., a vector of new nodes' positions is calculated. A given node is randomly selected and is moved in random direction at distance Δd . The value of Δd depends on the control parameter T and is restricted by the shrinking factor $\beta < 1$, $\Delta d_{\text{new}} = \beta \cdot \Delta d$. A simple cooling scheme is proposed, $T_{\text{new}} = \alpha \cdot T$, where T is the previous value of the temperature parameter. The coordinating parameters of the SA method, i.e., the initial temperature T_0 , T_f , Δd_0 , P , α and β have to be experimentally tuned.

2.2.2. TSA and TGA methods. The hybrid localization techniques TSA and TGA use a combination of the

Algorithm 1 Simulated annealing algorithm

```

1:  $T = T_0, T_0$ —initial temperature,  $T_f$ —final temperature
2:  $\Delta d = \Delta d_0, \Delta d_0$ —initial move distance
3: while  $T > T_f$  do
4:   for  $i = 1$  to  $P \cdot n$  do
5:     select a node to perturb
6:     generate a random direction and move a node at distance  $\Delta d$ 
7:     evaluate the change in the cost function,  $\Delta J_{SO}$ 
8:     if ( $\Delta J_{SO} \leq 0$ ) then
9:       //downhill move  $\Rightarrow$  accept it
10:      accept this perturbation and update the solution
11:     else
12:       //uphill move  $\Rightarrow$  accept with probability
13:       pick a random probability  $rp = \text{uniform}(0,1)$ 
14:       if ( $rp \leq \exp(-\Delta J_{SO}/T)$ ) then
15:         accept this perturbation and update the solution
16:       else
17:         reject this perturbation and keep the old solution
18:       end if
19:     end if
20:   end for
21:   change the temperature:  $T_{\text{new}} = \alpha \cdot T, T = T_{\text{new}}$ 
22:   change the distance  $\Delta d_{\text{new}} = \beta \cdot \Delta d, \Delta d = \Delta d_{\text{new}}$ 
23: end while
    
```

trilateration method, and stochastic optimization (Marks and Niewiadomska-Szynkiewicz, 2007). The algorithms operate in two phases. In the first phase, an auxiliary solution (initial localization) is provided. The solution of the first phase is modified by applying stochastic optimization methods. Various techniques, such as simulated annealing (Dekkers and Aarts, 1991) or genetic algorithms (Goldberg, 1989), can be applied. In this paper we compare both of them. Additional functionality (correction) is introduced to the second phase of the algorithm to remove incorrect solutions involved by the distances measurement errors.

Phase I—Trilateration. In the first phase a simple method of determining the relative positions of objects using the geometry of triangles is applied. Trilateration uses the known locations of anchor nodes a_k , $k = 1, \dots, m$, and the measured distances between pairs of nodes. To accurately and uniquely determine relative location of a non-anchor on a 2D plane using trilateration alone, generally at least three neighbors with known positions are needed. Hence, all nodes are divided into two groups: group $G1$, containing m nodes with known location (at the beginning only the anchor nodes), and group $G2$, of n nodes with unknown location. In each step of the algorithm, node i , where $i = 1, \dots, n$ from the group $G2$ is chosen. Next, the three nodes from the group $G1$ that are within node i radio range are randomly selected. If such nodes exist, the location of node i is calculated based on inter-nodes distances between three nodes selected from the group $G1$ and the measured distances between node i and these three nodes. The localized node i is moved to the group $G1$. Otherwise, another node from the group $G2$ is selected, and the operation is repeated. The first phase stops when there are no more nodes that can be localized based on the available information about all nodes localization. It switches to the second phase.

Phase II—Stochastic optimization. Due to distance measurement uncertainty, the coordinates calculated in Phase I are estimated with non-zero errors. In addition, the position of nodes that have less than three localized neighbors cannot be estimated. In the second phase, stochastic optimization is applied to increase the accuracy of the location estimation calculated in Phase I. Two variants are proposed—TSA utilizing the simulated annealing algorithm as presented in Algorithm 1, and TGA utilizing a genetic algorithm. In TGA, abstract representations of candidate solutions (chromosomes) are vectors of a random variable—coordinates of all non-anchor nodes, $[x_1, y_1, x_2, y_2, \dots, x_n, y_n]$, $x_i, y_i \in \mathbb{R}$. The initial population consists of a set of such chromosomes. The fitness function is defined in (13). At the reproduction (pre-selection) stage, tournament selection of size two is used.

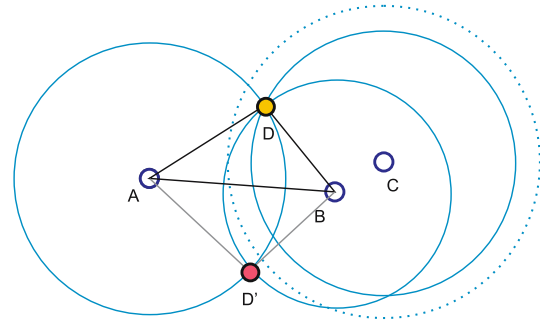


Fig. 1. Correct and incorrect node locations.

