Challenging Benchmark for Location Discovery in Ad Hoc Networks: Foundations and Applications

Davood Shamsi
Electrical & Computer Engineering
Rice University
Houston, TX, 77005
davood@rice.edu

Farinaz Koushanfar
Electrical & Computer Engineering
Rice University
Houston, TX, 77005
farinaz@rice.edu

Miodrag Potkonjak
Computer Science
University of California,
Los Angeles, CA, 90095
miodrag@cs.ucla.edu

ABSTRACT

We have created the first comprehensive and challenging benchmark data set for the ad-hoc location discovery (LD). The benchmark is a collection of representative real-life distance measurement data that establishes a common basis for understanding, characterization, evaluation and comparison of the LD algorithms and solvers. It is constructed using a novel analysis methodology that systematically establishes the difficulty of discovering the locations. Presence of measurement noise renders the problem difficult even in dense networks. The noise impacts the continuous optimization underlying the LD calculations. We focus on the difficulty of node localization in dense networks. In such networks, the location calculation is viewed as a continuous optimization problem instance with an objective function and a set of constraints. We devise a number of new metrics that evaluate the difficulty of the continuous optimization based on the data set properties. For the LD optimization, a fast simulation methodology is devised for rapid analysis of the sensitivity of the goodness with respect to the data set properties. We present a number of applications for the benchmark data and use it for evaluation and comparison of six popular LD algorithms. The LD benchmarks are publicly available at: http://www.ece.rice.edu/~mm7/benchLD/.

Categories and Subject Descriptors

I.6.7 [**Computing methodologies**]: Simulation and modeling,Simulation support systems; I.6.4 [**Computing methodologies**]: Simulation and modeling, Model validation and analysis; H.1 [**Information systems**]: Models and principles; G.1 [**Mathematics of computing**]: Numerical analysis; E.m [**Data**]: Miscellaneous

General Terms

Algorithms, Experimentation, Performance, and Standardization

Keywords

Add hoc networks, Localization, Benchmarks, and Hard instances, NP-complete problem

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

1. INTRODUCTION

Location discovery (LD) in wireless ad-hoc networks is a widely addressed and important task. Its popularity is well illustrated by continuously increasing number of papers that address LD or closely related topics, such as its impact on other tasks including target detection and tracking. Table 1 shows the total number of LD-related papers and their percentage of WWW deposited papers in Google Scholar database. The 2007 data is, only for the first half of the year.

year	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
LDP	729	978	1,200	1,410	2,090	2,700	3,610	4,110	3,460	1,240
TP	760	779	854	901	933	971	981	874	787	214
R(%)	.096	.125	.140	.157	.224	.278	.368	.470	.440	0.580

Table 1: Popularity of location discovery papers. LDP: number of LD papers; TP: total number of papers in thousands; R: % $\frac{LDP}{TP}$.

LD is an important technical problem because of its direct and profound impact on other tasks such as sensor coverage [1], sensor exposure [2], localized algorithms [3], and sleeping coordination [4]. Therefore, it is not surprising that a great variety of techniques and tools have been developed and evaluated for LD in terms of measurement type (e.g. distance and angle), measured signal modality (e.g., time of arrival, RSS), the underlying objective function (OF) (e.g., L_1 error norm, or consistency), type of beacons and their presence (e.g., stationary or mobile), and algorithms (e.g., iterative solving of linear systems, simulated annealing). In this paper, we focus, with no major loss of generality, on LD data sets where distances between node pairs are measured using audio and RSS signals, and the beacons are stationary.

To enable a proper basis for comparison of the various LD algorithms, a number of researchers have made their distance measurement data publicly available; examples include [5], [6]. However, no generic method or tool for determining the difficulty of modeling a data set free of imposing strong assumptions - such as normality or other closed form distributions of noise is available. As a result, the evaluation criteria for establishing the quality of an LD method or for comparing various methods is not well-justified. Our manual analysis of a set of LD solvers indicated that they perform drastically different on various types of instances that can be characterized in terms of parameters such as the number of nodes and the number of measurements, the percentage of beacons, average error rate, variations of error rate, and the number of dimensions of the embedding space. While on some types of instances numerous LD solvers perform well, on others only a few or one, and sometimes none perform well. Note that, the performance of LD can be

easily measured for the instances where the correct locations of all nodes (ground truth) are readily available, e.g., [7].

The combinatorial complexity (NP-completeness) of LD has been established. However, NP-completeness is a worst-case measure of the difficulty that manifests itself in a few sparse networks. Our evaluations of a number of publicly available LD measurement data sets has shown that the data is almost always gathered from dense networks that are not combinatorially complex. Therefore, to identify LD tools that are practically effective, one has to create instances that are both very challenging and based on actual experimental data.

In order to address this problem and to provide impetus for a more systematic, accurate, and statistically sound comparison of the LD solvers, we develop the first comprehensive *LD benchmark* that consists of a set of real-life instances with well characterized properties. The benchmark construction methodology systematically establishes the difficulty of performing LD with respect to the underlying distance measurement data set and the network topology.

Topology of the network and measurement noise are two major sources of difficulty in finding the locations that were subject of research and study. The significantly correlated and difficult to characterize measurement noise results in complex solution space of the underlying continuous optimization problem, even in dense networks. In this paper, the sources of difficulty are analyzed and employed in LD benchmark development. The reason is that the measured distance data sets that we studied are often not sparse and contain redundancy which helps with minimizing the location estimation error.

The benchmarks can also be applied for other tasks including identification of strong and weak aspects of an LD software, selecting of LD solver for a particular LD instance, tuning of LD software, guidance on which measurements should be conducted in a deployed network, and development of security attacks and defense mechanisms.

Our contributions include

- We devise a method, based on integer linear programming (ILP) that is used for drawing samples with specified structural properties from the real distance measurement data.
- A systematic continuous optimization problem framework for studying the LD problem is demonstrated; the data set is the problem input. The framework is employed in our studies of dense networks with highly variable noise characteristics.
- We introduce a number of new metrics (evaluation criteria) that quantify the difficulty of optimizing an LD instance. The quantification of the difficulty is conducted in terms of nonlinearity and fluctuations of the optimized OF operating on the pertinent input.
- We employ the Placket and Burman (PB) fast simulation methodology that enables rapid analysis and evaluation of multiple properties of the data set with respect to our new metrics. To the best of our knowledge, such fast simulation methods have not been employed in ad-hoc network literature earlier.
- Our new benchmark, metrics, and the evaluation criteria for both sparse and dense networks are publicly available for download and use. The instances are drawn from real measurement data. We also discuss the emerging applications for the LD benchmarks.

