A Study on the Application of Different Two-objective Evolutionary Algorithms to the Node Localization Problem in Wireless Sensor Networks

Massimo Vecchio, Roberto López–Valcarce
Departamento de Teoría de la Señal y las Comunicaciones, University of Vigo, C/ Maxwell s/n, 36310 Vigo - SPAIN
Email: {massimo,valcarce}@gts.uvigo.es

Francesco Marcelloni
Dipartimento di Ingegneria dell'Informazione, University of Pisa, Largo Lucio Lazzarino 1, 56122 Pisa - ITALY
Email: f.marcelloni@iet.unipi.it

***Abstract*—A number of applications of wireless sensor networks require to know the location of the sensor nodes. Typically, however, mainly due to costs and limited capacity of the batteries powering the sensor nodes, only a few nodes of the network, denoted *anchor nodes* in the literature, are endowed with their exact positions. Thus, given a number of anchor nodes, the problem of estimating the locations of all the nodes of a wireless sensor network has attracted a large interest in the last years. The localization task is based on the estimated distances between pairs of nodes in range of each other and is particularly hard in the most appealing scenario, that is, when the network connectivity is quite low. In a recent paper, we have proposed to tackle the localization problem as a two-objective optimization task with the localization accuracy and the number of connectivity constraints that are not satisfied by the candidate geometry as the two objectives. In this paper, we aim to evaluate the behavior of five state-of-the-art multi–objective evolutionary algorithms (MOEAs) in solving the localization problem on different network topologies. We show that one of these MOEAs, namely PAES, statistically outperforms the others in terms of localization error.**

***Keywords*-Stochastic Optimization; Multi–objective Evolutionary Algorithms; Range Measurements;**

I. Introduction

Wireless Sensor Networks (WSNs) consist of several low–cost nodes communicating among themselves for applications like environment monitoring, precision agriculture, vehicle tracking, etc. Tiny nodes are generally deployed in an area to be monitored, spanning potentially large geographical regions. The small size and low cost of sensor nodes impose several practical limitations: as they mount small and cheap memory and microprocessor units, tasks like large data storing and complex computations become unfeasible. At the same time, nodes are usually battery–powered, thus the network lifetime constitutes an important issue [1], [2].

In the aforementioned applications, knowledge about the location of sensor nodes may be essential. Although in principle the use of a Global Positioning System (GPS) could enable such "*location awareness*", this solution is not always viable in practice. First, in indoor and underground WSN deployments the communication with satellites may be unavailable; moreover the cost and the power consumption of a standard GPS receiver are generally not affordable by cheap, battery–powered nodes.

These limitations have motivated alternative approaches to the problem as reviewed in [3]. Among these, *fine–grained* localization techniques arise as a flexible option. In these schemes, only a few nodes of the network (termed *anchor nodes*) are endowed with their exact positions through GPS or manual placement, while all nodes are able to estimate their distances to nearby nodes by using some measurement technique. These distance–related techniques include Received Signal Strength (RSS) measurements, Time of Arrival (ToA), Time Difference of Arrival (TDoA), etc. Thus, assuming that the coordinates of anchor nodes are known, the fine–grained localization problem consists of determining the positions of all non–anchor nodes, by exploiting pairwise distance measurements among the nodes. This task has proved to be rather difficult: determining the locations of the nodes from a set of pairwise distance estimates is a nonconvex optimization problem; moreover, the measurements available to nodes are invariably corrupted by noise; finally, even if the distance estimates were perfectly accurate, sufficient conditions for the solution to be unique are not easily identified [4]. We will briefly discuss these issues in the following.