The crossover operator is defined as discrete recombination similar to elements exchanging applied to binary vectors. Both coordinates are recombined simultaneously. The mutation operator modifies the components of a given chromosome by adding a vector of generated $2 \cdot n$ Gaussian random variables. The elitist succession model is chosen.

Phase II—Correction. From the simulation experiments it was observed that the increased value of the location error is usually driven by incorrect location estimates calculated for a few nodes. Usually, the number of such incorrect results is not big, so the influence on the cost function (13) is not very strong and it is ignored by the optimization algorithm. This phenomenon is depicted in Fig. 1.

Let us consider three nodes, A, B, C , with known locations and measured distances between the node D to be localized and the nodes A, B, C . We can determine the coordinates of the nodes D and D' based on the distances d_{AD} and d_{BD} . As a final result, the location D should be selected, but in the case of non-zero measurement errors the incorrect location, i.e., D' will be chosen. The low quality of measurements may involve inaccurate location estimates.

To alleviate this phenomenon, new functionality is added to our algorithm—the correction operation performed during the second phase execution. Its objective is to make correction in the location estimates. The correction is triggered every iteration of the optimization process when the value of the performance function J_{SO} defined in (13) is less than the assumed threshold value θ , and the following algorithm is performed. Three nodes from the group of neighbors of a given node i that violate a smaller number of the constraints (1) than the other nodes are randomly selected. The roulette selection algorithm is used. Next, the position of node i is calculated based on the selected nodes locations. If the new location is more accurate, i.e., it violates a smaller number of the constraints (1), it replaces the previous one. The correction operation is repeated until all constraints are fulfilled or the assumed number of iterations is achieved.

The threshold value θ used in the correction operation depends on the number of anchor nodes, the deployment of all nodes in the area, the power of radio devices and the expected noise measurement factor nf . It is calculated according to the following formula:

$$\theta = \begin{cases} \mu \cdot nf \cdot s^2, & \frac{m}{n+m} < \gamma, \\ \lambda \cdot nf \cdot s^2, & \frac{m}{n+m} \geq \gamma, \end{cases} \quad (14)$$

where nf is a noise measurement factor, μ , λ and γ are experimentally tuned parameters. The variable s denotes the average number of neighbors of all nodes creating a network:

$$s = \frac{1}{n+m} \sum_{i=1}^{n+m} \sum_{j \in N_i} c_{ij}, \quad (15)$$

where

$$c_{ij} = \begin{cases} 1, & j \in N_i, \\ 0, & j \notin N_i, \end{cases}$$

where m is the number of anchor nodes, n is the number of non-anchors, c_{ij} is the connectivity between i and j nodes, and N_i is a set of neighbors of node i .

The SDP, SA, TSA and TGA schemes in their original versions require centralized computation. Namely, all nodes must transmit their destination information to a central computer to solve the optimization problems (3)–(5) or (13). In this paper we focus on the location estimation task and no further consideration is given to communication aspects (protocols, bandwidth limitation, congestions, etc.).

3. Numerical results

In order to evaluate the described approaches to sensor network localization, many numerical tests were performed for multihop network topologies. We compared the results obtained using semidefinite programming, pure simulated annealing and our two-phase algorithms—the TSA and TGA schemes. We performed a variety of simulation experiments to cover a wide range of network system configurations including the size of the network (number of nodes), the number of anchor nodes, anchor nodes deployment, the radio range, the distance measurement error and computation time. The key metric for evaluating all listed methods was the accuracy of the location estimates versus the deployment, equipment, communication and computation cost.

3.1. Simulation parameters. The sensor networks with 200–10000 nodes with randomly generated positions in a square region $[0, 1] \times [0, 1]$ were considered. The number of anchor nodes was 10% of all nodes. We assumed a fixed transmission range r for all devices in each considered network, equal to that presented in Table 1.

Table 1. Assumed transmission ranges for different networks.

Number of nodes	Radio range (r)
200	0.18
500	0.12
1000	0.08
2000	0.06
4000	0.04
5000	0.04
10000	0.03

We assumed the following values of the parameters used by the SA Algorithm 1: $\alpha = 0.8$, $\beta = 0.94$, $T_0 = 0.1$, $T_f = 10e-12$, $\Delta d_0 = 0.1$ and $P = 4$. The value of P was calculated as a solution of the two-criterion optimization problem. The obtained result was a trade off between the accuracy of location estimation and the time of calculation. The parameters μ , λ and γ in (14) used to calculate the threshold value in the correction operation were $\mu = 0.2$, $\lambda = 0.1$ and $\gamma = 0.05$.