The remainder of the paper is organized as follows. After discussing the related work and preliminaries in the next two sections, we present selection of instances with specific properties in Section 4. The challenging instances with respect to OF and continuous optimizations are discussed in Sections 5 and 6 respectively. Section 7 demonstrates using of the benchmark for evaluation and comparison of six popular LD algorithms. In Section 8, we describe a number of important applications for the benchmark instances. We conclude in Section 9.

2. RELATED WORK

As indicated in Table 1, a vast amount of research is dedicated to LD in ad hoc and sensor networks, e.g., [8–11]. Because of space limitations we refer the readers to the surveys [12,13]. Of particular relevance to this work are those that are based on studying and modeling real data and measurement errors [14–16].

Benchmarks are created and used in many fields of electrical and computer engineering and computer science to guide development of new architectures and systems and application software, ranging from embedded systems to information retrieval and databases [17–19]

Organization of experiments and techniques for identifying significant parameters has been a subject of active research for more than 6 decades [20]. Although Plackett-Burman scheme is one of the oldest, it has been serving as the basis for many other schemes and has been used in several fields [21].

Deferent research groups have developed localization tools and constructed distance measurement databases. However, none of them has exploited the complexity of the measurement databases. For example, in Cricket project [5] which is an indoor localization scheme, distance measurement data is available on the web; however, they do not provide a mechanism to construct hard instances of the localization problem.

Open source laboratory in UIUC [6] and CENS laboratory in UCLA [22] also provide databases of the distance measurements without considering the complexity of the implemented networks. A number of authors have used the data in developing their location discovery methods, e.g., the CENS data was used in [14–16]. The measurement data in both data sets is based on the Time Difference of Arrival (TDoA) tested on Mica-2 motes and SH4 nodes, respectively. Because of the deployment environment and measurement tools, distance measurement noise has different characteristics in the different data sets. In Section 7, we show that localization algorithms' performance depends on the underlying measurement data sets.

It is important to note that the localization problem is an NP-complete problem. Saxe [23] proved that embedding a weighted graph in R^d is strongly NP-hard. Also, It was proved that Unit Distance Graph (UDG) model for the network localization leads to NP-hard problem [24]. In UDG model two nodes can measure their mutual distances if and only if their distance is less that a spe-

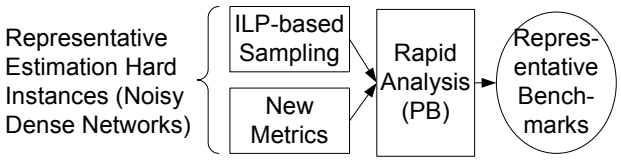

Figure 1: The flow of the LD benchmarking methodology.

cific range. NP-completeness is the worst-case complexity of the problem that only manifests itself on sparse networks. It is well-known that not all instances of an NP-complete problem are hard to solve [25]. The available measurement data that we consider are not sparse and thus, the worse-case complexity is not relevant.

3. PRELIMINARIES

Our target problem can be formally defined as follows.

Problem: IDENTIFYING HARD LD INSTANCES

Instance: Nodes denoted by v_n ($n = 1, \ldots, N$) placed in $\mathbb{R}^d$ ($d = 2, 3$); $B(> d)$ of the nodes are beacons $n_1, \ldots, n_B$ that know their locations; the mutual distance $l(v_n, v_m)$ measured between pairs of nodes v_n and v_m, $(m, n = 1, \ldots, N)$; measurement noise $\epsilon_{n,m}$, where $l(v_n, v_m) = d(v_n, v_m) + \epsilon_{n,m} - d(v_n, v_m)$ denotes the real distance. *Neighbors* are nodes that can measure their mutual distance.

The goal of the LD is to find the location of the non-beacon nodes. LD is often treated as an optimization problem. For example, the OF may be minimizing the difference between the measured and the final distances, while the constraints are formed by the available distances.

Objective: We target the problem of determining the difficulty of addressing a particular LD data set (instance) by different algorithms and methods. The difficulty is expressed as a function of various properties, such as the measurement error and the density of the graph.

A weighted graph $G(V, E)$ can be defined for LD such that $V = \{v_1, v_2, \cdots, v_N\}$ and $\{v_n, v_m\} \in E \iff \{v_n, v_m$ are neighbors$\}$. The weight of each edge is equal to the mutual distance between its two adjoining nodes. *Degree* of a node is the number of edges incident to it.

Realization of a weighted graph, $G(V, E)$, refers to assigning coordinates to the nodes. At it is expected in a realization of a weighted graph, distance between the two nodes might not be equal to the weighted edge between two nodes in the corresponding weighted graph.

Figure 1 shows the flow of our approach. It shows the process for drawing samples where the estimation of the final location is hard even in dense networks (Sections 5 and 6). The process consists of a new ILP-based sampling method (Section 4), introducing novel metrics (Subsection 5.1), and a fast analysis method (Subsection 5.2).

4. SAMPLING INSTANCES WITH SPECIFIED PROPERTIES

The starting point for the instance identification is to have an LD data set with specific properties. Our approach advocates the use of real data. Thus, instance determination translates to sampling the measurement data such that the sample follows the desired specifications. Many properties specify an LD instance, including the graph properties (e.g., number of nodes or edges), topological properties (e.g., number of dimensions), and technological properties (e.g., measurement noise).

Let S denote a sampled instance from the data set (distance measurements). In this paper, we specify and experiment on the following properties for S:

1. Number of nodes shown by N_S.
2. Number of beacons denoted by B_S.
3. Mean square distance measurement noise shown by $\overline{\epsilon_S^2}$.
4. Percentage of edges with square noise higher (or lower) than a given value ϵ_0^2, $PER_S^{\epsilon_0^2}$.
5. Mean degree of the graph denoted by $\overline{D_S}$.
6. Minimum and maximum edge length presented by MINL$_S$ and MAXL$_S$ respectively.
7. Mean edge length shown by $\overline{l_S}$.
8. Minimum degree of the non-beacon nodes, MIND$_S$.
9. Maximum squared error, MAXϵ_S^2.

Many other properties could also be considered, including the mean absolute error of the noise, longest distance in the network, and the area covered by the network. We believe that the nine properties specified above provide a good representation for the instances studied in the literature. We also emphasize that our instance sampling method can be applied for drawing samples with arbitrary desired properties.

We adopt the following problem formulation, for a given data set that contains a LD graph with the correct coordinates of the nodes, and a number of noisy distance measurements.