Assuming a statistical characterization of measurement noise (which will usually depend on the kind of measurement technique [3]), Maximum Likelihood (ML) estimation is the natural approach to the localization problem. However, as previously mentioned, the ML formulation results in a multivariable nonconvex optimization problem. Three different approaches to this task can be found in the literature, namely stochastic optimization, multidimensional scaling, and convex relaxation. The first approach attempts to avoid local maxima of the likelihood function by resorting to global optimization methods, such as *e.g.*

The work of M. Vecchio and R. López-Valcarce is supported by the Spanish Government and the European Regional Development Fund (ERDF) under projects DYNACS (TEC2010-21245-C02-02/TCM) and COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010).

simulated annealing [5]. Multidimensional scaling [6] is a *connectivity–based* technique that, in addition to distance measurements, exploits knowledge about the topology of the network; this information imposes additional constraints on the problem, since nodes within the communication ranges of each other cannot be arbitrarily far apart. The third approach relaxes the original nonconvex ML formulation in order to obtain a Semi–Definite Programming (SDP) or a Second–Order Cone Programming (SOCP) problems. Global solutions to these relaxed, convex problems can be then obtained with moderate computational effort [4] [7] and constitute approximate solutions to the original nonconvex problem. In [7] it was shown that the solutions obtained by SOCP relaxation are less accurate than those obtained by SDP relaxation.

In [8] we have modeled the localization problem as a two-objective optimization problem. The first objective, denoted *CF*, is given by the original nonconvex cost (the sum of squared differences between the estimated and the corresponding measured inter–node distances). The second objective, denoted *CV*, exploits the connectivity–based *a priori* information about the network, and is especially useful in order to alleviate *localizability* issues. We have tackled the optimization problem by using a multi-objective evolutionary algorithm, namely PAES [9]. We have shown that, on a variety of network topologies and connectivity ranges, PAES outperforms another metaheuristic technique based on simulated annealing [5]. In this paper, we apply five different state-of-the-art MOEAs in solving the localization problem on different network topologies. We show that PAES statistically outperforms the other MOEAs in terms of localization error, thus highlighting its particular effectiveness in this specific optimization problem.

The remainder of this paper is organized as follows. In Section II, we formulate the problem. Section III models the localization problem as a two-objective optimization problem. Section IV describes as the problem can be tackled by exploiting a multi-objective evolutionary approach and the five MOEAs used in the experiments. The experimental results are discussed in Section V. Finally, in Section VI we draw some conclusions.

II. Problem formulation

Let us consider a WSN with n nodes deployed in $\mathcal{T} = [0,1] \times [0,1] \subset \mathbb{R}^2$. We assume that nodes 1, ..., m, with $m < n$, are anchor nodes whose coordinates $\mathbf{p}_i = (x_i, y_i) \in \mathcal{T}$, $i = 1, \ldots, m$, are known. Further, we suppose that if two sensor nodes, say i and j, are within the communication range of each other, then their inter–node distance d_{ij} can be estimated by using some measurement technique (see Section I). Distance measurements d_{ij} are modeled as

$$d_{ij} = r_{ij} + e_{ij} \tag{1}$$

where $r_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ is the actual distance between nodes i and j ($\|\cdot\|$ denotes the Euclidean norm). Similar to [4], [5], we assume that measurement errors e_{ij} follow a zero–mean Gaussian distribution with variance σ^2. It is also assumed that the random variables e_{ij} and e_{kl} are statistically independent for $(i,j) \neq (k,l)$.

As commonly used in the literature, we adopt a simple disk model for network connectivity: nodes i and j can communicate with each other if and only if $r_{ij} \leq R$, where R is the connectivity range. We refer to nodes j such that $r_{ij} \leq R$ as *first–level neighbors* of node i. Further, we refer to all nodes j which are not first–level neighbors of node i, but which share at least a first–level neighbor with node i, as *second–level neighbors* of node i. Let

$$N_i = \{j \in 1 \ldots n, j \neq i : r_{ij} \leq R\} \tag{2}$$

$$\overline{N}_i = \{j \in 1 \ldots n, j \neq i : r_{ij} > R\} \tag{3}$$

be the set of the first–level neighbors of node i and its complement, respectively. We assume that sets N_i and $\overline{N}_i$ are known for all $i = 1, \ldots, n$. This is a reasonable assumption, since each node can easily determine which other nodes it can communicate with.

