Iterative Minimization Schemes for Solving the Single Source Localization Problem *

Amir Beck† Marc Teboulle‡ and Zahar Chikishev§

July 22, 2007

Abstract

We consider the problem of locating a single radiating source from several noisy measurements using a maximum likelihood (ML) criteria. The resulting optimization problem is nonconvex and nonsmooth and thus finding its global solution is in principal a hard task. Exploiting the special structure of the objective function, we introduce and analyze two iterative schemes for solving this problem. The first algorithm is a very simple explicit fixed-point-based formula, and the second is based on solving at each iteration a nonlinear least squares problem which can be solved globally and efficiently after transforming it into and equivalent quadratic minimization problem with a single quadratic constraint. We show that the nonsmoothness of the problem can be avoided by choosing a specific "good" starting point for both algorithms, and we prove the convergence of the two schemes to stationary points. We present empirical results that support the underlying theoretical analysis and suggest that despite of its nonconvexity, the ML problem can effectively be solved globally using the devised schemes.

Key words: Single source location problem, Weiszfield algorithm, nonsmooth and nonconvex minimization, fixed point methods, nonlinear least squares, generalized trust region, semidefinite relaxation.

*This research was partially supported by the Israel Science Foundation under Grant ISF No.489/06

†Department of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: becka@ie.technion.ac.il.

‡School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: teboulle@post.tau.ac.il.

§Department of Mathematics, Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: nosound@tx.technion.ac.il.

1 Introduction

1.1 The Source Localization Problem

Consider the problem of locating a single radiating source from noisy range measurements collected using a network of passive sensors. More precisely, consider an array of m sensors, and let $\mathbf{a}_j \in \mathbb{R}^n$ denote the coordinates of the jth sensor[1]. Let $\mathbf{x} \in \mathbb{R}^n$ denote the unknown source's coordinate vector, and let $d_j > 0$ be a noisy observation of the range between the source and the jth sensor:

$$d_j = \|\mathbf{x} - \mathbf{a}_j\| + \varepsilon_j, \quad j = 1, \ldots, m, \tag{1.1}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_m)^T$ denotes the unknown noise vector. Such observations can be obtained for example from the time-of-arrival (TOA) measurements in a constant-velocity propagation medium. The source localization problem is the following:

The Source Localization Problem: Given the observed range measurements $d_j > 0$, find a "good" approximation of the source $\mathbf{x}$ satisfying the equations in (1.1).

The source localization problem has received significant attention in the signal processing literature and specifically in the field of mobile phones localization [12, 5, 13]. It is also worth mentioning that the interest in wireless localization problems have increased since the first ruling of the Federal Communications Commission for detection of emergency calls in the United States in 1996[2]. Currently, a high percentage of E911 calls originate from mobile phones. Due to the unknown location of the wireless E911 calls, these calls do not receive the same quality of emergency assistance that fixed-network 911 calls enjoy. To deal with this problem, the FCC issued an order on 12 July 1996, requiring all wireless service providers to report accurate mobile station (MS) location information to the E911 operator.

In addition to emergency management, mobile position information is also useful in mobile advertising, asset tracking, fleet management, location-sensitive billing, [12] interactive map consultation, and monitoring of the mentally impaired [5].

1.2 The Maximum Likelihood Criteria

In this paper we adopt the maximum-likelihood approach for solving the source localization problem (1.1), see e.g., [4]. When $\boldsymbol{\varepsilon}$ follows a Gaussian distribution with a covariance matrix proportional to the identity matrix, the source $\mathbf{x}$ is the maximum likelihood (ML) estimate that is solution of the problem:

$$\text{(ML):} \quad \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) \equiv \sum_{j=1}^{m} (\|\mathbf{x} - \mathbf{a}_j\| - d_j)^2 \right\}. \tag{1.2}$$

Note that in addition to the statistical interpretation, the latter problem is a least squares problem, in the sense that it minimizes the squared sum of the errors.

[1] in practical applications $n = 2$ or 3.