Problem: SPECIFIED SAMPLING OF LD INSTANCES

Instance: A *full* LD data set with N nodes with correct coordinates $\forall v_n$, $n = 1, \ldots, N$ and noisy distance measurements between a subset of node pairs, constants N_S, B_S, $\overline{\epsilon_S^2}$, $PER_S^{\epsilon_0^2}$, $\overline{D_S}$, MINL$_S$, MAXL$_S$, $\overline{l_S}$, MIND$_S$, and MAXϵ_S^2.

Question: Is there a sub-sample S of the full data set, with the properties defined above for the instance?

We now formulate the data set sampling problem as an instance of an integer linear program (ILP). There are at least two reasons for using the ILP and not to use heuristic algorithms for this problem. First, even though the complexity of ILP for solving instances would increase for large data (e.g., more than 500 nodes), for our

scenarios of networks with less than 100 nodes we are able to find the best solutions quickly on a standard PC. Second, the ILP's underlying optimization mechanism is insensitive to the various property interactions. It is very hard (if not impossible) to build a heuristic method that is as insensitive as ILP. We solve our ILP using the commercial CPLEX package [26]. We first introduce the constants and variables, and then formulate the OF and constraints.

Given: An LD data set containing a graph $G(V;E)$ $|V|$=N; the real coordinates (x_n, y_n, z_n) for each node v_n n=$1,\ldots,N$; the matrix $E_{N\times N}$ with elements $e_{n,m}$ that are either zero or one corresponding to the existence of distances measurements (edges) between some node pairs $v_n \in G$ and $v_m \in G$, also, there are K matrices $L^k_{N\times N}$, $k = 1,\ldots,K$ that determine the K different noisy distance measurements $l^k_{n,m}$ for each edge in E $(m,n$=$1,\ldots,N)$; constants N_S, B_S, $\overline{\epsilon^2_S}$, $PER^{\epsilon^2_0}_S$, $\overline{D_S}$, MINL$_S$, MAXL$_S$, $\overline{l_S}$, MIND$_S$, and MAXϵ^2_S.

Variables: Two vectors Φ_S and Ψ_S with elements ϕ_n and ψ_n, $n = 1,\ldots,N$ such that,

$$\phi_n = \begin{cases} 1, & \text{If the node } v_n \text{ is selected in } S \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\psi_n = \begin{cases} 1, & \text{If the node } v_n \text{ is a beacon in } S \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

a matrix Ξ_S with elements $\xi_{n,m}$, $n,m = 1,\ldots,N$ such that,

$$\xi_{n,m} = \begin{cases} 1, & \text{If edge } e_{n,m} \text{ is selected in } S \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

a three dimensional matrix Υ_S with elements $\upsilon_{n,m,k}$,

$$\upsilon_{n,m,k} = \begin{cases} 1, & \text{If the } k\text{-th measurement is used} \\ & \text{for edge } e_{n,m} \text{ in } S \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Objective Function (OF): We define the objective function such that it maximizes the distance between the beacons so that they are spread in the ntwork. Given the location of all nodes, the matrix Γ with elements $\gamma_{n,m}$ corresponding to the distance between two nodes v_m and v_n, n,m=$1,\ldots,N$ can be computed. We maximize following objective function.

$$OF: \qquad \sum_n \sum_m \gamma_{n,m}\psi_n \wedge \psi_m.$$

Implementing "and" ($\wedge$) with linear operators is explained later.

Constraints: The problem has 12 sets of constraints. The first set of constraints (C_1) ensures for each selected edge, both its accompanying nodes are also selected,

$$C_1: \ \forall m,n,\ \xi_{n,m} \rightarrow (\phi_n \wedge \phi_m) \equiv (\sim \xi_{n,m} \vee (\phi_n \wedge \phi_m)) \quad (5)$$

To express this in linear terms, for two binary variables A and B we define two auxiliary operations $F_{OR} = A \vee B$ and $F_{AND} = A \wedge B$ as follows.

- For OR operator
  1. $F_{OR} \geq A$ and $F_{OR} \geq B$
  2. $F_{OR} \leq A + B$
  3. $0 \leq F_{OR} \leq 1$
- For AND operator
  1. $F_{AND} \leq A$ and $F_{AND} \leq B$
  2. $1 + F_{AND} \geq A + B$
  3. $0 \leq F_{AND} \leq 1$.

Generalization to more variables is straightforward. Using the two new auxiliary operators, we can now define C_1 in linear terms.

Also, we need a second constraint set denoted by C_2 that ensures if a node is in S ($\phi_n = 1$), it has at least one adjacent edge in $\xi_{n,m}$, $m = 1,\ldots,N$. Also, the degree of non-beacon nodes should be at least MIND$_S$, i.e.,

$$C_2: \quad \forall n, \sum_m \xi_{n,m} \wedge \phi_n \geq \text{MIND}_S\phi_n - (\text{MIND}_S - 1)\psi_n$$

The constraints C_3 and C_4 define the number of nodes and the number of beacons. For $n,m = 1,\ldots,N$,

$$C_3: \ \sum_n \phi_n = N_S \quad (6)$$

$$C_4: \ \sum_n \psi_n = B_S$$

Note that $\sum_n \phi_n$ includes all nodes (beacon and non-beacon nodes). Thus, each beacon should first be selected as a node,

$$C_5: \quad \forall n,\ \psi_n \rightarrow \phi_n \equiv \sim \psi_n \vee \phi_n$$

To implement the constraint on the mean square error of measured distance (denoted by C_6) we use the inputs to form a matrix with elements $\epsilon^2_{n,m,k}$ that are the square of difference between the correct node locations and the distance for each edge $e_{n,m}$ in the k-th measurement. Also, we consider a margin δ_ϵ around the given ϵ^2_S. Thus, for $m,n = 1,\ldots,N$,

$$C_6: \quad \epsilon^2_S - \delta_\epsilon \leq \sum_n \sum_{m>n} \sum_k \upsilon_{n,m,k}\epsilon^2_{n,m,k} \leq \epsilon^2_S + \delta_\epsilon$$

The next constraint ensures that just one of the K measurements is selected for each edge in the sampled instance:

$$C_7: \ \sum_k \upsilon_{n,m,k} = \xi_{m,n}$$

Similarly, the other sets of constraints can be expressed:

$$C_8: \quad \frac{(\overline{D_S} - \delta_D)N_S}{2} \leq \sum_n \sum_{m>n} \xi_{n,m} \leq \frac{(\overline{D_S} + \delta_D)N_S}{2}$$

$$C_9: \quad \sum_n \sum_{m>n} \sum_k [\upsilon_{n,m,k} \wedge (\epsilon^2_{n,m,k} \geq \epsilon^2_0)] \leq PER^{\epsilon^2_0}_S.E_S$$

$$C_{10}: \quad \forall n,m,k, \text{MINL}_S\upsilon_{n,m,k} \leq \upsilon_{m,n,k}l^k_{n,m} \leq \text{MAXL}_S$$

$$C_{11}: \quad (\overline{l_S} - \delta_l)E_S \leq \sum_n \sum_{m>n} \sum_k \upsilon_{n,m,k}l^k_{n,m} \leq (\overline{l_S} + \delta_l)E_S$$

$$C_{12}: \quad \forall n,m: \upsilon_{n,m,k}\epsilon^2_{n,m,k} \leq \text{MAX}\epsilon^2_S$$

where ϵ_0^2 denotes the predefined high threshold values for square noise; $E_S = \frac{\overline{D_S} N_S}{2}$ is the total number of edges; and δ_D and δ_l are positive margin values for the average graph degree and edge length.

