

# Sequential Greedy Localization in Wireless Sensor Networks With Inaccurate Anchor Positions

Qingjiang Shi<sup>†</sup>, Chen He<sup>†</sup>, Hongyang Chen<sup>‡</sup>, Lingge Jiang<sup>†</sup>, Wei Wang<sup>†</sup>

<sup>†</sup>Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, P. R. China

<sup>‡</sup>Institute of Industrial Science, The University of Tokyo, Japan

Email: {shiqj, chenhe, dolphinwang}@sjtu.edu.cn, hongyang@mcl.iis.u-tokyo.ac.jp

**Abstract**—In this paper, we consider the range-based sensor network localization with inaccurate anchor position information. First, a novel optimization algorithm named sequential greedy optimization (SGO) algorithm is proposed, and then two distributed localization algorithms are obtained: the first is obtained by applying the SGO algorithm to a convex formulation of the localization problem, named CSGLA; while the second is obtained by applying the SGO algorithm to a nonconvex formulation of the localization problem, named NCSGLA. The CSGLA must converge globally while the NCSGLA may converge locally. Both algorithms are partially asynchronous and can be implemented in a distributed fashion in networks. We demonstrate the localization performance via simulations. Simulation results show that, 1) the CSGLA algorithm works faster than the synchronous algorithm [16] with the same localization accuracy; 2) with a reasonably good initialization, the NCSGLA can work much better than the CSGLA.

## I. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of tiny, low-cost, and low-power sensor nodes that have sensing, processing, and communication capabilities [1]. Such networks have many applications such as environmental monitoring, search and rescue, target tracking, etc., most of which require node position information. However, it is too expensive to equip all nodes with Global Positioning System (GPS) receivers or configure the location for each node manually. Hence, node localization is an important topic in the field of WSNs.

Based on whether range measurements are used, there are two types of node localization problems. One is range-based localization and the other is range-free localization. Correspondingly, the localization algorithms can be divided into two categories: range-based algorithms and range-free algorithms. In this paper, we focus on the range-based node localization problem. Consider a sensor network  $(\mathcal{V}, \mathcal{A}, \mathcal{E}, \mathcal{D})$  where  $\mathcal{V}$  denotes the set of nodes of the network,  $\mathcal{A} \subset \mathcal{V}$  is the set of nodes whose positions are known (these nodes are commonly called anchors),  $\mathcal{E}$  is the set of edges between nodes, and  $\mathcal{D} \subset \mathcal{E}$  denotes the edges that are measured. The range-based localization problem is to find a realization of  $\mathcal{V}$  given the incomplete distance information  $\mathcal{D}$  and anchor position information  $\mathcal{A}$ . Range measurements are usually obtained using received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), or angle of arrival (AOA) techniques. Due to the existence of fading and interference, they are noisy in general, which complicate

the localization problem. In addition, the anchor positions cannot be exactly known since the accuracy of the civilian GPS is limited while the manual configuration may subject to observation errors. This further complicates the problem.

Localization algorithms can also be classified as either centralized algorithms or distributed algorithms based on the computational fashion. Centralized algorithms require transmission of all range or connectivity information between nodes to a fusion center (e.g., a sink node) for location estimation, which leads to large communication energy and bandwidth consumption, shortening the lifetime of the whole network (the nodes close to the sink node cost more energy and thus may foremost die out since they have to relay the information transmitted from other nodes). The distributed algorithms are energy-efficient and scalable to the size of the networks, where the whole task of node localization is cooperatively carried out by all nodes accompanied with local information exchange between neighboring nodes. Hence, it is much desirable to develop distributed node localization algorithms for WSNs.

In this paper, we consider the range-based localization problem with inaccurate anchor position information. We first present a novel iterative optimization algorithm, named sequential greedy optimization (SGO) algorithm, which is extraordinarily suitable for distributed optimization in networks. Based on the SGO algorithm and different formulations of the localization problem, two distributed localization algorithms are proposed. One is convex sequential greedy localization algorithm (CSGLA) and the other is nonconvex sequential greedy localization algorithm (NCSGLA). They can be implemented partially asynchronously, i.e., except neighboring nodes, other nodes can run algorithms concurrently, thus more suitable for in-network processing than the synchronous algorithms [16], [18], [19]. With a reasonably good initialization, the NCSGLA can work better than the CSGLA.

## II. RELATED WORK

In essence, the range-based localization problem is a complex nonconvex optimization problem. Perfectly solving this problem is not trivial. Many localization methods have been proposed for the range-based localization. Some are numerical, e.g., [16]–[19], while some have rigorous theoretical analysis for the convergence, e.g., [10]–[13]. Among the latter, convex relaxation techniques such as second order cone

programming (SOCP) relaxation and semidefinite programming (SDP) relaxation are commonly used to approximate the problem. The relaxation-based methods take advantage of the efficient polynomial algorithm–interior point algorithm to solve the resultant convex optimization problem. Due to relaxation, they may work poorly when there are not many anchors on the boundary of networks. However, in general, they can localize a large portion of nodes in networks. Here we review these methods, since they can be easily generalized to deal with the case when anchors are not accurately positioned.