For the purpose of simulation experiments, we need the values of measured distances between neighboring nodes i and j . To compare the performance of our two-phase scheme with algorithms developed by other researchers, we assumed that the measurement errors ξ_{kj} and ξ_{ij} in (2) were independent zero-mean Gaussian random variables of unit variance. Such an approach is provided in most of the literature devoted to WSN localization (Anderson *et al.*, 2007; Kannan *et al.*, 2006). Hence, in our experiments the measured distances \tilde{d}_{kj} and \tilde{d}_{ij} were assumed to be

$$\begin{aligned} \tilde{d}_{kj} &= d_{kj}(1 + \text{randn}()) \cdot nf, \\ \tilde{d}_{ij} &= d_{ij}(1 + \text{randn}()) \cdot nf, \end{aligned} \quad (16)$$

where d_{kj} and d_{ij} denote true physical distances between pairs of nodes, and nf is a noise factor. In most of the presented experiments, we assumed $nf=10\%$. We realize that the uncorrelated-Gaussian measurement error does not capture all practical cases but it is a good starting point for exposing error trends in sensor networks.

Due to measurement uncertainty, it is difficult to find a good metric to compare the results obtained using different localization methods. To evaluate the performance of the tested algorithms, we used the mean error between the estimated and the true physical location of non-anchor nodes in the network, defined as follows:

$$LE = \frac{1}{n} \sum_{i=1}^n \frac{(\|\hat{x}_i - x_i\|)^2}{r_i^2} 100\%, \quad (17)$$

where LE denotes a localization error, x_i is the true position of sensor node i in the network, \hat{x}_i is the estimated location of sensor node i , and r_i is the radio transmission range of the node i . The localization error is expressed as a percentage error. It is normalized with respect to the

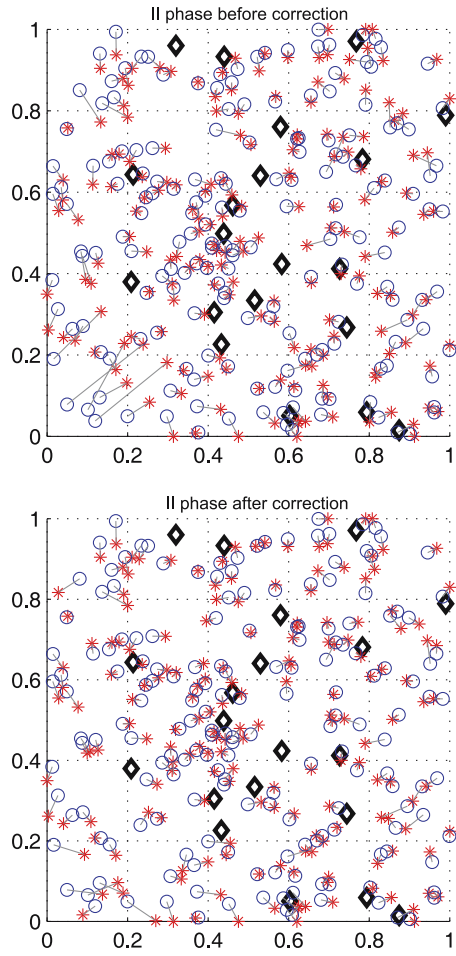


Fig. 2. Influence of correction on localization accuracy.

radio range to allow a comparison of results obtained for different size and range networks.

The calculations were carried out using an Intel Core2 Duo E6600, 2.4GHz, 2GB RAM computer. The presented experimental results are the average of five executions of each task. They are summarized in tables and figures.

3.2. Two-phase method evaluation. This section provides a detailed analysis of the performance of the two-phase scheme for node localization. We analyze the properties of our technique and report the results of numerical tests performed for a wide range of network system configurations including the size of the network (number of nodes), the number of anchor nodes, the radio range, the distance measurement error and computation time. We present the results obtained for the TSA variant of the two-phase technique.

3.2.1. TSA operation. The results of the first series of simulations presented in Table 2 and Figs. 2–3 show the estimates of the positions of non-anchor nodes calculated

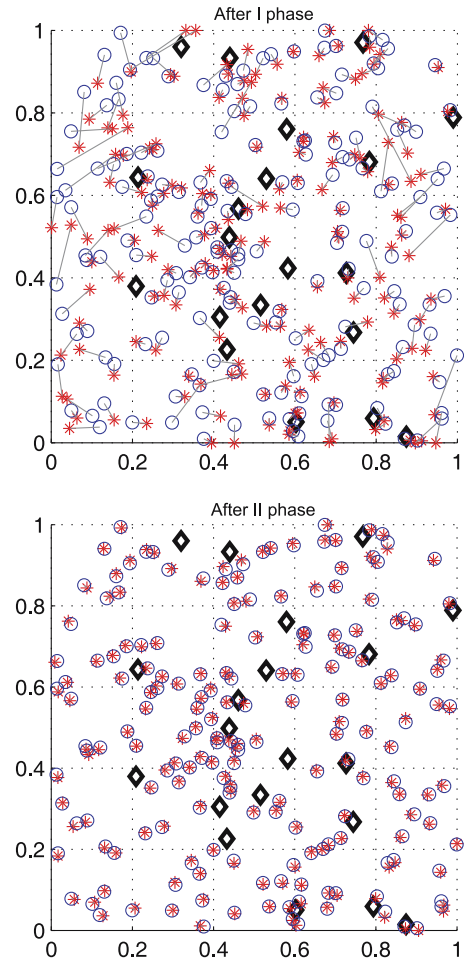


Fig. 3. Final network nodes localization (Phases I and II).