5. CREATION OF INSTANCES WITH DIFFICULT TO FIND OF

In this section, we address the issues related to the OF used for LD. Given a set of location data, a fundamental question is: what is the best OF that can be used for this data set? Alternatively, there is a need to define the performance of a given OF on an LD data set. In the Subsection 5.1, we introduce three methods that will be used as metrics for evaluation of the OF. In Subsection 5.2, we utilize the metrics and introduce a systematic fast method for studying the sensitivity of the OF on the various LD instance parameters.

5.1 Metrics for evaluating OF

Given the measurement data and the original nodes' locations, we introduce three metrics that will be used for evaluating the OF on the LD data. Perhaps the most straightforward way to evaluate an OF on LD data set would be to integrate the OF in an optimization engine and then check the optimized results. If the optimal algorithm is available, then it would yield the correct locations. However, since LD is a complex problem, one cannot hope to have the optimal algorithm. Only heuristics can be developed. Thus, the quality of the final solution is not just a function of the used OF, but also a consequence of the underlying heuristics.

We introduce evaluation metrics that are independent of the optimization engine. Our method for finding evaluation metrics utilizes the fact the LD data sets contain the correct locations.

1. *Rank-based consistency of the OF.* There are three steps that should be taken for finding the consistency. In the first step, we simply substitute the correct location as the solution and find the corresponding value of the OF. The second step consists of repeatedly generating random positions in the interval (0,ϵ) around the correct location of all the nodes in the network and then, finding the value of the OF for the new nodes' positions. The iteration in the second step is itself repeated K times, each time increasing the interval to (0,$k\epsilon$), $k = 1, \ldots, K$. In the third step, we model the resulting values of the OF versus the corresponding average distance between the generated points and the correct locations for each location configurations.

   In Figure 2, we plot OF versus the distance to the correct solution on a small example. The plot quantifies the changes in the OF value by altering the LD estimate using any algorithm that randomly generates a solution in the (0,$k\epsilon$) interval. The consistency of the OF is a metric defined using the plot. The rank-based consistency check is as follows. For an LD data set with N nodes, let OF(X_1) and OF(X_2) denote the values of the OF evaluated for the $2N$-dimensional points $X_1 \in \mathbb{R}^{2N}$ and $X_2 \in \mathbb{R}^{2N}$. Each $2N$-dimensional point corresponds to the x and y coordinates for the N nodes. Let X_c denotes the correct location of the points, and $d(X_c, X_1)$ as well as $d(X_c, X_2)$ denote the distance between X_c and X_1, and X_c and X_2 respectively. If for two pairs $(X_1, OF(X_1))$ and $(X_2, OF(X_2))$ we have $d(X_c, X_1) \geq d(X_c, X_2)$ and OF(X_1) $\geq$ OF(X_2), then the objective function is consistent for the two pairs of points.

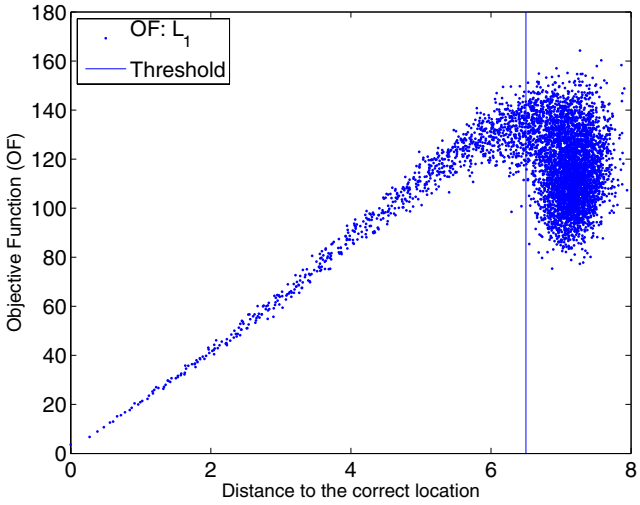

Figure 2: An example plot showing the changes in an objective function (OF) versus the distance of the solution to the correct one.

We count the number of times the node pair comparisons are consistent and use this number as a metric for the OF evaluation.

2. *Variance of the OF.* Another quantitative measure that can be found for the plot in Figure 2 is the variance of the OF values that is incurred by increasing the distance of the estimated location to the correct location. There is a transition point after which the OF estimation essentially becomes random. For example, in Figure 2 we see increase in the variance of the OF for points with a distance larger than 6.5ϵ. The transition point is illustrated by the vertical line on the. The metric is to check the variance of the model before the transition point. The variance shows how well the OF captures the changes in the distance of the final solution to the correct location. A high variance indicates that the OF is not a good metric for the underlying data set, while a low variance indicates sensitivity to the changes in the LD data.

3. *Drifting of the OF.* This metric quantifies the effect of the OF under evaluation in drifting from the correct location. Let $\text{OF}_M : \mathbb{R}^{dN} \rightarrow \mathbb{R}^+$ be the objective function mapping from the dN-dimensional space to one dimension. M is the measurement set. We assume $\text{OF}_M \in \mathcal{C}^0$ (i.e., OF_M is continuous). Denote the noiseless measurements by M_c. The global minimum of OF_{M_c} is at X_c; where X_c is the $2N$-dimensional point corresponding to the correct nodes' positions. Because of the measurement noise, X_c is not necessarily the global minimum of OF_{M_n} (OF with noisy measurements). This means that if we start from the correct point X_c and move in the opposite direction of the gradient, we will be trapped in a local minimum, denoted by X_l. This local minimum may not be the global minimum of the objective function OF_{M_n}. The distance between X_l and X_c is denoted by d_m which determines the drifting caused by the OF with noisy measurements, OF_{M_n}.

