A nonsmooth optimization approach to sensor network localization

Adil Bagirov [1], Daniel T.H. Lai [2], M. Palaniswami [2]
[1] *School of Information Technology and Mathematical Sciences, University of Ballarat, PO Box 663, Ballarat Victoria 3353 Australia, email: a.bagirov@ballarat.edu.au*
[2] *Department of Electrical and Electronic Engineering, University of Melbourne, Parkville Campus, Victoria 3010 Australia, email: d.lai, swami@ee.unimelb.edu.au*

Abstract

In this paper the problem of localization of wireless sensor network is formulated as an unconstrained nonsmooth optimization problem. We minimize a distance objective function which incorporates unknown sensor nodes and nodes with known positions (anchors) in contrast to popular semidefinite programming (SDP) methods which use artificial objective functions. We study the main properties of the objective function in this problem and design an algorithm for its minimization. Our algorithm is a derivative-free discrete gradient method that allows one to find a near global solution. The algorithm can handle a large number of sensors in the network. This paper contains the theory of our proposed formulation and algorithm while experimental results are included in later work.

Key words and phrases: sensor networks, nonsmooth optimization, derivative free algorithms.

1. Introduction

Wireless sensor networks (WSN) have drawn great interest recently, with applications ranging from environmental monitoring, patient observation in healthcare monitoring to military tracking on battlefields [1], [8]. Sensor networks potentially consist of a large number of sensor nodes (hundreds to thousands) which may be for example, strategically placed in hospitals for patient monitoring or randomly positioned (scattered) e.g., airdropped on a battlefield. One of the major deployment issues of wireless sensor networks is node placement which comprises two distinct but related problems. The first problem is known as *optimal placement* where one has to strategically position a number of nodes to fulfil certain criteria e.g., maximum coverage [22]. The second problem which is the focus of this paper is *self-localization*, where the node positions are unknown and the network tries to discover their positions [18]. This is achieved using only sensor information such as received signal strength (RSS), time difference of arrival (TDOA) [17] and hop connectivities [16]. Self-localization is important because sensor data frequently require node positions to be of practical use, for example tracking troop movements on the battlefield requires accurate knowledge of sensor positions. While global positioning systems (GPS) are attractive solutions for WSN localization, they are not yet cost effective enough to be deployed on every sensor node and may not function well in enclosed areas with no satellite line of sight.

Self-localization is not an easy problem as evidenced by the many existing techniques such as semidefinite programming (SDP) [6], [7], multidimensional scaling (MDS) [10], [19], [20], particle filter modelling [13] and kernel methods [15]. In SDP methods, one assumes that the network consists of N_x nodes with unknown positions and N_a nodes with known positions called *anchors*. These methods are easy to implement due to the availability of existing algorithms but suffer from dimensionality problems. The number of SDP variables increases quadratically regardless of the number of unknown nodes and anchor nodes causing larger problems to be difficult to handle. Modifications to the SDP method include relaxation [6] techniques which provide a faster localization algorithm at the cost of an approximate solution. MDS methods are attractive because they do not require anchor nodes, however the algorithms suffer from local minima meaning that algorithm initial conditions determine the final topology estimates. Local minima nevertheless could be handled by expressing node proximities as convex constraints and employing convex programming algorithms [12].

In this paper, we propose another method to handle local minima using a nonsmooth optimization approach. The original SDP constraints is first transformed to a least squares minimization problem and a derivative free nonsmooth optimization algorithm is applied to solve it. Our algorithm can handle nonsmooth objective functions and searches for a better solution as opposed to previous methods which terminate at approximate solutions. The algorithm finds a sequence of local solutions and specifically exploits the problem structure making it applicable for large scale sensor network localization problems.

The paper is organized as follows. In Section 2 the nonsmooth optimization formulation of wireless sensor network problem is given. The properties of the objective function in the optimization problem are described in Section 3. An algorithm for the approximation of subgradients of the objective is presented in Section 4. Finally, the minimization algorithm is described in Section 5 and Section 6 concludes the paper.

2. Formulation of problem

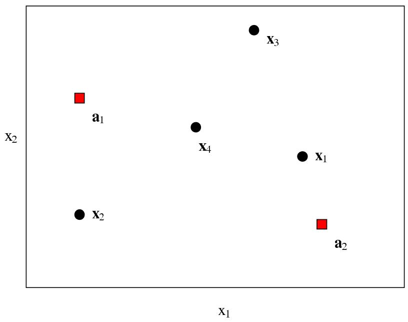

Fig. 1: A simple wireless sensor network topology depicting the coordinate system, nodes $\mathbf{x}_i$ with unknown positions and anchor nodes $\mathbf{a}_j$.