In [10], discarding all nonconvex constraints, Doherty et al. approximate the localization problem by solving a convex feasibility problem using SOCP. Another localization method using SOCP is proposed in [11], which is better than Doherty's method since the latter cannot work when the resultant feasibility problem has no solution. The SDP relaxation technique can provide tighter relaxation than the SOCP relaxation technique, and thus yields better localization performance. The first SDP method for sensor network localization is proposed in [12] and other SDP relaxation models can be found in [13]. The SDP methods can solve the localization problem perfectly when there is no noise in the range measurements, if the network is uniquely localizable. However, in the noisy case, they work very poorly, due to the fact that the SDP solution tends to be higher rank than the practical physical space in order to minimize the objective. To reduce the rank of the SDP solution, the authors add a regularization term to the objective in [14], leading to better localization performance for the noisy case.

Generally, the SDP methods cannot be implemented in a distributed way in networks due to their complex structures, although [15] provides a cluster-based distributed SDP method which behaves poorly when there are less anchors in networks and is not easy to implement in networks. In [16], a distributed SOCP method has been proposed, which can work in a totally distributed fashion in network, either synchronously or totally asynchronously. Moreover, the authors of [16] consider the case of inaccurate anchors. However, they demonstrate the convergence of their algorithm numerically and do not provide a theoretical guarantee for the convergence of the algorithm.

### III. SEQUENTIAL GREEDY OPTIMIZATION ALGORITHM

Many optimization algorithms have been developed for multivariate problems [6], [7]. Most of them fall into the category of iterative algorithms which are generally based on search directions. In contrast to general iterative optimization algorithms, there exists a special type of iterative algorithm, at each iteration of which, only one or a small fraction of variables are optimized while the rest are fixed [2]–[5]. For instance, the classical alternating minimization (or projection) algorithm [2] works by alternately minimizing an objective function of two variables<sup>1</sup> over one variable while fixing the other. We say this type of algorithm is greedy in the sense that it can guarantee the objective function nonincreases at

<sup>1</sup>Here, the variable can be any  $n$ -dimensional.

each iteration. By using such a greedy algorithm, a complex multivariate problem can be broken down into a sequence of small-scale subproblems and the algorithm can be easily realized, even in a distributed way.

The idea can be easily extended to the general multivariate problems (i.e., with more than two variables). However, such a greedy algorithm cannot always converge to local optima although it must converge if the objective function is bounded. For example, for the problem

$$\begin{aligned} & \min x_1 + x_2 \\ & \text{s.t. } x_1 x_2 \geq 1, x_1 \geq 0, x_2 \geq 0, \end{aligned} \quad (1)$$

the algorithm may converge to  $(c, \frac{1}{c})$  where  $c > 1$  which is not a local optimum of the problem. This hinders the application of this type of greedy algorithm.

In this section, we establish the algorithm in a more general form, named sequential greedy optimization (SGO) algorithm. We prove that the SGO algorithm must converge to local optima for sparse-constrained problems (see the definition below).

#### A. Algorithm

We say an optimization problem is *sparse-constrained* if it can be written in the form

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{s.t. } \mathbf{g}_i(\mathbf{x}_i) \preceq \mathbf{0}, \mathbf{h}_i(\mathbf{x}_i) = 0, i = 1, 2, \dots, n \end{aligned} \quad (2)$$

where,  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ ,  $\mathbf{x}_i \in \mathbb{R}^{d_i}$ ,  $d_i$  is a positive integer,  $i = 1, 2, \dots, n$ ;  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ ,  $N = \sum_{i=1}^n d_i$ ;  $\mathbf{g}_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{m_i}$ ,  $\mathbf{h}_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{n_i}$ ,  $i = 1, 2, \dots, n$ , are continuously differentiable functions;  $m_i$  and  $n_i$ ,  $i = 1, 2, \dots, n$ , are nonnegative integers. We assume the minimum of  $f(\mathbf{x})$  exists. A typical example of sparse-constrained problems is the box-constrained quadratic programming. Note that,  $m_i = 0$  ( $n_i = 0$ ) means there is no inequality (equality) constraint on  $\mathbf{x}_i$ . Particularly, when all  $m_i$  and  $n_i$  are equal to zero, the problem (2) is translated into a special sparse-constrained problem, i.e., unconstrained optimization problem.

Generally, the SGO algorithm sequentially minimizes subproblems. In fact, however, the SGO algorithm does not necessarily minimize each subproblem. Here, we will describe the SGO algorithm with a new operator 'dec' instead of 'min'. With the new operator, the expression ' $\mathbf{z}^k = \arg \text{dec}_{\mathbf{z} \in \mathbb{B}} \varphi(\mathbf{z})$ ', in an iterative algorithm is used to denote finding a new  $\mathbf{z} \in \mathbb{B}$  at  $k$ -th iteration (denoted by  $\mathbf{z}^k$ ) such that  $\varphi(\mathbf{z}^k) \leq \varphi(\mathbf{z}^{k-1})$  given  $\mathbf{z}^{k-1}$ . Note that, if using certain optimization algorithm we cannot find a new  $\mathbf{z}$  such that  $\varphi(\mathbf{z}^k) < \varphi(\mathbf{z}^{k-1})$ , then let  $\mathbf{z}^k = \mathbf{z}^{k-1}$ .