5.2 PB design for fast sensitivity analysis

Our statistical method is based on the foldover Plackett and Burman (PB) multifactorial experiment design [20] to determine the effect of various instance properties on the OF, to compare the effectiveness of different OFs, and to distinguish OFs based on their effect on the instance properties. The three metrics defined earlier will be used for evaluating the OFs.

The major problem in evaluation of the OF versus the instance properties is the number of the underlying properties. To have a thorough analysis, one would need to run many experiments. In each experiment run, there are a number of key parameters that must be selected, including the property values (e.g., number of nodes, number of beacons), the non-varied properties and their values, the varied properties and their range of variations, dependent parameters. Another key question is the number of experiments that would yield a certain confidence over the results. Evaluating all possible values of all properties is not feasible.

This paper proposes a formal method for improving the experiment set-up and sensitivity analysis. Our approach is based on a very fast and efficient known statistical method. To the best of our knowledge, such a method has never been used before for simulations in sensor networks where only ad-hoc evaluation methods have been applied. The methodology not only speeds-up the evaluation process, but also makes the results more exact by providing a statistical confidence. We use the method for both evaluating the sensitivity of the OF in this section, as well as evaluating the continuous optimization approaches in the next section.

If we denote the number of properties by N_p, PB multifactorial experiment design is a *saturated design* method that has a linear number of experiment runs with respect to N_p. A saturated design is where the information in all the experiments is used to estimate the sensitivity. Since a number of selected instance properties might have interactions, we opt to use a foldover PB design that uses twice the number of runs of the original PB while including a linear number of property interactions [21]. A foldover PB runs twice number of runs compared to the original PB design.

The values of the properties selected for each experiment run is defined by the original PB design matrix. Each row in the design matrix corresponds to one experiment, while each column corresponds to one of the N_p instance properties. The entries in the table are either +1 or -1. For each property in the experiment, the PB either sets a higher value (+1) compared to the properties' normal range or a lower value (-1) than the normal range. The selected low and high values remain constants for each property over all of the experiment runs. In cases where N_p is smaller than the number of rows denoted by N_r, the first column is taken from the original PB design matrix for N_r which has $N_r - 1$ components [20]. The subsequent columns are just each shifted circularly down by one compared to their previous columns. The last row of the matrix is filled with -1 in all its corresponding columns. The foldover is constructed from the PB design, by copying the rows and then just flipping the -1 and +1 signs.

In Table 2, we illustrate the original PB design table that we have used in our experiments with 10 parameters, $N_p = 10$. Columns 2-11 show the instance properties, while the last column illustrates the value of the evaluation metric for those properties. For an original PB design, the number of rows would be 12 and for the foldover PB, $N_r = 24$. Because of space limitations, we do not show the rows 13-24 since that they are the mirrored-sign version of the first 12 columns.

Run	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Metric
1	+1	-1	+1	-1	-1	-1	+1	+1	+1	-1	22
2	+1	+1	-1	+1	-1	-1	-1	+1	+1	+1	37
3	-1	+1	+1	-1	+1	-1	-1	-1	+1	+1	15
4	+1	-1	+1	+1	-1	+1	-1	-1	-1	+1	38
5	+1	+1	-1	+1	+1	-1	+1	-1	-1	-1	24
6	+1	+1	+1	-1	+1	+1	-1	+1	-1	-1	36
7	-1	+1	+1	+1	-1	+1	+1	-1	+1	-1	29
8	-1	-1	+1	+1	+1	-1	+1	+1	-1	+1	39
9	-1	-1	-1	+1	+1	+1	-1	+1	+1	-1	20
10	+1	-1	-1	-1	+1	+1	+1	-1	+1	+1	36
11	-1	+1	-1	-1	-1	+1	+1	+1	-1	+1	25
12	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	45

Table 2: The original Plackett and Burman (PB) experiment design table for $N_p = 10$.

Selection of the low and high value for the range of each property is also important. The high and low values should not be extreme, since that would exaggerate the effect of each property [21]. The best is to select high and low values to be on the boundary with the normal range of the values.

Finally, the results of the PB experiment design is used for identifying the effect of each of the properties on the OF. Assume that we use one of the metrics defined for evaluating the OF. For each experiment run, there will be one value for the metric. An example is shown in Table 2, where the last column shows the value for one metric. The sensitivity to each parameter is calculated by multiplying the metric in the PB parameter value (+1 or -1) and summing up through the columns. For example, for an original PB (without foldover) with 10 parameters and the metric shown on the last column, the effect of the metric on P_1 is: $(+1.22)$+ $(+1.37)$+ (-1.15)+ $(+1.38)$+ $(+1.24)$+ $(+1.36)$+ (-1.29)+(-1.39)+ (-1.20)+ $(+1.36)$+ (-1.25)+ (-1.45)= 20. Only the magnitude of the sensitivity (not its sign) is important.

The results of applying the PB design for LD are shown in Table 3. The nine parameters that are studied here were introduced in Section 4. Three objective functions were evaluated: L_1, L_2 and Maximum Likelihood (ML) (for exact definition please see [14]). In ML, the likelihood function of the measurement noise ℓ is maximized. ML objective function is obtained by modeling the distance measurements noise. The three metrics were used to evaluate OFs: consistency, variance and drifting. The elements of the table represent the rank of the parameters for the specific OFs and the evaluation metrics. Number 1 means the most important parameter. Though parameter ranks are not the same in all columns, this table illustrates that the number of nodes (N_S), average degree ($\overline{D_S}$), number of beacons (B_S), and mean square error ϵ_S^2 are the most important parameters. Moreover, it shows that the minimum and maximum edge lengths MINL_S and MAXL_S, do not have large impacts on the OFs.

Parameter	Consistency			Variance			Drifting		
	L_1	L_2	ML	L_1	L_2	ML	L_1	L_2	ML
N_S	1	1	1	4	2	2	4	4	2
B_S	5	3	3	2	3	3	2	1	1
$\overline{\epsilon_S^2}$	7	6	8	6	7	8	1	2	3
$\text{MAX}_{\epsilon_m^2}$	6	5	9	9	10	6	6	8	7
$PER_{\epsilon_0^2}$	8	7	5	8	6	9	7	9	10
$\overline{D_S}$	2	9	2	1	1	1	3	3	4
MINL_S	10	2	6	10	9	10	8	6	8
MAXL_S	4	10	10	5	5	5	10	10	9
$\overline{l_S}$	3	8	4	3	4	4	9	5	5
MIND_S	9	4	7	7	8	7	5	7	6

Table 3: Importance of parameters for various OFs and metrics. Generally, the number of nodes (N_S), average degree ($\overline{D_S}$), number of beacons (B_S), and mean square error $\overline{\epsilon_S^2}$ are the most important parameters.