[2] See http://www.fcc.gov/911/enhanced/

An alternative approach for estimating the source location $\mathbf{x}$ is by solving the following least squares (LS) problem in the squared domain:

$$\text{(LS):} \quad \min_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{m} (\|\mathbf{x}-\mathbf{a}_j\|^2 - d_j^2)^2. \tag{1.3}$$

Despite of its nonconvexity, the LS problem can be solved globally and efficiently by transforming it into a problem of minimizing a quadratic function subject to a single quadratic constraint [1] (more details will be given in Section 3.2). However, the LS approach has two major disadvantages compared to the ML approach: First, the LS formulation lacks the statistical interpretation of the ML problem. Second, as demonstrated by the numerical simulations in Section 4, the LS estimate provides less accurate solutions than those provided by the the ML approach.

The ML problem, like the LS problem, is nonconvex. However, as opposed to the LS problem for which a global solution can be computed efficiently [1], the ML problem seems to be a difficult problem to solve efficiently. A possible reason for the increased difficulty of the ML problem is its nonsmoothness. One approach for approximating the solution of the ML problem is via semidefinite relaxation (SDR) [4, 1]. We also note that the source localization problem formulated as (ML) can be viewed as a special instance of sensor network localization problems in which several sources are present, see for example the recent work [3]; for this class of problems, semidefinite programming based algorithms have been developed.

In this paper we depart from the SDR techniques and seek other efficient approaches to solve the ML problem. This is achieved by exploiting the special structure of the objective function which allows us to devise fixed-point based iterative schemes for solving the nonsmooth and nonconvex ML problem (1.2). The first scheme admits a very simple explicit iteration formula given by

$$\mathbf{x}^{k+1} = \mathcal{M}_1(\mathbf{x}^k, \mathbf{a}), \ \text{(where } \mathbf{a} \equiv (\mathbf{a}_1, \ldots, \mathbf{a}_m)\text{)},$$

while the second iterative scheme is of the form

$$\mathbf{x}^{k+1} \in \operatorname*{argmin}_{\mathbf{x}} \mathcal{M}_2(\mathbf{x}, \mathbf{x}^k, \mathbf{a}),$$

and requires the solution of an additional subproblem which will be shown to be efficiently solved. The main goals of this paper are to introduce the building mechanism of these two schemes, to develop and analyze their convergence properties, and to demonstrate their computational viability for solving the ML problem (1.2), as well as their effectiveness when compared with the LS and SDR approaches.

1.3 Paper Layout

In the next section, we present and analyze the first scheme which is a simple fixed point based method. The second algorithm, which is based on solving a sequence of least squares problems of a similar structure to that of (1.3), is presented and analyzed in Section 3.

The construction of both methods is motivated by two different interpretations of the well known Weiszfeld method for the Fermat-Weber location problem [16]. For both schemes, we show that the nonsmoothness of the problem can be avoided by choosing a specific "good" starting point. Empirical results presented in Section 4 provide a comparison between the two devised algorithms, as well as comparison to different approaches such as LS and SDR. In particular, the numerical results suggest that despite of its nonconvexity, the ML problem can, for all practical purposes, be globally solved using the devised methods.

1.4 Notation

Throughout the paper, the following notation is used: vectors are denoted by boldface lowercase letters, *e.g.,* $\mathbf{y}$, and matrices by boldface uppercase letters *e.g.,* $\mathbf{A}$. The ith component of a vector $\mathbf{y}$ is written as y_i. Given two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \succ \mathbf{B}$ ($\mathbf{A} \succeq \mathbf{B}$) means that $\mathbf{A}-\mathbf{B}$ is positive definite (semidefinite). The directional derivative of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $\bar{\mathbf{x}}$ in the direction $\mathbf{v}$ is defined (if it exists) by

$$f'(\mathbf{x};\mathbf{v}) \equiv \lim_{t\to 0^+} \frac{f(\bar{\mathbf{x}}+t\mathbf{v})-f(\bar{\mathbf{x}})}{t}. \tag{1.4}$$

The α-level set of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined by $\mathrm{Lev}(f,\alpha) = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \le \alpha\}$. The collection of m sensors $\{\mathbf{a}_1,\ldots,\mathbf{a}_m\}$ is denoted by $\mathcal{A}$.