In addition, we will use the following notations when describing the algorithm. Let  $S_i = \{\mathbf{y} \in \mathbb{R}^{d_i} \mid \mathbf{g}_i(\mathbf{y}) \preceq \mathbf{0}, \mathbf{h}_i(\mathbf{y}) = \mathbf{0}\}$ ,  $i = 1, 2, \dots, n$ , and  $S = \times_{i=1}^n S_i$ , i.e.,  $\{\mathbf{x} \in \mathbb{R}^N \mid \mathbf{x}_1 \in S_1, \mathbf{x}_2 \in S_2, \dots, \mathbf{x}_n \in S_n\}$ . Let  $\mathbb{N}_j$ ,  $j = 1, 2, \dots, p$ , be subsets of  $\mathbb{N} = \{1, 2, \dots, n\}$  and such that  $\bigcup_{j=1}^p \mathbb{N}_j = \mathbb{N}$ . Let  $\mathbb{L}_i$  be the  $i$ -th index set whose components are the indexes of those  $\mathbb{N}_j$ s which contain  $i$ . For example, if  $2 \in \mathbb{N}_1 \cap \mathbb{N}_2 \cap \mathbb{N}_5$ , then  $\{1, 2, 5\} \subseteq \mathbb{L}_2$ . In the proof of

Proposition 1 below, it will mean that the variable  $\mathbf{x}_2$  is not only optimized in the subproblem 1, but also in the subproblem 2 and 5.

The SGO algorithm is described as follows. *Given  $\mathbf{x}^{k-1}$  obtained after the  $(k-1)$ -th iteration, it then sequentially performs  $p$  greedy steps or subiterations in the  $k$ -th iteration:*

$$\begin{aligned} \mathbf{x}_{\mathbb{N}_1}^k &= \arg \operatorname{dec}_{\mathbf{x}_{\mathbb{N}_1} \in S_{\mathbb{N}_1}} f(\mathbf{x}_{\mathbb{N}_1}, \mathbf{x}_{-\mathbb{N}_1}^{k-1}) \\ \mathbf{x}_{\mathbb{N}_2}^k &= \arg \operatorname{dec}_{\mathbf{x}_{\mathbb{N}_2} \in S_{\mathbb{N}_2}} f(\mathbf{x}_{\mathbb{N}_2}, \mathbf{x}_{-\mathbb{N}_2}^{k-1}) \\ &\vdots \\ \mathbf{x}_{\mathbb{N}_p}^k &= \arg \operatorname{dec}_{\mathbf{x}_{\mathbb{N}_p} \in S_{\mathbb{N}_p}} f(\mathbf{x}_{\mathbb{N}_p}, \mathbf{x}_{-\mathbb{N}_p}^{k,p-1}) \end{aligned} \quad (3)$$

and finally get  $\mathbf{x}^k = \mathbf{x}^{k-1,p}$ , where  $\mathbf{x}^{k,j}$  denotes the new  $\mathbf{x}$  obtained after the  $j$ -th subiteration of the  $k$ -th iteration (Note,  $\mathbf{x}^{k,0} = \mathbf{x}^{k-1}$ );  $\mathbf{x}_{\mathbb{N}_j}$  ( $\mathbf{x}_{-\mathbb{N}_j}$ ) denotes the variable made up of  $\mathbf{x}_i$ s indexed by the components of  $\mathbb{N}_j$  ( $\mathbb{N} \setminus \mathbb{N}_j$ );  $S_{\mathbb{N}_j} = \times_{i \in \mathbb{N}_j} S_i$ .

### B. Convergence analysis and discussion

**Proposition 1:** *For the sparse-constrained problem (2), the SGO algorithm converges to a local optimum under some regularity conditions [9].*

*Proof:* Let  $f_{k,j} \triangleq f(\mathbf{x}_{\mathbb{N}_j}^k, \mathbf{x}_{-\mathbb{N}_j}^{k,j-1})$ , so we have  $f_{k,p} = f(\mathbf{x}^k)$ . From (3), we conclude that

$$f_{k,p} \leq f_{k,p-1} \leq \dots \leq f_{k,1} \leq f(k-1, p).$$

Thus we have for all  $k > 0$

$$f(\mathbf{x}^k) \leq f(\mathbf{x}^{k-1}).$$

This implies the sequence  $\{f(\mathbf{x}^k)\}_{k=0}^\infty$  is nonincreasing, and must converge since  $f(\mathbf{x})$  is bounded below by its minimum on  $S$ . It follows that  $\{\mathbf{x}\}_{k=0}^\infty$  converge in view of the algorithmic rule.