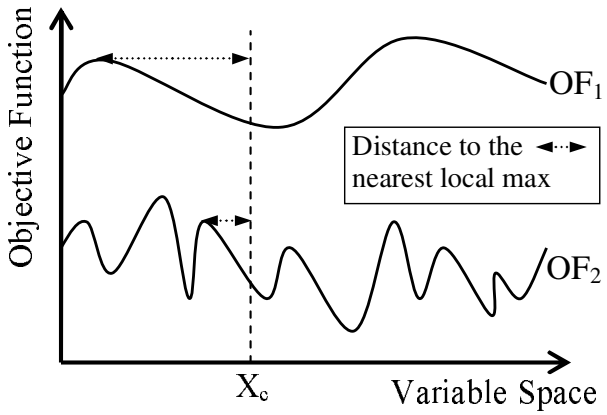

Figure 3: A smooth (OF_1) and a bumpy (OF_2) OF over the 1D variable space; X_c is the correct location.

Param	L_1	L_2	ML	EC	Param	L_1	L_2	ML	EC
N_S	6	5	6	1	$PER_{\epsilon_0^2}$	10	8	5	6
$\overline{\epsilon_S^2}$	2	3	4	4	$\text{MAX}_{\epsilon_m^2}$	9	10	10	10
B_S	4	2	3	7	MINL_S	7	6	9	9
$\overline{D_S}$	1	1	1	3	MAXL_S	5	9	8	2
$\overline{l_S}$	8	7	7	5	MIND_S	3	4	2	8

Table 4: Importance of parameters for OF distances to the local maximum. The mean degree of the graph is the most important parameter and $\text{MAX}_{\epsilon_m^2}$ is the least important parameter.

6. CREATION OF INSTANCES THAT ARE HARD FOR CONTINUOUS OPTIMIZATION

In this section, we evaluate the impact of the optimization method on the LD instances. The selection of the optimization algorithm is closely related to the underlying OF. The three metrics introduced for evaluation of OF do not measure its interactions with the optimization method. To get more insight to this interaction, we have performed intensive set of various optimizations for the OFs under study. An important observation was that in some cases, an OF that had a high quality in terms of the three introduced metrics (consistency, variance and drifting) did not yield a high quality solution. In parallel with a good OF, optimization mechanisms that can achieve a good LD result are also needed.

A key question to answer is which shape of the OF complicates the optimization algorithm. Location discovery is a nonlinear optimization problem in the $\mathbb{R}^{dN}$ space of its variables. It is well known that nonlinear optimization problems with a large number of local minima are hard to solve. This is so because the space of the underlying variables is huge and thus an exhaustive search is out of question. Nonlinear optimization methods search subparts of the space to find the best global minimum of an OF. If the OF is smooth, the search is easy. However, if the hypersurface of OF in the space of variables is bumpy (a lot of hills and valleys) the optimization algorithm would have a hard time to find the global minima, unless it searches the space for a long time. Figure 3 illustrates a simplified example with two nonlinear objective functions in the $\mathbb{R}$ space. From what we have just described, it is easy to see that it is much harder to find the global minimum of the OF_2 when compared to OF_1.

Nearest directional local maximum. As shown in Figure 3, distance to the nearest local maxima is a good indicator for the OF bumpiness. Because of curse of dimensionality of OF in $\mathbb{R}^{dN}$, it is not feasible to study this property for all dimensions simultaneously. Therefore, we study the property for changing each node separately and then combine the results.

For each node v_n, we select a random direction $\vec{a} \in \mathbb{R}^2$. All nodes are placed in their correct positions. We start from the correct location of v_n denoted by u and move in the direction $\vec{a}$, i.e.,

$$u'(t) = u + t\vec{a}, t > 0.$$

Then, we find the local maximum correspond to the smallest t, $u'(t_n)$. The local maximum indicates how far one can go from the correct point before hitting a bump. We introduce g as a metric for the bumpiness of OF.

$$g_n = \frac{1}{t_n}.$$

Small g_n indicates one can go far from the correct location since the OF is smooth. The overall behavior of the OF is found by averaging over g_n.

We conduct the PB-design to evaluate the effect of different parameters on the bumpiness of the OFs. Table 4 is representing the rank of each parameter in making OF bumpy. In addition to the three objective functions L_1, L_2, and ML, Edge Consistency (EC) is also studied [14]. EC (Similar to the previous section) measures the inconsistency between measurement edges and edges length in the current realization of the graph.

While the results are not drastically different from the previous section, they illustrate that the mean degree of the graph is the most important parameter. Moreover, maximum squared error $\text{MAX}_{\epsilon_m^2}$, mean edge length $\overline{l_S}$, and minimum edge length MINL_S have the least effect on the OF bumpiness.

In Figure 4, Mean Distances to the nearest directional local Maximum (MDM) $Mean(t_n)$ is plotted for different OFs. Having a large MDM indicates that the OF is not bumpy. This figure sug-

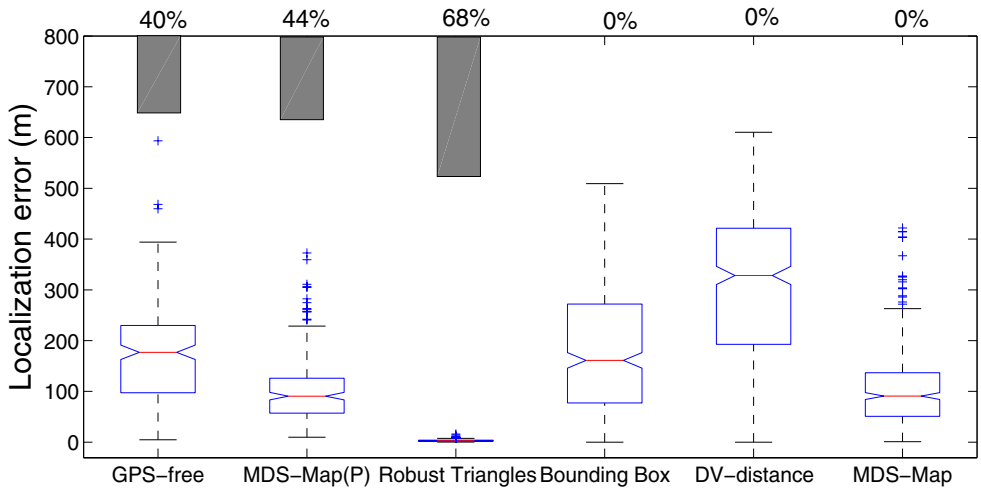

Figure 5: Localization error for CENS data set. Bar plot on the top of the figure shows percent of the nodes that are not localized. The first three algorithms could not localized all the nodes.