Table 2. Localization errors in two phases of the TSA method.

Phase	Localization error (LE) [%]
I	10.64
II	0.14

in both phases of the TSA method, and before and after correction operation. A sensor network consisting of 200 nodes—20 anchor nodes (marked with diamonds) and 180 non-anchor nodes (marked with circles) was considered. The estimated positions of non-anchor nodes are marked with stars in the figure. The localization error as defined in (17) is denoted by lines connecting the real and estimated locations.

As has been described, the TSA scheme consists of two phases and additional functionality—correction—introduced to the second phase. The objective of the correction is to remove incorrect solutions involved by the distance measurement errors. Figure 2 presents the results of the correction operation. The coordinate estimates calculated at the first iteration of the second phase, and before and after the correction are compared. From the simulation experiments we observed that the localization

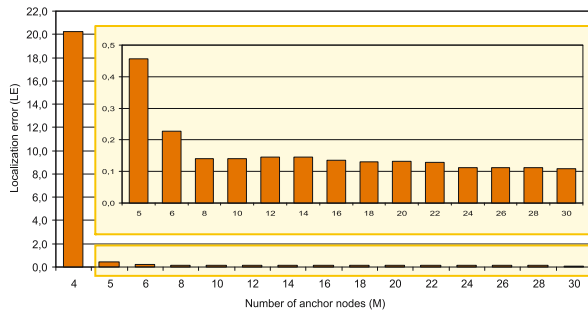


Fig. 4. Impact of the network size on localization accuracy.

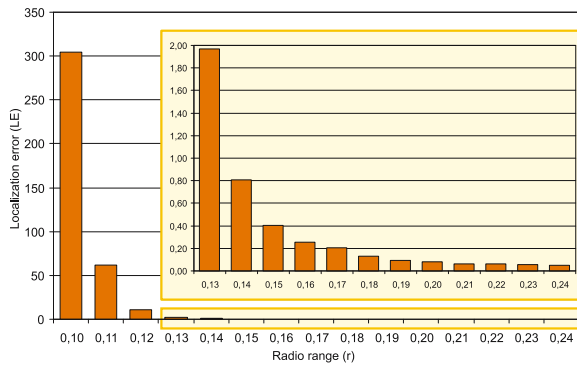


Fig. 5. Impact of the radio range on localization accuracy.

error was reduced from 6.25% to 3.04% after the correction operation. The value of reduction depends on nodes deployment. Figure 3 and Table 2 present the calculated positions of nodes – the final results of both phases.

3.2.2. Number of anchor nodes. We analyzed the effect of varying the number of anchors on the localization error (17). The results for the TSA method are summarized in Fig. 4. It shows that the localization error decreases as the number of anchor nodes increases. Increasing the density of anchors makes location easier, but it increases the network size and deployment costs. A trade off is necessary.

3.2.3. Radio range. The value of the transmission range r determines the number of neighbors of each node in the network. It has a substantial impact on localization accuracy depending on distance-based techniques. At a low level of r , the first phase of the TSA algorithm may perform poorly. Figure 5 shows the impact of the value of r on the localization error. The radio range considered was from the interval $[0.1, 0.24]$.

3.2.4. Measurement uncertainty. The accuracy of localization techniques based on inter-node distance measurements strongly depends on measurement errors. Figure 6 shows the impact of the value of the noise factor nf

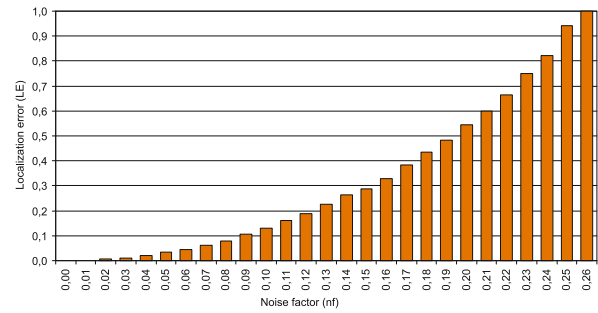


Fig. 6. Impact of distance measurement errors on localization accuracy.

Table 3. Localization error and computation times for different network sizes.

Number of nodes	Localization error (LE) [%]	Computation time [s]
200	0.16	0.44
500	0.11	2.44
1000	0.16	8.75

on the distance measurement (16) used in our simulations, and finally on the localization accuracy. The noise factor considered was from the range $[0, 26\%]$.

3.2.5. Computation time and scalability. In the described variant of the TSA method, the whole computations are performed centrally on a single machine. In the case of a large scale network, computation time can limit the application of centralized localization techniques. We performed multiple tests of the TSA application for sensor networks with different numbers of nodes. Selected results are presented in Table 3. From the experimental results we observed that the computation time increased proportionally to the square of the number of nodes.