Let  $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_n)$  be the convergence point, so we must have

$$\bar{\mathbf{x}}_{\mathbb{N}_j} = \arg \operatorname{dec}_{\mathbf{x}_{\mathbb{N}_j} \in S_{\mathbb{N}_j}} f(\mathbf{x}_{\mathbb{N}_j}, \bar{\mathbf{x}}_{-\mathbb{N}_j}), \quad j = 1, 2, \dots, p, \quad (4)$$

implying that, for each subproblems,  $\bar{\mathbf{x}}_{\mathbb{N}_j}$  must be at least a local minimizer. Under the assumption of regularity for each subproblem, the first-order necessary condition, generally known as *Karush–Kuhn–Tucker* (KKT) condition, holds for each subproblem, i.e., for each  $j \in \{1, 2, \dots, p\}$  and  $i \in \mathbb{N}_j$ , there exist Lagrange multipliers  $\lambda_i^j \in \mathbb{R}^{m_i}$  and  $\mu_i^j \in \mathbb{R}^{n_i}$ , such that<sup>2</sup>

$$\begin{cases} \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}) + \mathbf{J}_i^T(\bar{\mathbf{x}}_i) \boldsymbol{\lambda}_i^j + \bar{\mathbf{J}}_i^T(\bar{\mathbf{x}}_i) \boldsymbol{\mu}_i^j = \mathbf{0} \\ \mathbf{g}_i(\bar{\mathbf{x}}_i) \preceq \mathbf{0}, \quad \mathbf{h}_i(\bar{\mathbf{x}}_i) = \mathbf{0} \\ \boldsymbol{\lambda}_i^j \succeq \mathbf{0}, \quad \boldsymbol{\lambda}_i^j \cdot \mathbf{g}_i(\bar{\mathbf{x}}_i) = \mathbf{0} \end{cases} \quad (5)$$

<sup>2</sup>When  $m_i = 0$  ( $n_i = 0$ ), the inequality (equality) constraints-related terms in (5) should be canceled. For example, when  $m_i = 0$  and  $n_i \neq 0$ , (5) can be simplified as  $\{\nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}) + \bar{\mathbf{J}}_i^T(\bar{\mathbf{x}}_i) \boldsymbol{\mu}_i^j = \mathbf{0}, \mathbf{h}_i(\bar{\mathbf{x}}_i) = \mathbf{0}\}$ . In this paper, for simplicity, we write the first-order necessary condition in a unified framework.

where  $\mathbf{J}_i(\mathbf{x}_i) = [\nabla_{\mathbf{x}_i} g_{i,1}(\mathbf{x}_i) \quad \nabla_{\mathbf{x}_i} g_{i,2}(\mathbf{x}_i) \quad \dots \quad \nabla_{\mathbf{x}_i} g_{i,m_i}(\mathbf{x}_i)]^T$  is the Jacobi matrix of  $\mathbf{g}_i(\mathbf{x}_i)$  w.r.t.  $\mathbf{x}_i$ ,  $\bar{\mathbf{J}}_i(\mathbf{x}_i) = [\nabla_{\mathbf{x}_i} h_{i,1}(\mathbf{x}_i) \quad \nabla_{\mathbf{x}_i} h_{i,2}(\mathbf{x}_i) \quad \dots \quad \nabla_{\mathbf{x}_i} h_{i,n_i}(\mathbf{x}_i)]^T$  is the Jacobi matrix of  $\mathbf{h}_i(\mathbf{x}_i)$  w.r.t.  $\mathbf{x}_i$ , and the symbol ‘ $\cdot$ ’ denotes the component by component multiplication of vectors.

Combining the KKT conditions of all subproblems, we have

$$\begin{cases} \nabla_{\mathbf{x}} f(\bar{\mathbf{x}}) + \mathbf{J}_i^T(\bar{\mathbf{x}}_i) \boldsymbol{\lambda}_i + \bar{\mathbf{J}}_i^T(\bar{\mathbf{x}}_i) \boldsymbol{\mu}_i = \mathbf{0}, \\ \mathbf{g}_i(\bar{\mathbf{x}}_i) \preceq \mathbf{0}, \quad \mathbf{h}_i(\bar{\mathbf{x}}_i) = \mathbf{0}, \\ \boldsymbol{\lambda}_i \succeq \mathbf{0}, \quad \boldsymbol{\lambda}_i \cdot \mathbf{g}_i(\bar{\mathbf{x}}_i) = \mathbf{0}, \\ i = 1, 2, \dots, n \end{cases} \quad (6)$$

where  $\lambda_i = \frac{1}{|\mathbb{L}_i|} \sum_{j \in \mathbb{L}_i} \lambda_i^j$ ,  $\mu_i = \frac{1}{|\mathbb{L}_i|} \sum_{j \in \mathbb{L}_i} \mu_i^j$ .

This implies that for the convergence point  $\bar{\mathbf{x}}$ , there exist Lagrange multiplier  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_n)$  and  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)$  satisfying the KKT condition of the primal problem. Therefore,  $\bar{\mathbf{x}}$  is a local minimizer of the primal problem. ■

Based on Proposition 1 and the fact that local optima are also global optima for a convex optimization problem, we can straightforwardly obtain the following proposition.