A. Geometrical constraints

The connectivity ranges and the positions of the anchor nodes determine subsets of the overall search space where each single non–anchor node can be positioned. These subsets depend on the type of non–anchor node, and can be defined by means of geometrical constraints. In [8], we have adopted the following classification based on the position of a non–anchor node with respect to anchor nodes:

- *Class 1 node:* a non–anchor node which is first–level neighbor to at least one anchor node.
- *Class 2 node:* a non–anchor node which is second–level neighbor to at least one anchor node.
- *Class 3 node:* a non–anchor node which belongs to neither class 1 nor class 2.

The knowledge of the membership of a non–anchor node to a class allows restricting the space where the node can be located. This information can be exploited both in the generation of the initial population of the MOEA and for constraining the application of the mating operators during the evolutionary process. Thus, by avoiding the generation of solutions which certainly cannot be optimal, it is possible to speed up the execution of the evolutionary algorithm.

III. Modeling the localization problem as a two-objective optimization problem

With the aim of estimating the positions of the non–anchor nodes as accurately as possible, in [8] we have proposed to concurrently minimize two objectives. Let $\hat{\mathbf{p}}_i = (\hat{x}_i, \hat{y}_i), i =$

$m+1, \ldots, n$ be the estimated positions of the non–anchor nodes i. The first objective *CF* is defined as

$$CF = \sum_{i=m+1}^{n} \left(\sum_{j \in N_i} \left(\hat{d}_{ij} - d_{ij} \right)^2 \right), \quad (4)$$

where $\hat{d}_{ij}$ is the estimated distance between nodes i and j computed as follows:

$$\hat{d}_{ij} = \begin{cases} \sqrt{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}, & \text{if node } j \text{ is an anchor,} \\ \sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}, & \text{otherwise.} \end{cases} \quad (5)$$

A network is said to be localizable if there is only one possible geometry compatible with the data, that is, with the set of anchor nodes and the inter–node distance measurements. If the network is not localizable, then multiple minima will be present in *CF*, with only one of them corresponding to the actual geometry of the deployment. Thus, in settings which are close to not being localizable, any localization algorithm will become extremely sensitive to these false minima of *CF*, resulting in very large localization errors [10]. The simplest effect leading to lack of localizability is the so–called *flip ambiguity* phenomenon, shown in Fig. 1. The neighbors of node i (*i.e.* nodes j, k, l and m) are almost collinear (double line in the figure), and thus, it is clear that if the location of node i is flipped with respect to this line to the new position denoted by i', then the new geometry so obtained is almost compatible with the original inter–node distance measurements (it would be fully compatible if nodes j, k, l and m were perfectly aligned). In Fig. 1, one can observe that whereas the flipped position i' is within the communication range of node n (shown by the circle in the figure), the actual position i is not, thus violating a connectivity constraint. The number of such violations in a candidate topology constitutes our second objective function *CV*.

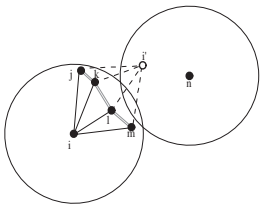

Figure 1. The flip ambiguity problem.

Formally, *CV* counts the number of connectivity constraints which are not satisfied by the candidate geometry, and is defined as

$$CV = \sum_{i=1}^{n} \left(\sum_{j \in N_i} \delta_{ij} + \sum_{j \in \overline{N}_i} (1 - \delta_{ij}) \right), \quad (6)$$

where $\delta_{ij} = 1$ if $\hat{d}_{ij} > R$ and 0 otherwise.