3.3. Comparison of localization techniques.

3.3.1. Localization accuracy and computation time. Next, the results obtained for the TSA and TGA schemes were compared with those obtained applying the SDP and SA methods. Table 4 presents the localization errors and computation times obtained for all methods considered. From this table, we can see that the TSA and TGA methods estimate the location of nodes quite accurately, with the localization error less than 3% (especially in the case of TSA, with the localization error equal to 0.16%). The localization errors for the SDP and SA algorithms and the same network are about 30%. In this experiment the positions of anchor nodes were generated randomly. The same WSN network was considered in the case of all tested methods.

3.3.2. Node deployment. Localization accuracy strongly depends on node location in the network. In

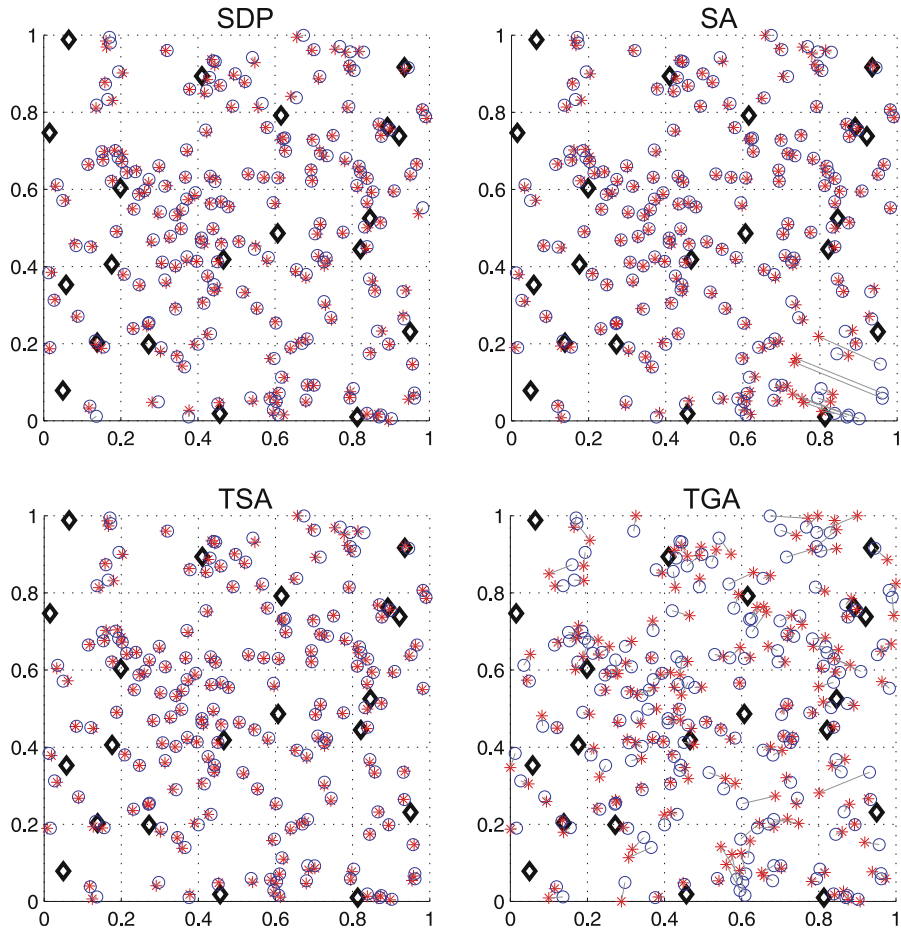


Fig. 7. Network with evenly distributed anchors.

Table 4. Computing time and localization errors—different methods.

Method	Localization error (<i>LE</i>) [%]	Computation time [s]
SDP	32.04	6.21
SA	34.34	2.94
TSA	0.16	0.44
TGA	2.34	2.91

the first series of experiments we analyzed networks with evenly distributed non-anchor nodes, and evenly and unevenly distributed anchors in the square region considered. The performance differences between the SDP, SA, TSA and TGA methods were compared.

If the anchor nodes are evenly distributed, then we obtain quite accurate solutions using all four methods (see Fig. 7), otherwise the results of location estimation are much worse and in some cases not satisfied (see Fig. 8). Table 5 and Fig. 8 show that the TSA method is much more robust with respect to the anchor distribution than the SDP and SA ones.

In contrast to the SDP and SA methods, TSA gives satisfactory accuracy in location estimation for networks with anchor nodes deployed only in a part of the region to

Table 5. Computing time and localization error—various deployment of anchor nodes.

Task	Method	Localization error [%]	Computation time [s]	
anchors evenly distributed	SDP	0.18	6.95	
	SA	2.76	3.04	
	TSA	0.13	0.46	
	TGA	3.80	2.85	
anchors unevenly distributed	a	SDP	174.91	5.51
		SA	233.89	2.85
		TSA	1.78	0.44
		TGA	20.61	2.34
	b	SDP	330.56	6.25
		SA	293.01	3.06
		TSA	1.81	0.47
		TGA	56.06	2.90
	c	SDP	434.83	8.95
		SA	446.13	3.84
		TSA	433.09	0.61
		TGA	133.78	3.46

be covered by sensors, cf. Figs. 8(a) and 8(b) and evenly distributed non-anchors.