We consider the problem of localization of sensor networks in a planar two dimensional environment. Formally, all points $\mathbf{x} = (x_1, x_2)$ belong to a well defined compact set $X \subset \mathbb{R}^2$ where x_1 and x_2 are coordinates in the two dimensional plane (Figure 1). Assume that for a wireless sensor network in $\mathbb{R}^2$ there are m anchors with known locations $a^k \in \mathbb{R}^2,\ k = 1, \ldots, m$ and n sensors with unknown locations $x^j,\ j = 1, \ldots, n$. We assume that all distances are known exactly, that is we consider the localization problem where measurements are noiseless.

We denote by d_{ij} the Euclidean distance between i-th and j-th sensors while the distance between the i-th sensor and the k-th anchor is denoted by d_{ik}. In general, not all pairs of distances may be known, so the pairs of nodes for which mutual distances are known are denoted as $(i, j) \in N_x$ for sensor/sensor and $(i, k) \in N_a$ for sensor/anchor pairs, respectively. The WSN localization problem [6] can be formally stated as: Given m anchor locations $a^k,\ k = 1, \ldots, m$ and some distance measurements $d_{ij},\ (i, j) \in N_x,\ d_{ik},\ (i, k) \in N_a$ find the locations $x^j,\ j = 1, \ldots, n$ of n sensors such that

$$\|x^i - x^j\|^2 = d_{ij}^2, \forall (i, j) \in N_x, \tag{1}$$

$$\|x^i - a^k\|^2 = d_{ik}^2, \forall (i, k) \in N_a. \tag{2}$$

We now reduce the system of equations (1)-(2) to the following unconstrained optimization problem:

$$\text{minimize} \quad f(x^1, \ldots, x^n) = \sum_{(i,j)\in N_x} \left(\|x^i - x^j\|^2 - d_{ij}^2\right)^2 + \sum_{(i,k)\in N_a} \left(\|x^i - a^k\|^2 - d_{ik}^2\right)^2 \tag{3}$$

subject to

$$x^i \in \mathbb{R}^2,\ j = 1, \ldots, n.$$

The objective function f in Problem (3) is nonconvex, potentially possessing a large number of local minima. The number of local minimizers of this function is also strongly dependant on the number of sensors.

Now let us consider the localization problem with measurements noises. Then we have the following equations:

$$d_{ij} = d(x^i, x^j) + \omega_{ij} \tag{4}$$

$$d_{ik} = d(x^i, a^k) + \omega_{ik} \tag{5}$$

In [6] two different approaches were considered to take into account noise. In the first approach it is assumed $\omega_{ij} \sim N(0, \sigma_{ij}^2)$ and $\omega_{ik} \sim N(0, \sigma_{ij}^2)$, where $N(0, \sigma^2)$ is a random variable with mean zero and variance σ^2, and they are independent.

In the second approach, a distance feasibility problem with upper and lower bounds is solved. In this paper we will consider the second approach to take into account noise in the distance measurements. Then noisy distance measures may be represented in a confidence interval form of a lower bound $\underline{d}_{ij}$ and an upper bound $\overline{d}_{ij}$ between sensors x^i and x^j, or lower bound $\underline{d}_{ik}$ and upper bound $\overline{d}_{ik}$ between a sensor x^i and an anchor a^k. In this case we reformulate the equations (1) and (2) as follows:

$$\underline{d}_{ij}^2 \leq \|x^i - x^j\|^2 \leq \overline{d}_{ij}^2, \forall (i, j) \in N_x, \tag{6}$$

$$\underline{d}_{ik}^2 \leq \|x^i - a^k\|^2 \leq \overline{d}_{ik}^2, \forall (i, k) \in N_a. \tag{7}$$

Then the objective function in presence of noise is as follows:

$$f(x^1, \ldots, x^n) = \sum_{(i,j)\in N_x} \max\left(0, -\|x^i - x^j\|^2 + \underline{d}_{ij}^2, \|x^i - x^j\|^2 - \overline{d}_{ij}^2\right) + \sum_{(i,k)\in N_a} \max\left(0, -\|x^i - a^k\|^2 + \underline{d}_{ik}^2, \|x^i - a^k\|^2 - \overline{d}_{ik}^2\right). \tag{8}$$

The problem of localization in the presence of noise in distance measurements is reduced to the following optimization problem:

$$\text{minimize} \quad f(x^1, \ldots, x^n) \quad \text{s.t.} \quad x^i \in \mathbb{R}^2. \tag{9}$$

Unlike Problem (3) the objective function in Problem (9) is nonconvex and nonsmooth. Such problems can be studied by applying the Clarke generalized gradients (see [9]).