In order to evaluate the accuracy of the estimates, we consider the *normalized localization error* (NLE), defined as

$$NLE = \frac{1}{R} \sqrt{\frac{1}{(n-m)} \sum_{i=m+1}^{n} \left(\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2 \right)} \times 100\%. \quad (7)$$

Thus, assuming that the estimate is unbiased, NLE can be interpreted as the ratio of the standard deviation to the connectivity radius.

IV. The Multi-objective Evolutionary Approach

We have compared the effectiveness of five different MOEAs in tackling the localization problem in WSNs, namely the Pareto Archived Evolution Strategy (PAES) [9], the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [11], the Non–dominated Sorting Genetic Algorithm II (NSGA–II) [12], the Multi–Objective Evolutionary Algorithm based on Decomposition with Differential Evolution crossover [13] (MOEA/D–DE) [14] and an asynchronous version of the Multi–Objective Cellular algorithm 4 (aMOCell4) [15]. In [8], we have already applied PAES to the specific problem and have shown that PAES outperforms another metaheuristic approach, namely SAL, based on simulated annealing. Here, we aim to evaluate whether other different MOEAs can outperform PAES.

PAES, SPEA2 and NSGA-II are well-known and very popular MOEAs: they are often applied in the literature as benchmarks when proposing a novel MOEA. Due to space limitations, we will not describe these algorithms in detail.

The original MOEA based on decomposition (MOEA/D) was introduced by Zhang and Li in [16]. It relies on conventional aggregation approaches in which a multi–objective problem is decomposed into a number of scalar objective optimization problems (sub-problems). Each scalar objective represents a weighted aggregation of the single original objectives. Neighborhood relations among sub–problems are defined based on the distances between their aggregation weight vectors. Sub–problem i is considered a neighbor of sub–problem j if the weight vector of sub–problem i is close to that of sub–problem j. Each sub–problem is optimized in the MOEA/D by using information mainly from its neighboring sub–problems [17]. Zhang and Lin have also proven in [14] that using a DE crossover operator rather than classical simulated binary crossover (SBX) improves the performance of MOEA/D. By performing several different trials of the algorithms, we have verified that the parameter setup suggested in [14] provides the best performance and that MOEA/D with DE crossover (MOEA/D–DE) outperforms MOEA/D with SBX. For the sake of brevity, we will not discuss this experimentation in this paper.

Cellular genetic algorithms (CGAs) exploit the concept of small neighborhood in the sense that an individual may only interact with its nearby neighbors in the breeding loop.

The overlapped small neighborhoods help in exploring the search space because the induced slow diffusion of solutions through the population provides a kind of exploration (diversification), while exploitation (intensification) takes place inside each neighborhood by genetic operations. CGAs have proven to be very effective tools for solving a wide range of single objective optimization problems. Effective variants for the multi–objective framework are, however, still under development. One of the first attempts to extend CGAs to the multi–objective framework has been MOCell [18]. MOCell uses, like many other MOEAs, an external archive to store the non-dominated solutions and selects a fixed number of individuals from the archive to replace the same number of randomly chosen individuals from the population (archive feedback) at the end of each iteration of the algorithm. This helps speeding up the convergence by taking advantage of the search experience maturated in the previous iterations. This type of replacement coexists with the typical replacement of a canonical CGA, where the newly generated individual replaces the current one if the latter is worse than the former. In [15], the same authors of MOCell have also proposed six different variants and have concluded that the variant, namely aMOCell4, corresponding to the asynchronous version of MOCell which replaces the worst cell in the neighborhood and uses an individual from the archive in the selection operator, allows achieving the best results in their problem instances. We have tested the six different variants in our localization problem and have concluded that, also in our problem, aMOCell4 outperforms the other variants. For the sake of brevity, we omit this analysis.