**Proposition 2:** *For convex sparse-constrained problems, the SGO algorithm can converge globally.*

Two important remarks are made as follows:

**Remark 1:** if  $\mathbb{N}_i = \{i\}$  and  $p = n$ , then the SGO algorithm reduces to

$$\begin{aligned} \mathbf{x}_1^k &= \arg \operatorname{dec}_{\mathbf{x}_1 \in S_1} f(\mathbf{x}_1, \mathbf{x}_2^{k-1}, \mathbf{x}_3^{k-1}, \dots, \mathbf{x}_n^{k-1}) \\ \mathbf{x}_2^k &= \arg \operatorname{dec}_{\mathbf{x}_2 \in S_2} f(\mathbf{x}_1^k, \mathbf{x}_2, \mathbf{x}_3^{k-1}, \mathbf{x}_4^{k-1}, \dots, \mathbf{x}_n^{k-1}) \\ &\vdots \\ \mathbf{x}_n^k &= \arg \operatorname{dec}_{\mathbf{x}_n \in S_n} f(\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{n-1}^k, \mathbf{x}_n) \end{aligned} \quad (7)$$

The SGO algorithm of (7) is a special case of (3) (and known as nonlinear Gauss–Siedel algorithm [8]) but it is usually used in practice, while the algorithm (3) can be exploited to devise some distributed algorithms for in-network processing [22].

**Remark 2:** The order of the subproblems can be varied at different iterations. Moreover, it is not necessary to run all the subproblems at each iteration. These will make the algorithm much more suitable for asynchronous implementation<sup>3</sup> in networks but not influence the convergence of the algorithm.

## IV. SEQUENTIAL GREEDY NODE LOCALIZATION ALGORITHM

In this section, based on the SGO algorithm, we propose two distributed localization algorithms, respectively named convex sequential greedy localization algorithm (CSGLA) and non-convex sequential greedy localization algorithm (NCSGLA). The former is obtained by applying the SGO algorithm to a SOCP relaxation-based convex formulation of the localization problem while the latter is obtained by directly applying the

<sup>3</sup>In implementing the SGO algorithm, it is not allowed that two subproblems, which contain coupling variable (i.e., when  $\mathbb{N}_i \cap \mathbb{N}_j \neq \emptyset$ ), run concurrently. Hence, ‘asynchronous’ here doesn’t mean ‘totally asynchronous’ [8].

SGO algorithm to the nonconvex least-squares formulation of the localization problem. The NCSGLA may converge to local optimum, while the CSGLA can converge globally.

#### A. Problem statement

Assume a sensor network in  $\mathbb{R}^2$  has  $N$  sensors in total, with  $m$  anchor nodes whose locations are inaccurately known and  $N - m$  sensor nodes whose locations are unknown ( $m < N - m$ ), and the maximum radio ranges of the nodes are all  $R$ . Let  $\mathbf{x}_i$  and  $\mathbf{a}_i$ ,  $i = 1, 2, \dots, m$ , denote the exact locations and the inaccurate locations of anchor nodes, respectively, and let  $\mathbf{x}_i$ ,  $i = m + 1, m + 2, \dots, N$ , denote the locations of sensor nodes. Noisy range measurements are taken by any two nodes if they are neighbors (i.e., they can communicate with each other directly), denoted by  $d_{ij}$  for neighboring nodes  $i$  and  $j$ . In addition, we assume that the location errors of anchors are bounded by  $\delta$ , i.e.,  $\|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta$ . Then the range-based localization problem is to find a realization of  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  such that

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\| &\approx d_{ij}, \forall (i, j) \in \mathcal{N}, \\ \|\mathbf{x}_i - \mathbf{x}_j\| &> R, \forall (i, j) \notin \mathcal{N}, \\ \|\mathbf{x}_i - \mathbf{a}_i\| &\leq \delta, i = 1, 2, \dots, m \end{aligned} \quad (8)$$

where  $\mathcal{N}$  denotes the set of index pairs  $(i, j)$  of neighboring nodes with  $i < j$ .

#### B. Convex sequential greedy localization algorithm

By neglecting the nonconvex inequality constraints in (8), the range-based node localization problem can be cast as the following constrained least-squares problem

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} \sum_{(i, j) \in \mathcal{N}} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta, i = 1, 2, \dots, m. \end{aligned} \quad (9)$$

It is a complex nonconvex problem that is difficult to solve. Similar to [11], [16], we relax it as the following convex problem

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_N, y_{ij}s} \sum_{(i, j) \in \mathcal{N}} (y_{ij} - d_{ij})^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{x}_j\| \leq y_{ij}, \forall (i, j) \in \mathcal{N}, \\ \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta, i = 1, 2, \dots, m. \end{aligned} \quad (10)$$

This problem is not sparse-constrained. A straightforward way to turn the problem into a sparse-constrained one is using the log-barrier method [7] [22]. However, a simpler way can be used, noticing the special structure of this problem.