Next, the experiments with both anchors and non-anchors only in some parts of the considered domain con-

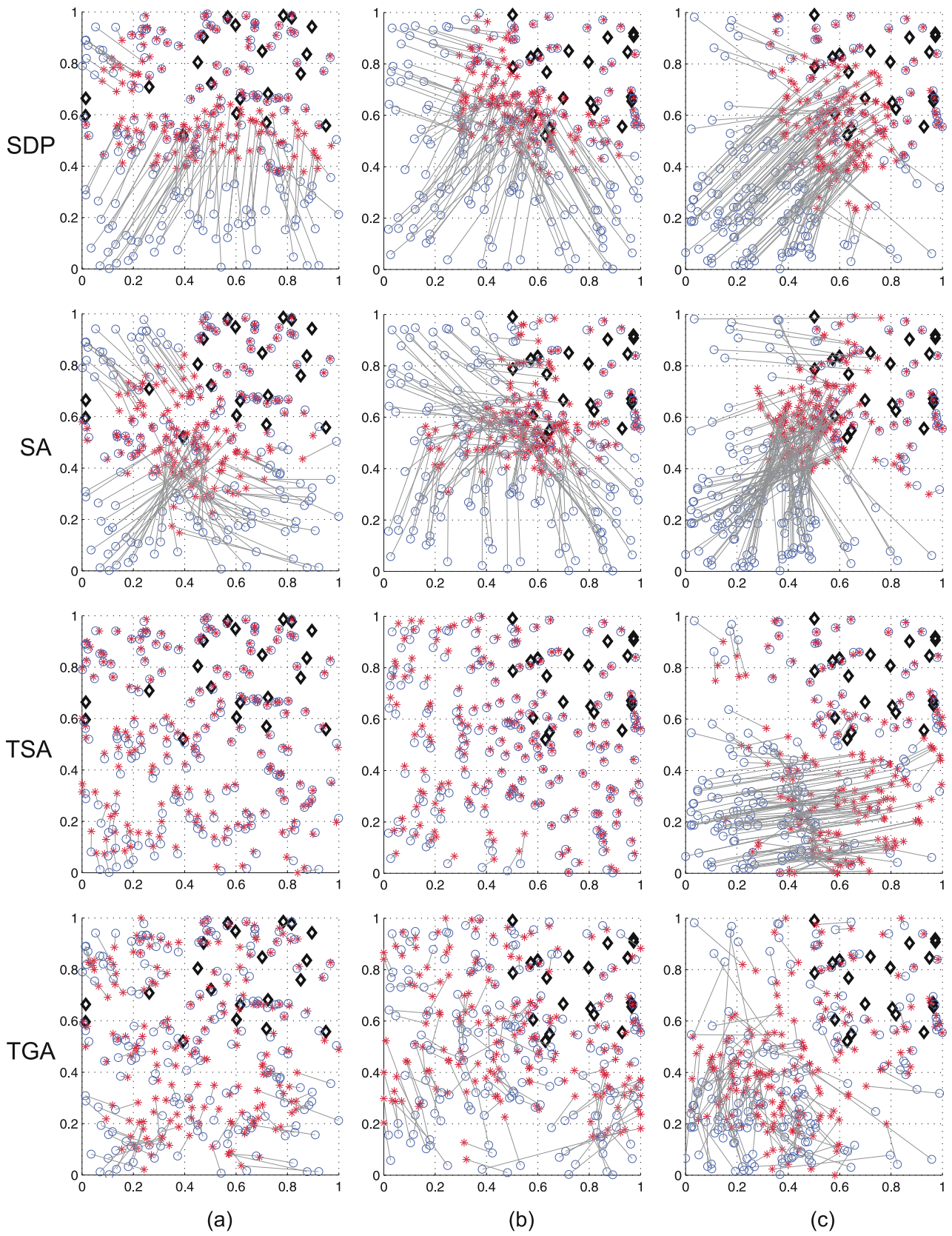


Fig. 8. Network with unevenly distributed anchors.

centrated were performed. From Table 5 and Fig. 8 we see that TSA, similarly to SDP and SA, gives an unsatisfactory solution. As a final result, we can say that it is not suggested to apply distance-based localization methods to networks with unevenly distributed non-anchor and anchor nodes.

4. Distributed version of the two-phase scheme

4.1. Implementation. The centralized two-phase method provides quite accurate location estimates even in the case of unevenly distributed nodes with known positions. However, we have to gather measurements of distances between all pairs of network nodes in a single computer to solve the optimization problem (13). The data transmission to the central station involves time delays, high communication cost and high energy consumption. Because of these disadvantages, centralized techniques cannot be employed in many applications (e.g., mobile networks and large scale networks).