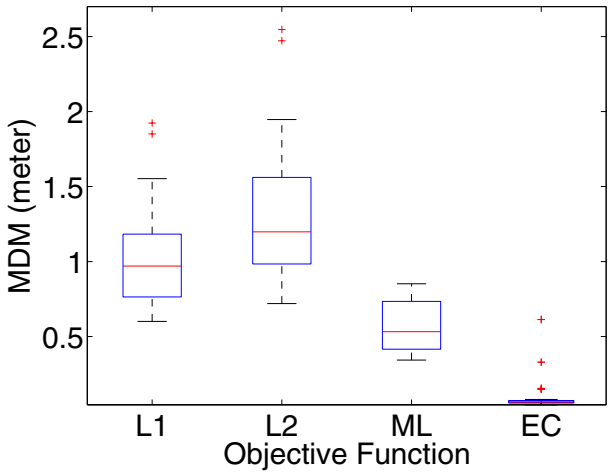

Figure 4: Boxplot of the mean directional distance to the nearest local maximum for various OFs. MDM: Mean Distances to the nearest directional local Maximum.

gests that EC is highly bumpy around the correct solution. Thus, finding the global minimum for EC is hard for heuristic non-linear optimization methods. Also, ML has a relatively low MDM even though it is analytically the best OF (i.e., yielding the solution with the least error) if the correct noise model is known. However, it is bumpy compared to L_1 and L_2 OFs. Thus, it is harder to find optimization methods that find its global minimum.

7. BENCHMARK EVALUATION

The benchmark set introduced in this paper is evaluated by 6 different LD algorithms. We evaluate benchmarks from PB-design under these localization algorithms. Silhouette simulator [27] is used for LD. We have used following six algorithms to evaluate the benchmarks:

- GPS-Free [28].
- MDS-Map [29].
- MDS-Map (P): MDS-Map based on patches of local maps [30].
- Robust Triangle [28].
- Bounding Box [31].
- DV-distance [32].

Figures 5 and 6 compare localization error for six localization algorithms. The percentage of non-localizable nodes are shown by numbers and barplots at the top of the figures. Figures 5 and 6 are for Hard10 benchmarks drawn from the CENS measurement data [22] and ultrasound [27] measurements, respectively, that are available online. Here again, networks are scaled to fit in a $1000m \times 1000m$ square area. The plots indicate that solvers behave differently for various network instances, e.g., Robust Triangles performs bad on benchmarks drawn from CENS measurements (Figure 5) but it performs reasonable for benchmarks based on the ultrasound measurements (Figure 6). As a result, characteristics of the measurement noise dramatically affect the performance of the localization algorithm.

8. APPLICATIONS

A broad variety of technical applications can directly benefit from challenging LD benchmark (LDB). They include evaluation and improvement of a particular LD solver, identifying classes of LD problems that can be easily solved using the current technology, and LD security. In this section, we briefly discuss a sample of applications that can leverage LD challenging instances in our benchmark. The benchmark is available for non-commercial public use at: http://www.ece.rice.edu/~mm7/benchLD/. In the web page, we have determined samples of hard LD problems with different properties.

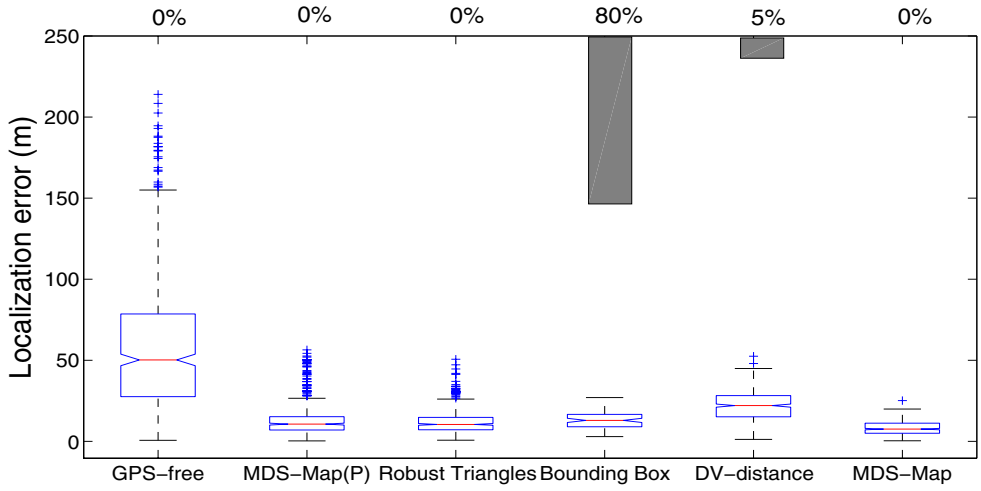

Figure 6: Localization error for ultrasound data set. Bar plot on the top of the figure shows percent of the nodes that are not localized. Bounding Box algorithm localizes just 20% of the nodes.

1. *Evaluation and characterization of a particular LD software package.* The LDB can be used to identify the weaknesses and potentials of a solver as well as the directions for its improvement. It can find when a solver is applicable with acceptable accuracy. For example, LDB can diagnose the inefficiency of the underlying OF or optimization algorithms and suggest alternations. For instance, our LDB evaluation indicates that conjugate gradient maximum likelihood LD solver [14] does not perform well on sparse instances (average degree < 3) with small percentage of beacons (the average relative L_1 error was 38% on LD instances of this type).

2. *Selecting LD software for a pertinent LD instance.* Our comprehensive LD instance analysis indicates that for several instance types, majority of the evaluated LD tools are performing well, with relatively small differences in the quality of the solutions. However, for different types of difficult instances, a few LD solvers perform significantly better than the others. For example, for sparse instance with average graph degree below 4, our new convex programming-based LD solver outperformed any other considered solver by 40%. On the other hand, conjugate gradient consistency solver CGCS with difficulty driven partitioning [15] had superior performances on large and high density instances.

3. *Parameter selection for LD software.* A significant percentage of LD solvers have either explicit or implicit mechanisms for exploring the *quality of solution-runtime* tradeoff or for emphasizing different optimization mechanisms. For instance, often LD solvers employ discrete or continuous iterative improvement: they either iteratively visit sets of nodes that are optimized using atomic multilaterations [8] or employ different gradient descent directions [14]. In addition, it is often straightforward to replace one OF with another [15, 16]. By providing a set of difficult instances, we facilitate selection of the parameters in such a way that difficult instance can be solved well or in short time. Also, it can provide a guidance for identifying the termination time of the nonlinear algorithm.