A. The chromosome coding

In our optimization framework each chromosome encodes the positions of all non–anchor nodes in the network. Thus, each chromosome consists of $n - m$ pairs of real numbers, where each pair represents the coordinates $\hat{x}$ and $\hat{y}$ of a non–anchor node. We enforce compliance with constraints described in Section II-A in the initial population. Further, whenever mutations are applied during the evolutionary process, only mutated individuals satisfying these constraints are generated. Each chromosome is associated with a vector of two elements, which represent the values of the two objective functions CF and CV (Eqs. (4) and (6) in Section III).

B. The genetic operators

We have defined two mutation operators. The first mutation operator, denoted *Node Mutation* (NM) operator, performs a uniform–like mutation: the position of each non–anchor sensor node is mutated with probability $P_U = 1/(n-m)$. Positions are randomly generated within the geometrical constraints imposed on the specific node location.

The second mutation operator, denoted *neighborhood mutation* (NHM) operator, mutates, with probability $P_U = 1/(n-m)$, the position of each non–anchor sensor node within the geometrical constraints determined for the specific node, but unlike the first operator, it applies the same rigid translation (RT), which has brought the mutated node i from the pre–mutation position to the post–mutation position, to the neighbors of i with a certain probability. As we have already discussed in [8], NHM results to be particular suitable for dealing with particular topological configurations.

Except for PAES (that only uses mutation) and MOEA/D–DE (that uses DE crossover), the crossover operator used in the MOEAs considered in this paper is the classical one–point crossover operator, denoted CX. The common point is chosen by picking a random integer from $\{1, (n-m)\}$. We preferred the one–point crossover operator to more suitable crossover operators for real coding to avoid to check for each gene the compliance with the geometrical constraints. On the other hand, we verified that the synergy between the one-point crossover and the described mutation operators allows us to perform an adequate exploration of the search space.

V. Experimental results

A. Simulation setup

With the aim of comparing the five MOEAs, we built 10 different network topologies by randomly placing 200 nodes with a uniform distribution in $\mathcal{T} = [0,1] \times [0,1] \subset \mathbb{R}^2$. We fixed the percentage of anchor nodes to 10% and the connectivity range $R = 0.13$. The distance measurements between neighboring nodes were generated according to the model (1). We assume that these distance estimates are derived from RSS measurements, which are commonly affected by log–normal shadowing with standard deviation of the errors proportional to the actual range r_{ij} [3]. Thus, the variance of e_{ij} is given by $\sigma^2 = \alpha^2 r_{ij}^2$. A value of $\alpha = 0.1$ was used in the simulations.

Table I
Main network indicators for the different network topologies.

	node degree	Cl.1	Cl.2	0 anch.	3(or more) anch.
TOP0	9.34	108	61	72	14
TOP1	9.88	120	47	60	15
TOP2	9.34	129	46	51	7
TOP3	9.50	124	50	56	8
TOP4	9.70	121	57	59	8
TOP5	9.32	111	58	69	13
TOP6	9.15	126	48	54	6
TOP7	9.19	123	46	57	4
TOP8	9.21	100	66	80	12
TOP9	9.45	113	56	67	9

Table I shows, for each topology, the main network indicators, namely the node degree (considering anchor and non–anchor nodes), the number of non–anchor nodes classified in Class 1 and Class 2, the number of non–anchor nodes with no anchor node in their neighborhoods and the number

Table II
PARAMETER SETUP OF THE FIVE MULTI-OBJECTIVE EVOLUTIONARY STRATEGIES. P, N, S, D AND C DENOTE, RESPECTIVELY, PAES, NSGA–II, SPEA2, MOEA/D–DE AND AMOCELL4.