2 A Simple Fixed Point Algorithm

In this section we introduce a simple fixed point algorithm that is designed to solve the ML problem (1.2). The algorithm is inspired by the celebrated Weiszfeld algorithm for the Fermat-Weber problem which is briefly recalled in Section 2.1. In Section 2.2 we introduce and analyze the fixed point scheme designed to solve the ML problem.

2.1 A Small Detour: Weiszfeld Algorithm for the Fermat-Weber Problem

As was already mentioned, the ML problem (1.2) is nonconvex and nonsmooth and thus finding its exact solution is in principle a difficult task. We propose a fixed point scheme motivated by the celebrated Weiszfeld algorithm [16, 7] for solving the Fermat-Weber location problem:

$$\min_{\mathbf{x}} \left\{ s(\mathbf{x}) \equiv \sum_{j=1}^{m} \omega_j \|\mathbf{x}-\mathbf{a}_j\| \right\}, \tag{2.1}$$

where $\omega_j > 0$ and $\mathbf{a}_j \in \mathbb{R}^n$ for $j = 1,\ldots,m$. Of course, the Fermat-Weber problem is much easier to analyze and solve than the ML problem (1.2) since it is a well structured nonsmooth convex minimization problem. This problem has been extensively studied in the location theory literature, see for instance [11]. Our objective here is to mimic Weiszfeld algorithm [16] to obtain an algorithm for solving the nonsmooth and nonconvex ML problem

(1.2). Weiszfeld method is a very simple fixed point scheme that is designed to solve the Fermat-Weber problem. One way to derive it is to write the first order global optimality conditions for the convex problem (2.1):

$$\nabla s(\mathbf{x}) = \sum_{j=1}^{m} \omega_j \frac{\mathbf{x}-\mathbf{a}_j}{\|\mathbf{x}-\mathbf{a}_j\|} = 0, \ \forall \mathbf{x} \notin \mathcal{A},$$

that can be written as:

$$\mathbf{x} = \frac{\sum_{j=1}^{m} \omega_j \frac{\mathbf{a}_j}{\|\mathbf{x}-\mathbf{a}_j\|}}{\sum_{j=1}^{m} \frac{\omega_j}{\|\mathbf{x}-\mathbf{a}_j\|}},$$

which naturally calls for the iterative scheme:

$$\mathbf{x}^{k+1} = \frac{\sum_{j=1}^{m} \omega_j \frac{\mathbf{a}_j}{\|\mathbf{x}^k-\mathbf{a}_j\|}}{\sum_{j=1}^{m} \frac{\omega_j}{\|\mathbf{x}^k-\mathbf{a}_j\|}}. \tag{2.2}$$

For convergence analysis of the Weiszfeld algorithm (2.2) and modified versions of the algorithm, see e.g., [10, 15] and references therein.

2.2 The Simple Fixed Point Algorithm: Definition and Analysis

Similarly to Weiszfeld method, our starting point for constructing a fixed point algorithm to solve the ML problem is by writing the optimality conditions. Assuming that $\mathbf{x} \notin \mathcal{A}$ we have that $\mathbf{x}$ is a stationary point for problem (ML) if and only if

$$\nabla f(\mathbf{x}) = 2\sum_{j=1}^{m}(\|\mathbf{x}-\mathbf{a}_j\| - d_j)\frac{\mathbf{x}-\mathbf{a}_j}{\|\mathbf{x}-\mathbf{a}_j\|} = \mathbf{0}, \tag{2.3}$$

which can be written as

$$\mathbf{x} = \frac{1}{m}\left\{\sum_{j=1}^{m}\mathbf{a}_j + \sum_{j=1}^{m} d_j \frac{\mathbf{x}-\mathbf{a}_j}{\|\mathbf{x}-\mathbf{a}_j\|}\right\}.$$

The latter relation calls for the following fixed point algorithm which we term the *standard fixed point (SFP) scheme*:

Algorithm SFP:

$$\mathbf{x}^{k+1} = \frac{1}{m}\left\{\sum_{j=1}^{m}\mathbf{a}_j + \sum_{j=1}^{m} d_j \frac{\mathbf{x}^k-\mathbf{a}_j}{\|\mathbf{x}^k-\mathbf{a}_j\|}\right\}, \quad k \geq 0. \tag{2.4}$$