In contrast to the centralized method, we proposed a fully distributed technique where all nodes are involved in the calculation process. Each node is responsible for determining its position using information about its neighbors. It offers a significant reduction in computation requirements because usually the set of neighbors consists of several nodes, so the number of connections is usually a few orders of magnitude less. A fully distributed computation model is also tolerant to node failures, and it distributes the communication cost evenly across the nodes. On the other hand, in the case of a distributed variant of the algorithm, a loss of information may occur. There are two reasons for that: the loss of information due to parallel computation and the loss of information due to an incomplete network map.

The loss of information due to parallel computation is connected with the way how optimization in the second phase of the method is performed. For example, in TSA, at each value of the coordinating parameter T (temperature), P times n non-anchor nodes are randomly selected for modification, i.e., coordinate estimates of chosen nodes are perturbed with a small distance Δd in a random direction. Modifications are done in order—the location of a current node is determined based on the previous transformations. In a distributed algorithm each node has to perform P small displacements. This is done in parallel and the information is exchanged between neighboring nodes every P iterations. An example is presented in Fig. 9. The node considered (marked with a square) updates the location using the information about previous positions of its neighbor nodes (lighter circles). Because all movements are done independently and simultaneously, there is no guarantee that the performance value is better after the iteration.

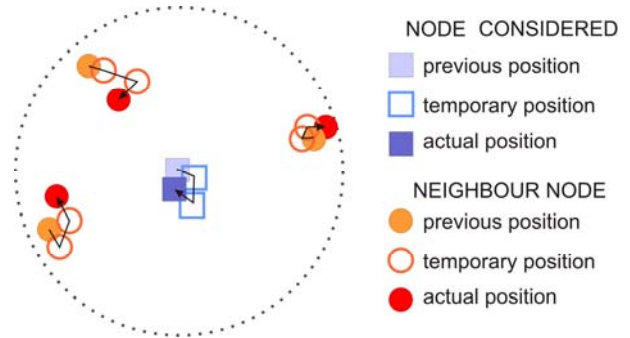


Fig. 9. Parallel node displacements.

Table 6. Localization errors for one-hop and two-hop correction—different deployment of anchors.

Task	Localization error (LE) [%]	
	1-hop neighbor correction	2-hop neighbor correction
anchors evenly distributed	0.34	0.31
anchors unevenly distributed	12.84	3.01

The loss of information due to an incomplete network map has a major impact on the correction. The correction operation depends on the neighborhood constraints (1) involved in the transmission range. In the centralized approach, a complete network state (location estimates of all nodes) is available. Due to this information, it is possible to detect the situation when the estimated distance between two neighbor nodes is greater than the transmission range. In the same way it can be detected that the estimated distance between two nodes is less than the transmission range, but due to the data of the whole network these nodes cannot be in neighborhood. In the distributed approach, the situation is different. Certainly, each node can detect the situation when the estimated distance between it and its neighbor is greater than the transmission range, but it is impossible to find out that from the estimated location it appears that the node is close to another node which is not a true neighbor. In order to overcome this problem, we propose to modify the correction operation. In the centralized scheme all neighbors are considered. In the distributed one, we distinguish two versions: the basic one—only one-hop neighbors are considered, and the extended one—both, one-hop and two-hop neighbors are considered. In Table 6, the results obtained for networks with evenly and unevenly distributed anchors with one-hop neighbor and two-hop neighbor correction are presented.

4.2. Comparison of centralized and distributed TSA.

Figure 10 presents the solution quality difference between

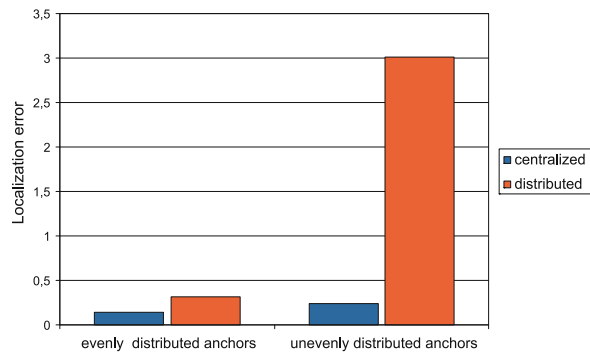


Fig. 10. Localization error for the centralized and distributed algorithms.

the centralized and distributed TSA algorithms (using two-hop correction). The obtained results confirm that, from the perspective of location estimation accuracy, the centralized algorithm provides more accurate location estimates than the distributed one. As a final result, we can say that for evenly distributed anchors we obtain quite accurate solutions using both methods, otherwise the results of location estimation are much worse in the case of the distributed version of our scheme.

5. Summary and conclusions

Many applications of wireless sensor networks depend on accurate determination of the positions of all network nodes. The paper provides a review of more representative localization techniques based on measurements of inter-sensor distances. Two methodologies for formulating a sensor network position estimation problem as a linear and a nonlinear task and the application of linear programming and stochastic algorithms to solve them are presented. We describe and investigate novel distance-based methods—two-phase TSA and TGA schemes—that combine simple geometry of triangles and stochastic optimization algorithms to determine the location of nodes with unknown positions in the sensor network. The originality in our approach is the combination of deterministic and stochastic techniques and adding the additional operation—correction—to simulated annealing and genetic algorithms. The simulation experiments demonstrate that our method gives accurate location estimates in sensible computing time even in the case of unevenly distributed nodes with known positions, and it outperforms localization schemes based on semidefinite programming and pure simulated annealing under a wide range of conditions. Numerical results confirm that TSA and TGA are efficient and robust localization algorithms.