In the next section we observe that the objective function in the optimization problem is semismooth, quasidifferentiable and piecewise partially separable. The use of these properties allow us to design an efficient algorithm for approximation of subgradients of the objective function and to apply the discrete gradient method for its minimization.

3. Properties of the objective function

In this section we describe some properties of the objective functions in Problems (3) and (9). Since Problem (9) reflects a more realistic situation we choose to concentrate on it. First we recall the definitions of the Clarke subdifferential, quasidifferential, semismooth and piecewise partially separable functions from nonsmooth analysis.

A function f, defined on $\mathbb{R}^n$, is called locally Lipschitz continuous if for any bounded subset $X \subset \mathbb{R}^n$ there exists an $R > 0$ such that

$$|f(x) - f(y)| \leq R\|x - y\| \quad \forall x, y \in X.$$

Clarke introduced generalized gradients for Lipschitz functions [9]. Since a locally Lipschitz function f is differentiable almost everywhere we can define for it a subdifferential by

$$\partial f(x) = \text{co}\,\{v \in \mathbb{R}^n : \exists (x^k \in D(f)) : \; x = \lim_{k\to\infty} x^k \text{ and } v = \lim_{k\to\infty} \nabla f(x^k)\},$$

here $D(f)$ denotes the set where f is differentiable, co denotes the convex hull of a set. The mapping $\partial f(x)$ is upper semicontinuous and bounded on bounded sets [9]. The generalized directional derivative of f at x in the direction g is defined as

$$f^0(x, g) = \limsup_{y\to x, \alpha\downarrow 0} \alpha^{-1}[f(y + \alpha g) - f(y)].$$

For the locally Lipschitz function f the generalized directional derivative exists and $f^0(x, g) = \max\{\langle v, g\rangle : v \in \partial f(x)\}$. f is called a Clarke regular function on $\mathbb{R}^n$, if it is directionally differentiable and $f'(x, g) = f^0(x, g)$ for all $x, g \in \mathbb{R}^n$, where $f'(x, g)$ is a derivative of the function f at the point x in the direction g:

$$f'(x, g) = \lim_{\alpha\downarrow 0} \alpha^{-1}[f(x + \alpha g) - f(x)].$$

Let f be a locally Lipschitz continuous function defined on $\mathbb{R}^n$. For a point x to be a local minimizer of the function f on $\mathbb{R}^n$, it is necessary that $0 \in \partial f(x)$.

A function $f : \mathbb{R}^n \to \mathbb{R}^1$ is called semismooth at $x \in \mathbb{R}^n$, if it is locally Lipschitz at x and for each $g \in \mathbb{R}^n$ and for any sequences $\{t_k\} \subset \mathbb{R}^1, \; \{g^k\} \subset \mathbb{R}^n, \; \{v^k\} \subset \mathbb{R}^n$ such that $t_k \downarrow 0, \; g^k \to g, \; v^k \in \partial f(x + t_k g^k)$, the limit

$$\lim_{k\to\infty} \langle v^k, g\rangle$$

exists [14]. The semismooth function f is directionally differentiable and

$$f'(x, g) = \lim_{k\to\infty} \langle v^k, g\rangle, \quad v^k \in \partial f(x + t_k g^k).$$

A function f is called quasidifferentiable at a point x, if it is locally Lipschitz continuous, directionally differentiable at this point and there exist convex, compact sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ such that:

$$f'(x, g) = \max\{\langle u, g\rangle : u \in \underline{\partial} f(x)\} + \min\{\langle v, g\rangle : v \in \overline{\partial} f(x)\}$$

The set $\underline{\partial} f(x)$ is called a subdifferential, the set $\overline{\partial} f(x)$ a superdifferential and the pair $[\underline{\partial} f(x), \overline{\partial} f(x)]$ a quasidifferential of the function f at a point x [11].