Notice that

$$\min_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq y_{ij}} (y_{ij} - d_{ij})^2 \equiv (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \quad (11)$$

where  $(\cdot)_+ = \max(\cdot, 0)$ . Hence, (10) is equivalent to

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} \sum_{(i, j) \in \mathcal{N}} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta, i = 1, 2, \dots, m \end{aligned} \quad (12)$$

which is a sparse-constrained convex problem and thus can be solved by the SGO algorithm with global convergence. We decompose the whole problem into  $N$  subproblems, each of which corresponds to one node and can be obtained by fixing the neighboring nodes location, leading to the CSGLA. The  $i$ -th subproblem is

$$\begin{aligned} \min_{\mathbf{x}_i} \sum_{j \in \mathcal{N}_i} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta \end{aligned} \quad (13)$$

if the node  $i$  is an anchor, or

$$\min_{\mathbf{x}_i} \sum_{j \in \mathcal{N}_i} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \quad (14)$$

if the node  $i$  is a generic sensor node. Here,  $\mathcal{N}_i$  denotes the index set of the neighboring nodes of the node  $i$ . Note that, the subproblem can be solved in the node  $i$  upon receiving the estimates from neighboring nodes. Hence, the CSGLA can be implemented in a distributed way. In particular, since the subproblems are only related to the estimates of neighboring nodes, the CSGLA can be implemented partially asynchronously, i.e., any two non-neighbor nodes can run their algorithm simultaneously.

Interestingly, to solve (13) and (14), we can instead respectively solve the following two SOCP problems:

$$\begin{aligned} \min_{\mathbf{x}_i, y_{ij}s} \sum_{j \in \mathcal{N}_i} (y_{ij} - d_{ij})^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta \\ \|\mathbf{x}_i - \mathbf{x}_j\| \leq y_{ij}, j \in \mathcal{N}_i \end{aligned} \quad (15)$$

and

$$\begin{aligned} \min_{\mathbf{x}_i, y_{ij}s} \sum_{j \in \mathcal{N}_i} (y_{ij} - d_{ij})^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{x}_j\| \leq y_{ij}, j \in \mathcal{N}_i \end{aligned} \quad (16)$$

which can be solved by using the efficient primal-dual interior algorithm [7]. In this paper, we use the popular cone programming solver SeDuMi [20] to solve (15) and (16).

#### C. Nonconvex sequential greedy localization algorithm

Clearly, (9) is sparse-constrained, so it can be directly solved by the SGO algorithm with local convergence, leading to the NCSGLA. In the NCSGLA, the  $i$ -th subproblem is

$$\begin{aligned} \min_{\mathbf{x}_i} \sum_{j \in \mathcal{N}_i} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta \end{aligned} \quad (17)$$

corresponding to the anchor node  $i$ , or

$$\min_{\mathbf{x}_i} \sum_{j \in \mathcal{N}_i} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})_+^2 \quad (18)$$

corresponding to the sensor node  $i$ . The above two subproblems can be solved by using some nonlinear optimization methods, e.g., Newton method or Sequential Quadratic Programming. Interestingly, these problems can be turned into sparse-constrained ones and thus solved by the SGO algorithm.

By introducing intermediate variables, (17) can be equivalently written in a sparse-constrained form as follows

$$\begin{aligned} \min_{\mathbf{x}_i, \mathbf{v}_j} \sum_{j \in \mathcal{N}_i} \|\mathbf{x}_i - \mathbf{v}_j\|^2 \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}_i\| \leq \delta \\ \|\mathbf{v}_j - \mathbf{x}_j\|^2 = d_{ij}^2, j \in \mathcal{N}_i \end{aligned} \quad (19)$$

where  $\mathbf{v}_j$ s are intermediate variables. The equivalence between (17) and (19) can be proved by noticing that, for any fixed  $\mathbf{x}_i \neq \mathbf{x}_j$ , we must have

$$\mathbf{v}_j = \mathbf{x}_j + d_{ij} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \quad (20)$$

to minimize  $\|\mathbf{x}_i - \mathbf{v}_j\|^2$  over  $\mathbf{v}_j$  subject to  $\|\mathbf{v}_j - \mathbf{x}_j\|^2 = d_{ij}^2$ .

Now we can apply the SGO algorithm to (19). Fixing  $\mathbf{x}_i$ , we have (20), while fixing  $\mathbf{v}_j$ s, we get

$$\mathbf{x}_i = \begin{cases} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j, & \text{if } \frac{1}{|\mathcal{N}_i|} \left\| \sum_{j \in \mathcal{N}_i} (\mathbf{v}_j - \mathbf{a}_i) \right\| \leq \delta \\ \mathbf{a}_i + \delta \frac{\sum_{j \in \mathcal{N}_i} (\mathbf{v}_j - \mathbf{a}_i)}{\left\| \sum_{j \in \mathcal{N}_i} (\mathbf{v}_j - \mathbf{a}_i) \right\|}, & \text{otherwise} \end{cases} \quad (21)$$

Hence, a stepsize-free iterative algorithm for (17) can be obtained, i.e., given  $\mathbf{x}_i^{k-1}$ ,  $\mathbf{x}_i^k$  can be derived from (21) where  $\mathbf{v}_j$  is calculated according to (20) with  $\mathbf{x}_i$  being replaced with  $\mathbf{x}_i^{k-1}$ .

Similarly, the problem (18) can also be written in a sparse-constrained form and then solved by the SGO algorithm, yielding the following simple iterative solution

$$\mathbf{x}_i^k = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left( \mathbf{x}_j + d_{ij} \frac{\mathbf{x}_i^{k-1} - \mathbf{x}_j}{\|\mathbf{x}_i^{k-1} - \mathbf{x}_j\|} \right). \quad (22)$$

## V. SIMULATION RESULTS

In this section, we conduct simulations to evaluate the performance of the proposed CSGLA and NCSGLA. In simulations, the NCSGLA is initialized from the result of the CSGLA or the simple DV-Hop method [21], while the CSGLA is randomly initialized from the sensing region.