Parameter name	Value	P	N	S	D	C
Population size	20		✓	✓	✓	✓
Archive size	20	✓				
External Archive size	100			✓		✓
Number of regions	5	✓				
fitness evaluations	100,000	✓	✓	✓	✓	✓
NM probability	0.9	✓	✓	✓	✓	✓
RT probability	0.3	✓	✓	✓	✓	✓
CX probability	0.9		✓	✓		✓
DE probability	1.0				✓	

of non–anchor nodes having at least 3 anchor nodes in their neighborhoods. The analysis of Table I reveals that the localization problem may become rather complex: despite a rather constant average node degree (9.40) the number of non–anchor nodes with no anchor neighbor varies from 51 (in TOP2) to 80 (in TOP 8), while the number of non–anchor nodes with 3 or more anchor neighbors varies from 4 (in TOP7) to 15 (in TOP1). For each scenario 30 trials of the selected algorithms were executed, with parameter values summarized in Table II. In the table, P, N, S, D and C denote, respectively, PAES, NSGA–II, SPEA2, MOEA/D–DE and aMOCell4. With the aim of performing a fair comparison, we adopted a population size equal to the archive size.

Once the Pareto front approximation has been generated, a solution must be chosen. In our experiments, we verified that the variation interval of CF for the solutions on the final Pareto front approximation is quite small. Thus, we can assume that each solution on the Pareto front can be acceptable with respect to the CF objective. We have validated this hypothesis with a Wilcoxon test, by selecting from each final archive the solutions characterized by the minimum value of CV and the minimum value of CF (in practice, the solutions on the extremes of the Pareto front approximation). Since no statistical difference exists in terms of NLE among the solutions in the final Pareto front approximations, each solution can be actually selected in order to perform a comparison. For the sake of brevity, we take the solution characterized by the lowest value of CV.

B. Performance analysis

Fig. 2 shows the boxplots of the NLE values achieved by the five algorithms on the different scenarios. The different colors of the boxplots have been used to represent the results of a non-parametric test, namely the Kruskal-Wallis test, applied to the NLE values for each scenario. The Kruskal-Wallis test represents the nonparametric version of the classical one–way ANOVA, and is an extension of the Wilcoxon rank sum test to more than two groups [19]. Briefly, the test compares the medians of the samples, and returns the p–value for the null hypothesis that all samples are drawn from the same population (or equivalently, from different populations with the same distribution). If the p-value is lower than 0.05, we deduce that the null hypothesis does not hold, that is, at least one sample median is significantly different from the others. To determine which sample medians are statistically different, we have applied the built–in Matlab multiple comparison procedure. A white boxplot denotes the best distribution of NLE values obtained for a specific topology. A black boxplot identifies a distribution of NLE values whose median is larger than the median of the corresponding white boxplot with a statistical significance. A gray boxplot denotes a distribution of NLE values whose median is larger than the median of the corresponding white boxplot, but the difference between the medians is not statistically significant.

We can observe that PAES is always able to produce the best NLEs results over the 10 scenarios. Moreover, these results are almost always statistically significant when compared to the other approaches, except for NSGA–II and MOEA/D–DE in TOP1, and aMOCell4 in TOP8 and TOP9, where the multi comparison procedure could not reject the null–hypothesis that the distributions are statistically equivalent with a confidence level of 95%. By the analysis of the distribution of the thirty Pareto fronts for each network topology, we realized that the good behavior of PAES is mainly due to the stability of the algorithm for this specific problem. Indeed, for each topology, the Pareto fronts obtained in the thirty trials are always very close to each other. On the contrary, the other algorithms are less stable.

VI. CONCLUSIONS

In our previous work, the fine–grained localization problem in WSNs had been modeled as a two-objective optimization problem and had been tackled by using a multi-objective evolutionary algorithm, namely PAES. In this paper, we have applied five different state-of-the-art MOEAs in solving the localization problem on different network topologies. All the experimented MOEAs have proved to be able to solve the localization problem with high accuracy, thus confirming the validity of the proposed approach. Further, we have shown that PAES statistically outperforms the other MOEAs in terms of localization error, thus highlighting its particular effectiveness in this specific optimization problem.

REFERENCES

[1] F. Marcelloni and M. Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *The Computer Journal*, vol. 52, no. 8, pp. 969–987, 2009.