The function f is called a partially separable if there exists a family of $n \times n$ diagonal matrices $Q_i, \; i = 1, \ldots, M$ such that the function f can be represented as follows:

$$f(x) = \sum_{i=1}^{M} f_i(Q_i x).$$

We assume that the matrices Q_i are binary, that is they contain only 0 and 1 and the number of non-zero elements in the diagonal of the matrix Q_i is much smaller than n. In other terms, the function f is called partially separable if it can be represented as the sum of functions of a much smaller number of variables. If $M = n$ and $diag(Q_i) = e_i$ where e_i is the $i - th$ orth vector, then the function f is separable.

The function f is said to be piecewise partially separable if there exists a finite family of closed sets $D_1, \ldots, D_m$ such that $\bigcup_{i=1}^{m} D_i = \mathbb{R}^n$ and the function f is partially separable on each set $D_i, \; i = 1, \ldots, m$ (see [3]).

Now we can describe some of properties of the objective function in Problem (9).

It is clear that the function f is locally Lipschitz continuous.

Proposition 1: The function F is quasidifferentiable and its subdifferential and superdifferential are polytopes.

Proof: Consider the function

$$\varphi_{ij}(x^i, x^j) = \max\left(0, -\|x^i - x^j\|^2 + \underline{d}_{ij}^2, \|x^i - x^j\|^2 - \overline{d}_{ij}^2\right).$$

One of functions under maximum is constant and two others are quadratic functions. Both quadratic functions are not convex, however it follows from Proposition 3.2 in [21] that they are d.c. functions that is they can be represented as a difference of two smooth convex quadratic functions. Since the maximum of d.c. functions is again d.c. function (see [21]) then the function φ_{ij} is d.c., therefore it is quasidifferentiable and its subdifferential and superdifferential are polytopes.

The function

$$\theta_{ik}(x^i, x^j) = \max\left(0, -\|x^i - a^k\|^2 + \underline{d}_{ik}^2, \|x^i - a^k\|^2 - \overline{d}_{ik}^2\right)$$

is a maximum of constant function, one concave and convex functions. All three functions are smooth. Therefore the function θ_{ik} is d.c. function, it is quasidifferentiable and its sub and superdifferential are polytopes.

Since the sum of d.c. functions is again d.c. function then one can see the function f is d.c., it is quasidifferentiable and its sub and superdifferential are polytopes. △

Proposition 2: The function f is semismooth.

Proof: The proof follows from the facts that distance functions are smooth and consequently they are semismooth, maximum of semismooth functions is also semismooth and finally, the sum of semismooth functions is also semismooth. △

Proposition 3: The function f is piecewise partially separable.

Proof: The distance functions by their definition are separable. Maximum of separable functions is piecewise separable and finally sum of piecewise separable functions is piecewise partially separable (see [3]). △

The function f is not regular and Clarke calculus for such functions exists in the form of inclusions and such calculus cannot be used to estimate subgradients. Therefore the computation of subgradients of such functions is quite difficult task. In the next section we consider one scheme to approximate subgradients of the function f.

4. Approximation of subgradients

In this section a scheme to approximate subgradients of the function f is described. This approach is introduced in [4], [5]. All necessary proofs also can be found in these papers.

We consider a function f defined on $\mathbb{R}^n$ and assume that this function is quasidifferentiable. We also assume that both sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ are polytopes at any $x \in \mathbb{R}^n$. We denote by Φ the class of all semismooth, quasidifferentiable functions defined on $\mathbb{R}^n$, whose subdifferential and superdifferential are polytopes at any $x \in \mathbb{R}^n$. Results from the previous section show that the objective function f in Problem (9) belongs to this class.

Let $G = \{e \in \mathbb{R}^n : e = (e_1, \ldots, e_n), |e_j| = 1, \ j = 1, \ldots, n\}$ be a set of all vertices of the unit hypercube in $\mathbb{R}^n$. We take $e \in G$ and consider the sequence of n vectors $e^j = e^j(\alpha), \ j = 1, \ldots, n$ with $\alpha \in (0, 1]$:

$$\begin{array}{lcl} e^1 & = & (\alpha e_1, 0, \ldots, 0), \\ e^2 & = & (\alpha e_1, \alpha^2 e_2, 0, \ldots, 0), \\ \ldots & = & \ldots\ldots\ldots \\ e^n & = & (\alpha e_1, \alpha^2 e_2, \ldots, \alpha^n e_n). \end{array}$$