Like in the Weiszfeld algorithm, the SFP scheme is not well defined if $\mathbf{x}^k \in \mathcal{A}$ for some k. In the sequel we will show that by carefully selecting the initial vector $\mathbf{x}^0$ we can *guarantee* that the iterates are not in the sensors set $\mathcal{A}$, therefore establishing the well definiteness of the method. At this juncture, it is interesting to notice that the approach we suggest here for dealing with the points of nonsmoothness that occur at $\mathbf{x}^k \in \mathcal{A}$ is quite different from the common approaches handling the nonsmoothness. For example, in order to avoid the nondifferentiable points of the Fermat-Weber objective function, several modifications of Weiszfeld method were proposed, see e.g., [10, 15] and references therein. However, there does not seem to have been any attempts in the literature to choose "good" initial starting points to avoid the nonsmoothness difficulty. A constructive procedure for choosing a good starting point for the SFP method will be given at the end of this section.

Before proceeding with the analysis of the SFP method, we record the fact that, much like the Weiszfeld algorithm ([7]), the scheme SFP is a gradient method with a fixed step size.

Proposition 2.1 *Let* $\{\mathbf{x}^k\}$ *be the sequence generated by the SFP method (2.4) and suppose that* $\mathbf{x}^k \notin \mathcal{A}$ *for all* $k \geq 0$. *Then*

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{1}{2m}\nabla f(\mathbf{x}^k). \tag{2.5}$$

Proof. Follows by a straightforward calculation, using the gradient of f computed in (2.3). □

A gradient method does not necessarily converge without additional assumptions (e.g., assuming that ∇f is Lipschitz continuous and/or using a line search, [2]). Nevertheless, we show below that scheme (2.4) *does* converge.

By Proposition 2.1 the SFP method can be compactly written as

$$\mathbf{x}^{k+1} = T(\mathbf{x}^k), \tag{2.6}$$

where $T : \mathbb{R}^n \setminus \mathcal{A} \to \mathbb{R}^n$ is the operator defined by

$$T(\mathbf{x}) = \mathbf{x} - \frac{1}{2m}\nabla f(\mathbf{x}). \tag{2.7}$$

In the convergence analysis of the SFP method we will also make use of the auxiliary function:

$$h(\mathbf{x}, \mathbf{y}) \equiv \sum_{j=1}^{m} \|\mathbf{x} - \mathbf{a}_j - d_j r_j(\mathbf{y})\|^2, \quad \forall \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n \setminus \mathcal{A}, \tag{2.8}$$

where

$$r_j(\mathbf{y}) \equiv \frac{\mathbf{y} - \mathbf{a}_j}{\|\mathbf{y} - \mathbf{a}_j\|}, \quad j = 1, \ldots, m.$$

Note that for every $\mathbf{y} \notin \mathcal{A}$, the following relations hold for every $j = 1, \ldots, m$:

$$\|r_j(\mathbf{y})\| = 1, \tag{2.9}$$

$$(\mathbf{y} - \mathbf{a}_j)^T r_j(\mathbf{y}) = \|\mathbf{y} - \mathbf{a}_j\|. \tag{2.10}$$

In Lemma 2.1 below, we prove several key properties of the auxiliary function h defined in (2.8).

Lemma 2.1 *(a)* $h(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$ *for every* $\mathbf{x} \notin \mathcal{A}$.

(b) $h(\mathbf{x},\mathbf{y}) \geq f(\mathbf{x})$ *for every* $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n \setminus \mathcal{A}$.