[2] F. Marcelloni and M. Vecchio, "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization," *Information Sciences*, vol. 180, no. 10, pp. 1924–1941, 2010.

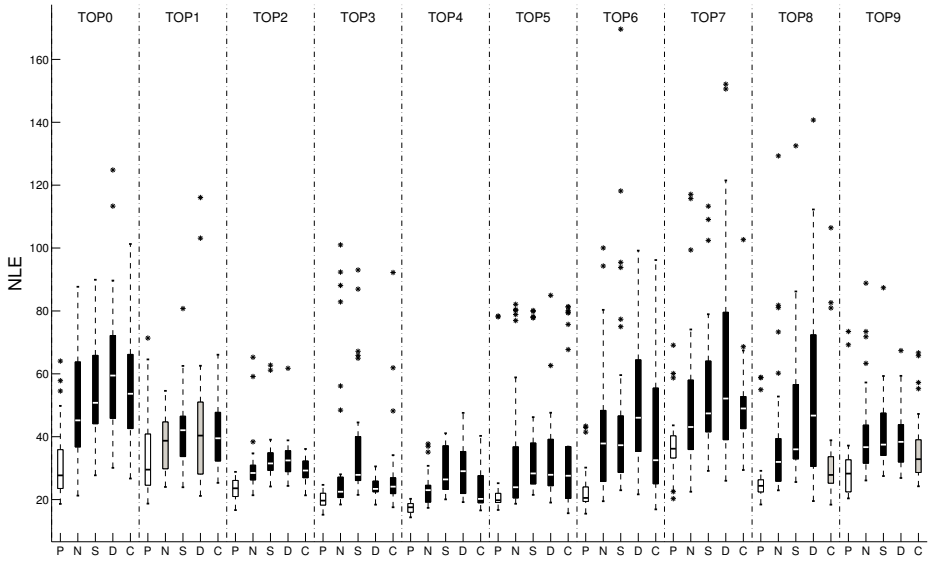

Figure 2. Performance of the five algorithms. P, N, S, D and C denote, respectively, PAES, NSGA–II, SPEA2, MOEA/D–DE and aMOCell4.

[3] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 54–69, 2005.

[4] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.

[5] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based wireless sensor network localization with flip ambiguity mitigation," in *Proc. of the 63-rd IEEE Vehicular Technology Conference*, 2006, pp. 1022–1026.

[6] J. A. Costa, N. Patwari, and A. O. Hero, III, "Distributed weighted–multidimensional scaling for node localization in sensor networks," *ACM Trans. on Sensor Networks*, vol. 2, no. 1, pp. 39–64, 2006.

[7] P. Tseng, "Second–order cone programming relaxation of sensor network localization," *SIAM J. on Optimization*, vol. 18, no. 1, pp. 156–185, 2007.

[8] M. Vecchio, R. López-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, 2011.

[9] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[10] A. A. Kannan, B. Fidan, and G. Mao, "Analysis of flip ambiguities for robust sensor network localization," *IEEE Trans. Vehicular Technology*, vol. 59, no. 4, pp. 2057–2070, 2010.

[11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, K. Giannakoglou *et al.*, Eds. Barcelona, Spain: International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[13] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer, 2005.

[14] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA–II," *IEEE Trans. Evolutionary Computation*, vol. 13, pp. 284–302, 2009.

[15] A. J. Nebro, J. Durillo, J, F. Luna, B. Dorronsoro, and E. Alba, "Design issues in a multiobjective cellular genetic algorithm," ser. Lecture Notes in Computer Science, vol. 4403, 2007, pp. 126–140.

[16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evolutionary Computation*, pp. 712–731, 2007.

[17] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[18] A. J. Nebro, J. Durillo, J, F. Luna, B. Dorronsoro, and E. Alba, "MOCell: a cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, pp. 25–36, 2007.

[19] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. New York, NY, USA: John Wiley & Sons, Inc., 1973.