Let $e \in G$ be a given vector and $\lambda > 0, \ \alpha > 0$ be given numbers. Consider the following points

$$x^0 = x, \quad x^j = x^0 + \lambda e^j(\alpha), \quad j = 1, \ldots, n.$$

It is clear that

$$x^j = x^{j-1} + (0, \ldots, 0, \lambda \alpha^j e_j, 0, \ldots, 0), \quad j = 1, \ldots, n.$$

Let $v = v(\alpha, \lambda) \in \mathbb{R}^n$ be a vector with the following coordinates:

$$v_j = (\lambda \alpha^j e_j)^{-1} \left[f(x^j) - f(x^{j-1}) \right], \quad j = 1, \ldots, n. \quad (10)$$

For any fixed $e \in G$ and $\alpha > 0$ we introduce the set:

$$V(e, \alpha) = \{w \in \mathbb{R}^n : \ \exists (\lambda_k \to +0, \ k \to +\infty), \\ w = \lim_{k \to +\infty} v(\alpha, \lambda_k)\}.$$

Proposition 4: [4], [5]. Assume that $f \in \Phi$. Then there exists $\alpha_0 > 0$ such that

$$V(e, \alpha) \subset \partial f(x), \ \forall \ \alpha \in (0, \alpha_0].$$

Remark 1: It follows from Proposition 4 that in order to approximate subgradients of the function f one can choose a vector $e \in G$, sufficiently small $\alpha > 0, \ \lambda > 0$ and apply (10) to compute a vector $v(\alpha, \lambda)$. This vector is an approximation to a subgradient.

A. Computation of subdifferentials

Now we can describe an algorithm for the computation of subdifferentials. This algorithm is based on the notion of a discrete gradient. We start with the definition of the discrete gradient, which was introduced in [2].

Let f be a locally Lipschitz continuous function defined on $\mathbb{R}^n$. Let

$$S_1 = \{g \in \mathbb{R}^n : \|g\| = 1\},$$

$$P = \{z : \mathbb{R}^+ \to \mathbb{R}^+, \ \beta^{-1} z(\beta) \downarrow 0, \ \beta \downarrow 0\}.$$

Here S_1 is the unit sphere and P is the set of univariate positive infinitesimal functions. We take any $g \in S_1, \ e \in G$ and a positive number $\alpha \in (0, 1]$. Then we define $|g_i| = \max\{|g_k|, \quad k = 1, \ldots, n\}$ and the sequence of n vectors $e^j(\alpha), \ j = 1, \ldots, n$. For given $x \in \mathbb{R}^n$ and $z \in P$ consider a sequence of $n + 1$ points:

$$\begin{array}{lll} x^0 = & x + & \lambda g, \\ x^1 = & x^0 + & z(\lambda) e^1(\alpha), \\ \ldots = & \ldots & \ldots \\ x^n = & x^0 + & z(\lambda) e^n(\alpha). \end{array}$$

Definition 1: [2] The discrete gradient of the function f at the point $x \in \mathbb{R}^n$ is the vector $\Gamma(x, g, e, z, \lambda, \alpha) = (\Gamma_1, \ldots, \Gamma_n) \in \mathbb{R}^n, g \in S_1$ with the following coordinates:

$$\Gamma_j = [z(\lambda) \alpha^j e_j)]^{-1} \left[f(x^j) - f(x^{j-1}) \right], \ j = 1, \ldots, n, \ j \neq i,$$

$$\Gamma_i = (\lambda g_i)^{-1} \left[f(x + \lambda g) - f(x) - \lambda \sum_{j=1, j \neq i}^{n} \Gamma_j g_j \right].$$

It follows from Definition 1 that

$$f(x + \lambda g) - f(x) = \lambda \langle \Gamma(x, g, e, z, \lambda, \alpha), g \rangle \quad (11)$$

for all $g \in S_1, \ e \in G, \ z \in P, \ \lambda > 0, \ \alpha > 0$.

Remark 2: One can see that the discrete gradient is defined with respect to a given direction $g \in S_1$ and in order to compute it, first we define a sequence of points $x^0, \ldots, x^n$ and compute the values of the function f at these points that is we compute $n+2$ values of this function including the point x. $n-1$ coordinates of the discrete gradient are defined similar to those of the vector $v(\alpha, \lambda)$ and i-th coordinate is defined so that to satisfy the equality (11) which can be considered as some version of the mean value theorem.