Most works devoted to sensors localization describe techniques implemented a centralized manner. We compare the implementation of TSA both in a centralized and a distributed manner. The conclusion is that distributed

implementation improves scalability and reduces the calculation complexity, but it decreases location estimation accuracy, especially in the case of unevenly distributed nodes with known positions.

There are several directions for future research. First, we plan to extend our technique to mobile networks (MANET). Second, we are going to perform experiments using software environments for sensor networks simulation and, finally, apply the two-phase method to a testbed network of sensors in the laboratory to verify its properties in real-life applications.

Acknowledgment

This work was partially supported by the Polish Ministry of Science and Higher Education under Grant No. NN514 416934.

References

- Anderson, B. D. O., Mao, G. and Fidan, B. (2007). Wireless sensor network localization techniques, *Computer Networks* **51**(10): 2529–2553.
- Biswas, P. and Ye, Y. (2004). Semidefinite programming for ad hoc wireless sensor network localization, *IPSN '04: Proceedings of the 3-rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA*, ACM Press, New York, NY, pp. 46–54.
- Borchers, B. (1999). CSDP, a C library for semidefinite programming, *Optimization Methods & Software* **11**(1-4): 613–623.
- Boyd, S., Ghaoui, L. E., Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA.
- de Brito, L. M. P. L. and Peralta, L. M. R. (2007). Collaborative localization in wireless sensor networks, *SENSORCOMM 2007: Proceedings of the International Conference on Sensor Technologies and Applications, Valencia, Spain*, IEEE Computer Society, pp. 94–100.
- Dekkers, A. and Aarts, E. (1991). Global optimization and simulated annealing, *Mathematical Programming* **50**(8): 367–393.
- Doherty, L., Pister, K. and Ghaoui, L. E. (2001). Convex position estimation in wireless sensor networks, *INFOCOM 2001: Proceedings of the 20-th Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage, USA*, pp. 1655–1663.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Studies in Applied Mathematics, Addison-Wesley, Boston, MA.
- Hightower, J. and Borriello, G. (2001). Localization systems for ubiquitous computing, *Computer* **34**(8): 57–66.
- Hu, L. and Evans, D. (2004). Localization for mobile sensor networks, *MobiCom 2004: Proceedings of the 10-th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA*, IEEE Computer Society, pp. 45–57.

- Ji, X. and Zha, H. (2004). Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling, *INFOCOM 2004: Proceedings of the 23-rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China*, pp. 2652–2661.
- Kannan, A. A., Mao, G. and Vucetic, B. (2005). Simulated annealing based localization in wireless sensor network, *LCN '05: Proceedings of the IEEE Conference on Local Computer Networks. 30-th Anniversary, Sydney, Australia*, IEEE Computer Society, pp. 513–514.
- Kannan, A. A., Mao, G. and Vucetic, B. (2006). Simulated annealing based wireless sensor network localization with flip ambiguity mitigation, *Proceedings of the 63-rd IEEE Vehicular Technology Conference, Melbourne, Australia*, pp. 1022–1026.
- Marks, M. and Niewiadomska-Szynkiewicz, E. (2007). Two-phase stochastic optimization to sensor network localization, *SENSORCOMM 2007: Proceedings of the International Conference on Sensor Technologies and Applications, Valencia, Spain*, IEEE Computer Society, pp. 134–139.
- Niculescu, D. and Nath, B. (2001). Ad hoc positioning system (APS), *GLOBECOM: Proceeding of the Global Telecommunications Conference, San Antonio, CA, USA*, pp. 2926–2931.
- Shang, Y., Ruml, W., Zhang, Y. and Fromherz, M. (2004). Localization from connectivity in sensor networks, *IEEE Transactions on Parallel and Distributed Systems* **15**(11): 961–974.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods & Software* **11**(1-4): 625–653.



Ewa Niewiadomska-Szynkiewicz received her Ph.D. degree in 1996 and her D.Sc. degree in 2006 in control and computation engineering. Since 1987 she has been with the Warsaw University of Technology as the head of the Complex Systems Group and since 2000 with the Research and Academic Computer Network (NASK) as the head of the Network Modeling and Simulation Group. Her research interests focus on complex systems modeling and control, computer simulation, global optimization, parallel computations, and computer networks.



Michał Marks received his M.Sc. degree in computer science from the Warsaw University of Technology, Poland, in 2007. Currently he is a Ph.D. student at the Institute of Control and Computation Engineering of the same University. Since 2007 he has been with the Research and Academic Computer Network (NASK). His research area focuses on global optimization, multiple criteria optimization, decision support, and machine learning.

Received: 3 July 2008

Revised: 15 February 2009