4. *Instance Partitioning.* There is sometimes a need to partition an LD instance so that nodes in each part can be separately or sequentially located. There are three main reasons for LD partitioning. The first is to minimize the communication needs of localized LD algorithms. The second is scaling: many of the contemporary good solvers take prohibitively large running times on larger instances. For example, instances with thousand nodes approximately require around 1 day PC runtime to generate high quality solutions [14]. Finally, as a consequence of scaling, it has been demonstrated that partitioning often has as one of its ramifications creation of better LD solutions: while after partitioning a part of the available measurements and beacon locations information is lost, the smaller network parts with specific structures are easier to address.

   LDB helps LD instance partitioning by indicating the size and the structure of instances on which one of the available solvers performs well. For example, we were able to suppress the error by a factor of 3, by partitioning a network of 1000 nodes into subparts of 100 nodes, for relative L_1 OF [14].

5. *LD Knowledge.* LDB can be used for gaining numerous insights about LD in sensor networks.

   5.1. Identification of solved/unsolved types of instances. One can identify what type of instances are solvable in a sense that there is an available LD solver that addresses them with a specified accuracy. Also, instances that are currently not addressed properly can be identified. In general, it was difficult to solve instances that had average graph degrees below 4, instances with large number of nodes and a few beacons, instances with high average error and low error variance, and 3D instances. Probably the single most important conclusion of our LDB evaluations is that much more often the cause for low quality LD is inadequate OF and not inefficiency of optimization mechanisms. Thus, we advocate that LD research focuses on derivation of better OFs that more accurately grasp the essence of the measurements error characteristics.

   5.2. The number of distance measurements for a targeted accuracy. One important LDB benefit is that it can be used to

deduce the required number of measurements for a given sensor network, LD solver, and targeted LD accuracy. Another important LDB benefit is that it can guide us in figuring out which measurements (between which pair of nodes) is the most beneficial to conduct so that the resulting LD instance can be best solved for a given network and a given LD solver.

Equally importantly, LDB can be used for deciding where to place beacons either for a given network or for a given area where mobile nodes should be supported for calculating their current positions or tracking trajectory. Finally, LDB can be used to organize the most effective security attacks that prevent accurate measurements or alter measurements between pairs of nodes. LDB can identify which network structures and topologies, measurement scenarios, and LD tools are resilient against a specified number of measurement alternations.

9. CONCLUSION

We have developed a challenging location discovery (LD) benchmark. The benchmark consists of instances that are not just difficult to be accurately solved by majority of the currently available LD solvers, but also will remain challenging for yet to be developed solvers. We identify difficult instances by conceptually and statistically analyzing the relationship between LD objective functions and discrete and continuous optimization mechanisms of existing and potential LD solvers. The benchmark includes actually deployed LD measurement data and can be used not only for quantitative comparison of LD solvers but also for many other purposes ranging from diagnosis of strong and week aspects of a specific LD solver to addressing LD-related security attacks.

10. REFERENCES

[1] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Worst- and best-case coverage in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 84–92, 2005.

[2] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in wireless sensor networks: Theory and practical solutions," *ACM Journal of Wireless Networks*, vol. 8, no. 5, pp. 443–454, 2002.

[3] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure," in *MobiHoc*, 2001, pp. 106–116.

[4] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *Infocom*, 2006, pp. 1–13.

[5] "http://cricket.csail.mit.edu, seen in nov. 2007."

[6] "http://www-osl.cs.uiuc.edu/research?action=topic&topic=sensor+networks, seen in nov. 2007."

[7] L. Girod, M. Lukac, V. Trifa, and D. Estrin, "The design and implementation of a self-calibrating distributed acoustic sensing platform," in *Sensys*, 2006, pp. 71–84.

[8] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobicom*, 2001, pp. 166–179.

[9] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX*, 2002, pp. 317–327.

[10] F. Koushanfar, S. Slijepcevic, M. Potkonjak, and A. Sangiovanni-Vincentelli, *Location Discovery in Ad-hoc Wireless Sensor Networks*. Kluwer, 2003.

[11] N. Kiyavash and F. Koushanfar, "Anti-collusion position estimation in wireless sensor networks," in *IEEE Internatonal conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007, pp. 1–9.

[12] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[13] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Mobicom*, 2004, pp. 45–57.

[14] J. Feng, L. Girod, and M. Potkonjak, "Location discovery using data-driven statistical error modeling," in *Infocom*, 2006, pp. 1–14.

[15] J. Feng and M. Potkonjak, "Consistency error modeling-based localization in sensor networks," in *SECON*, 2006, pp. 356–364.

[16] J. Feng, L. Girod, and M. Potkonjak, "Consistency-based on-line localization in sensor networks," in *DCOSS*, 2006, pp. 529–545.

[17] M. Carey, D. DeWitt, and J. Naughton, "The oo7 benchmark," in *SIGMOD*, 1993, pp. 12–21.

[18] C. Lee, W. Mangione-Smith, and M. Potkonjak, "Mediabench: A tool for evaluating multimedia and communication systems," in *MICRO*, 1997, pp. 330–335.

[19] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.

[20] R. Plackett and J. Burman, "The design of optimum multifactorial experiments," *Biometrika*, vol. 33, no. 4, pp. 305–325, 1946.

[21] G. Box and J. Hunter, "The 2^{k-p} fractional factorial designs parts I and II," *Technometrics*, vol. 3, no. 3/4, pp. 311–351/449–458, 1961.

[22] Center for Embedded Networked Sensing, "http://research.cens.ucla.edu/."

[23] J. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Allerton*, 1979, pp. 480–489.

[24] H. Breu and D. G. Kirkpatrick, "Unit disk graph recognition is NP-hard," *Computational Geometry*, vol. 9, no. 1-2, pp. 3–24, 1998.

[25] D. Achlioptas, A. Naor, and Y. Peres, "Rigorous location of phase transitions in hard optimization problems," *Nature*, no. 435, pp. 759–764, 2005.

[26] CPLEX mathematical optimization package, seen in Nov. 2007, "http://www.ilog.com/products/cplex/."

[27] Localization simulator, seen in Nov. 2007, "http://www.cs.virginia.edu/ white-house/research/localization/."

[28] K. Whitehouse and D. Culler, "A robustness analysis of multi-hop ranging-based localization approximations," in *IPSN*, 2006, pp. 317–325.

[29] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *MobiHoc*, 2003, pp. 201–212.

[30] Y. Shang and W. Ruml, "Improved mds-based localization," in *Infocom*, 2004, pp. 2640–2651.

[31] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *IWSNA*, 2002, pp. 112–121.

[32] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *Globecom*, 2001, pp. 2926–2931.