Remark 3: Since the objective function f in Problem (9) is piecewise partially separable we will use a special scheme described in [3] to compute its discrete gradients. This scheme allows us to use only two evaluations instead of $n + 2$ evaluations of the objective function f in Problem (9) to compute one discrete gradient. Such an approach allow us to apply the above described algorithm to functions with large number of variables.

For a given $\alpha > 0$ we define the following set:

$$B(x, \alpha) = \{v \in \mathbb{R}^n : \ \exists (g \in S_1, \ e \in G, \ z_k \in P, \ \lambda_k \in \mathbb{R}^1) : \\ z_k \downarrow 0, \ \lambda_k \downarrow 0, \ k \to +\infty$$

and $v = \lim_{k\to+\infty} \Gamma(x, g, e, z_k, \lambda_k, \alpha)\}$. (12)

Proposition 5: [4], [5] Assume that $f \in \Phi$. Then there exists $\alpha_0 > 0$ such that

$$\text{co}\, B(x, \alpha) \subset \partial f(x),\ \forall\ \alpha \in (0, \alpha_0].$$

Remark 4: Proposition 5 shows that one can use discrete gradients to approximate the Clarke subdifferentials.

5. The discrete gradient method

In this section we describe the discrete gradient method for solving Problem (9). An important step in this method is the computation of descent directions. Therefore we start with the description of an algorithm for finding descent directions.

A. Computation of descent directions

Let $z \in P, \lambda > 0, \alpha \in (0, 1]$, the number $c \in (0, 1)$ and a tolerance $\delta > 0$ be given.

Algorithm 1: An algorithm for the computation of the descent direction.

Step 1. Choose any $g^1 \in S_1, e \in G$, compute $i = \text{argmax}\,\{|g_j|,\ j = 1, \ldots, n\}$ and a discrete gradient $v^1 = \Gamma^i(x, g^1, e, z, \lambda, \alpha)$. Set $\overline{D}_1(x) = \{v^1\}$ and $k = 1$.

Step 2. Compute the vector $\|w^k\|^2 = \min\{\|w\|^2 : w \in \overline{D}_k(x)\}$. If

$$\|w^k\| \le \delta, \tag{13}$$

then stop. Otherwise go to Step 3.

Step 3. Compute the search direction by $g^{k+1} = -\|w^k\|^{-1} w^k$.

Step 4. If

$$f(x + \lambda g^{k+1}) - f(x) \le -c\lambda\|w^k\|, \tag{14}$$

then stop. Otherwise go to Step 5.

Step 5. Compute $i = \text{argmax}\,\{|g_j^{k+1}| :\ j = 1, \ldots, n\}$ and a discrete gradient

$$v^{k+1} = \Gamma^i(x, g^{k+1}, e, z, \lambda, \alpha),$$

construct the set $\overline{D}_{k+1}(x) = \text{co}\,\{\overline{D}_k(x) \bigcup \{v^{k+1}\}\}$, set $k = k + 1$ and go to Step 2.

Algorithm 1 terminates after a finite number of iterations [5].

B. The method

Let sequences $\delta_k > 0,\ z_k \in P,\ \lambda_k > 0,\ \delta_k \to +0,\ z_k \to +0,\ \lambda_k \to +0,\ k \to +\infty$, sufficiently small number $\alpha > 0$ and numbers $c_1 \in (0, 1), c_2 \in (0, c_1]$ be given.

Algorithm 2: The discrete gradient method

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Set $s = 0$ and $x_s^k = x^k$.

Step 3. Apply Algorithm 1 for the computation of the descent direction at $x = x_s^k, \delta = \delta_k, z = z_k, \lambda = \lambda_k, c = c_1$. This algorithm terminates after a finite number of iterations $l > 0$. As a result we get the set $\overline{D}_l(x_s^k)$ and an element v_s^k such that

$$\|v_s^k\|^2 = \min\{\|v\|^2 :\ v \in \overline{D}_l(x_s^k)\}.$$

Furthermore either $\|v_s^k\| \le \delta_k$ or for the search direction $g_s^k = -\|v_s^k\|^{-1} v_s^k$

$$f(x_s^k + \lambda_k g_s^k) - f(x_s^k) \le -c_1\lambda_k\|v_s^k\|. \tag{15}$$

Step 4. If

$$\|v_s^k\| \le \delta_k \tag{16}$$

then set $x^{k+1} = x_s^k, k = k + 1$ and go to Step 2. Otherwise go to Step 5.