Simulations are performed on networks of 110 randomly distributed nodes in a square region of  $[0, 1] \times [0, 1]$  with 10 randomly positioned anchor nodes. The radio range of each node is set to be 0.2, i.e.,  $R = 0.2$ , leading to an average connectivity level (the average number of nodes within the communication range of each node) of about 11. The noisy range measurements are generated according to

$$\hat{d} = d \times |1 + nf \times randn| \quad (23)$$

where  $d$  denotes the true range measurement,  $nf$  is the noise factor,  $randn$  is a standard norm variable. The inaccurate anchor positions are randomly chosen from the disks centered at the true anchor locations with the radius of  $\delta$ .

### A. Convergence performance

In this set of simulations, we examine the convergence performance of the CSGLA and the NCSGLA. The convergence performance of the synchronous algorithm [16] is also evaluated. Particularly, we set  $\delta = 0$  and  $nf = 0$ , so that the minimum cost function values of (9) and (12) both are 0. The results are presented in Fig. 1. From the figure, it can be observed that the CSGLA and the NCSLA converge with the property of nonincreasing at each iteration, which is consistent with our theoretical analysis. In addition, we can find that the CSGLA is faster than the synchronous algorithm.

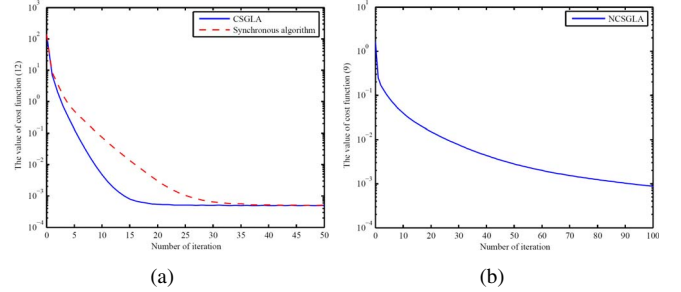


Fig. 1. The convergence performance: (a) The CSGLA and the synchronous algorithm [16]; (b) The NCSGLA algorithm

### B. Localization performance

In this subsection, we evaluate the localization performance of the CSGLA and the NCSGLA in terms of the localization error calculated according to  $\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i^* - \mathbf{x}_i\|$  where  $\mathbf{x}^*$  denotes the true location. In simulations, we set  $nf = 0.1$  and  $\delta = 0.2R$ . The CSGLA (also the NCSGLA and the synchronous algorithm) terminates if either  $\max_i \|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\| \leq 0.01R$  or the total number of  $K > 50$ . An example localization of various methods including the centralized SOCP method [i.e., directly solving (10)] and the synchronous algorithm, when most of anchors are placed on the boundary of networks, is presented in Fig. 2, where the true anchors, the inaccurate anchors, and the estimated anchors are respectively denoted by blue, black, red diamonds, the true sensor nodes are denoted by green circles and their estimates by red asterisks, the lines indicate the estimation errors. We can find from the figure that, initialized from the result of the CSGLA, the NCSGLA outperforms the other three methods which almost have the same localization performance.

As is known, the relaxation-based methods behave poorly when there are not many anchors on the boundary of networks. Hence, in this case, the CSGLA cannot provide a good initialization for the NCSGLA, resulting in poor localization performance. Fortunately, a good initialization is still available from a simple distributed method, DV-Hop [21]. Here, we slightly modify the DV-Hop method by using an iterative solution that is similar to (22) to estimate the nodes' location, in place of the triangulation method. The simulation results are presented in Fig. 3. It can be observed that the NCSGLA with the DV-Hop initialization yields better localization performance than it with the CSGLA initialization.

## ACKNOWLEDGEMENT

This work was supported in part by National Nature Science Foundation of China (Grant No. 60772100, 60832009, 60872017) and in part by the Cooperative Research Project with Fujitsu R&D Beijing Center.

## REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," in *Statistic and Decisions, Supplement Issue No. 1*, E. J. Dudewicz, et al., EDS., pp. 205–237, 1984.
- [3] W. S. Wang, J. C. Juang, and Y. Y. Sun, "On the application of alternating projection under the constraint of linear matrix inequalities," *IMA Journal of Mathematical Control and Information*, vol. 16, no. 1, pp. 15–22, 1999.
- [4] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Chapter 12, Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, eds., pp. 185–208, MIT Press, 1999.
- [5] A. Tkacenko and P. P. Vaidyanathan, "Iterative greedy algorithm for solving the FIR Paraunitary approximation problem," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 144–160, Jan. 2006.
- [6] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice Hall, 1983.
- [7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [8] D. P. Bertsekas and J. N. Tsitsiklis, *Parrell and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [9] M. Kuhn, "The Karush–Kuhn–Tucker Theorem", Nov. 2006 [online]. Available: [http://en.wikipedia.org/wiki/Karush-Kuhn-Tucker\\_conditions](http://en.wikipedia.org/wiki/Karush-Kuhn-Tucker_conditions). Also: <http://weburum.uni-mannheim.de/vwl/mokuhn/public/KarushKuhn-Tucker.pdf>
- [10] L. Doherty, L. E. Ghaoui, and S. J. Pister, "Convex position estimation in wireless sensor networks," in *Proc. IEEE INFOCOM*, vol. 3, pp. 1655–1663, Apr. 2002.
- [11] P. Tseng, "Second-order cone programming relaxation of sensor network localization," *SIAM J. Optim.*, vol. 8, no. 1, pp. 156–185, Feb. 2007.
- [12] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. ACM IPSN*, pp. 46–54, 2004.
- [13] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, 2006.
- [14] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, "Semidefinite programming based algorithms for sensor network localization," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.
- [15] P. Biswas and Y. Ye, "A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization," in *Multiscale Optimization Methods and Applications*, pp. 69–84, Jan 2006.
- [16] S. Srirangarjan, A. H. Tewfik, and Z.-Q. Luo, "Distributed sensor network localization using SOCP relaxation," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4886–4895, Dec. 2008.
- [17] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Netw.*, vol. 43, pp. 499–518, 2003.
- [18] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.
- [19] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *Proc. ACM MobiHoc*, pp. 250–261, May. 2006.
- [20] J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones", *Optimiz. Meth. Softw.*, vol. 11–12, pp. 625–653, 1999.
- [21] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE GLOBECOM*, vol. 7, pp. 2926–2931, 2001.
- [22] Q. Shi, C. He, et al., "Robust Wireless Sensor Network Localization via Sequential Greedy Optimization Algorithm," Tech. Rep. [online]. Available: [http://sites.google.com/site/qingjiangshi/tsp2/technical\\_report.pdf](http://sites.google.com/site/qingjiangshi/tsp2/technical_report.pdf)

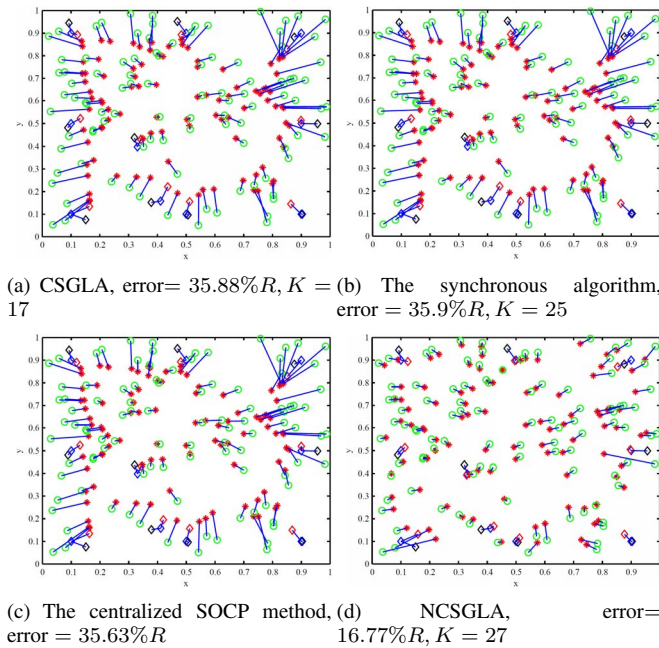


Fig. 2. The localization results of various methods when the anchors are on the boundary of networks.

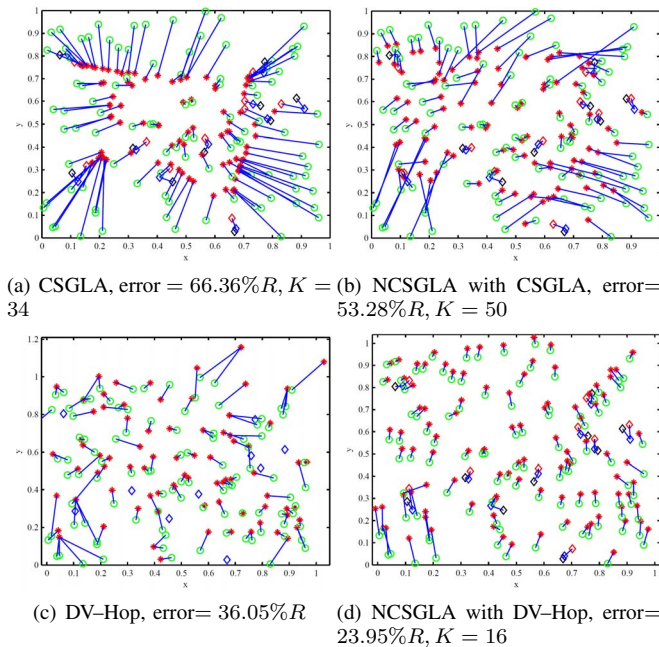


Fig. 3. The localization results of various methods when there are not many anchors on the boundary of networks.

## VI. CONCLUSIONS

We have proposed the sequential greedy optimization algorithm with the property of nonincreasing at each iteration. Although in this paper we only consider the application of the algorithm in SOCP-based node localization, the SGO algorithm can be also applicable to distributed SDP-based node localization [22], where the SDP localization model is a general multivariate problem (i.e., not sparse-constrained).