Step 5. Construct the following iteration $x_{s+1}^k = x_s^k + \sigma_s g_s^k$, where σ_s is defined as follows

$$\sigma_s = \text{argmax}\, \left\{\sigma \ge 0 :\ f(x_s^k + \sigma g_s^k) - f(x_s^k) \le -c_2\sigma\|v_s^k\|\right\}.$$

Step 6. Set $s = s + 1$ and go to Step 3.

Remark 5: The discrete gradient method can be applied to solve both Problems (3) and (9). The discrete gradients contain three parameters and the parameter $\lambda > 0$ is most important among them. Large values of $\lambda > 0$ allows us to find descent directions from local minimizers. Therefore, the discrete gradient is capable to escape from shallow local minimizers and to find near global solution.

6. Conclusion

In this paper the wireless sensor network localization problem is formulated as an unconstrained nonsmooth optimization problem. Such a formulation allows us to reduce the number of variables in an optimization problem and design efficient algorithm for its solution. We proposed to apply the derivative free discrete gradient method for solving the wireless sensor network localization problem.

Acknowledgement

This research was supported by the Australian Research Council. Dr. Adil Bagirov is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

References

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on Sensor Networks. *IEEE Communications Magazine*, 1:102–114, 2002.

[2] A. M. Bagirov. Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices. *In: A. Eberhard et al. (eds.) Progress in Optimization: Contribution from Australia, Kluwer Academic Publishers*, pages 147–175, 1999.

[3] A. M. Bagirov, and J. Ugon. Piecewise partially separable functions and a derivative-free method for large-scale nonsmooth optimization. *Journal of Global Optimization* 35:163-195, 2006.

[4] A. M. Bagirov, M. Ghosh and D. Webb. A derivative-free method for linearly constrained nonsmooth optimization, *Journal of Industrial and Management Optimization,* 2(3): 319-338, 2006.

[5] A. M. Bagirov, B. Karasozen and M. Sezer. Discrete gradient method: a derivative free method for nonsmooth optimization, *Journal of Optimization Theory and Applications,* accepted for publication.

[6] P. Biswas, T.C. Lian, T.C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–220, 2006.
[7] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor nework localization. *Proceedings of the 3rd IPSN*, pages 46–54, 2006.
[8] Chee-Yee Chong and Srikanta P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.
[9] F. H. Clarke. *Optimization and Nonsmooth Analysis,* New York: John Wiley, 1983.
[10] J.A. Costa, N. Patwari, and A. Hero. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):39–64, 2006.
[11] V. F. Demyanov and A. M. Rubinov. *Constructive Nonsmooth Analysis.* Peter Lang, Frankfurt am Main, 1995.
[12] L. Doherty, L.E. Ghaoui, and K.S.J. Pister. Convex position estimation in wireless sensor networks. *in Proceedings of IEEE Infocom, Ancorage, Alaska,*, pages 1655–1663, 2001.
[13] A. Ihler, J. Fisher, R. Moses, and A. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):809–819, 2005.
[14] R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization* 15:957-972, 1977.
[15] X. Nguyen, M. Jordan, and B. Sinopoli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.
[16] D. Nichulescu and B. Nath. Ad hoc positioning system (APS). *GLOBECOM*, pages 2926–2931, 2001.
[17] N. Patwari, J.N. Ash, S. Kyperountas, III Hero, A.O., R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54– 69, 2005.
[18] C. Savarese, J.M. Rabaey, and J. Beutel. Location in distributed ad-hoc wireless sensor networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:2037–2040, 2001.
[19] Y. Shang and W. Ruml. Improved MDS-based localization. *In Proceedings of the 23rd Conference of the IEEE Communications Society (Infocom2004)*, pages 2640–2651, 2001.
[20] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212, New York, NY, USA, 2003. ACM Press.
[21] H. Tuy. *Convex Analysis and Global Optimization,* Kluwer Academic Publishers, Dordrecht, Boston, 1999.
[22] X. Zhang and S. B. Wicker. On the optimal distribution of sensors in a random field. *ACM Transactions on Sensor Networks*, 1(2):301–306, 2005.