(c) If $\mathbf{y} \notin \mathcal{A}$ *then*

$$T(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} h(\mathbf{x},\mathbf{y}). \tag{2.11}$$

Proof. (a) For every $\mathbf{x} \notin \mathcal{A}$,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^{m} (\|\mathbf{x}-\mathbf{a}_j\| - d_j)^2 \\ &= \sum_{j=1}^{m} (\|\mathbf{x}-\mathbf{a}_j\|^2 - 2d_j\|\mathbf{x}-\mathbf{a}_j\| + d_j^2) \\ &\overset{(2.9),(2.10)}{=} \sum_{j=1}^{m} (\|\mathbf{x}-\mathbf{a}_j\|^2 - 2d_j(\mathbf{x}-\mathbf{a}_j)^T r_j(\mathbf{x}) + d_j^2\|r_j(\mathbf{x})\|^2) = h(\mathbf{x},\mathbf{x}), \end{aligned}$$

where the last equation follows from (2.8).
(b) Using the definition of f and h given respectively in (1.2),(2.8), and the fact (2.9), a short computation shows that for every $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n \setminus \mathcal{A}$,

$$\begin{aligned} h(\mathbf{x},\mathbf{y}) - f(\mathbf{x}) &= 2\sum_{j=1}^{m} d_j \left(\|\mathbf{x}-\mathbf{a}_j\| - (\mathbf{x}-\mathbf{a}_j)^T r_j(\mathbf{y})\right) \\ &\geq 0, \end{aligned}$$

where the last inequality follows from Cauchy-Schwartz inequality and using again (2.9).
(c) For any $\mathbf{y} \in \mathbb{R}^n \setminus \mathcal{A}$, the function $\mathbf{x} \mapsto h(\mathbf{x},\mathbf{y})$ is strictly convex on $\mathbb{R}^n$, and consequently admits a unique minimizer $\mathbf{x}^*$ satisfying

$$\nabla_{\mathbf{x}} h(\mathbf{x}^*,\mathbf{y}) = \mathbf{0}.$$

Using the definition of h given in (2.8), the latter identity can be explicitly written as

$$\sum_{j=1}^{m} (\mathbf{x}^* - \mathbf{a}_j - d_j r_j(\mathbf{y})) = \mathbf{0},$$

which by simple algebraic manipulation can be shown to be equivalent to $\mathbf{x}^* = \mathbf{y} - \frac{1}{2m}\nabla f(\mathbf{y})$, establishing that $\mathbf{x}^* = T(\mathbf{y})$. □

Using Lemma 2.1 we are now able to prove the monotonicity property of the operator T with respect to f.

Lemma 2.2 *Let* $\mathbf{y} \notin \mathcal{A}$. *Then*

$$f(T(\mathbf{y})) \leq f(\mathbf{y})$$

and equality holds if and only if $T(\mathbf{y}) = \mathbf{y}$.

Proof. By (2.11) and the strict convexity of the function $\mathbf{x} \mapsto h(\mathbf{x}, \mathbf{y})$, one has

$$h(T(\mathbf{y}), \mathbf{y}) < h(\mathbf{x}, \mathbf{y}) \text{ for every } \mathbf{x} \neq T(\mathbf{y}).$$

In particular, if $T(\mathbf{y}) \neq \mathbf{y}$ then

$$h(T(\mathbf{y}), \mathbf{y}) < h(\mathbf{y}, \mathbf{y}) = f(\mathbf{y}), \tag{2.12}$$

where the last equality follows from Lemma 2.1(a). By Lemma 2.1(b), $h(T(\mathbf{y}), \mathbf{y}) \geq f(T(\mathbf{y}))$, which combined with (2.12), establishes the desired strict monotonicity. □

Theorem 2.1 given below states the basic convergence results for the SFP method. In the proof we exploit the boundedness of the level sets of the objective function f, which is stated and proved in the following lemma.

Lemma 2.3 *The level sets of f are bounded.*

Proof. Let $\alpha \in \mathbb{R}$. We will show that $\text{Lev}(f, \alpha) = \{\mathbf{x} : f(\mathbf{x}) \leq \alpha\}$ is bounded. Indeed, $\|\mathbf{x}\| = \|\mathbf{x} - \mathbf{a}_j + \mathbf{a}_j\| \leq \|\mathbf{x} - \mathbf{a}_j\| + \|\mathbf{a}_j\|$. Therefore,

$$\begin{aligned} m\|\mathbf{x}\| - \sum_{j=1}^{m} d_j &\leq \sum_{j=1}^{m} (\|\mathbf{x} - \mathbf{a}_j\| - d_j) + \sum_{j=1}^{m} \|\mathbf{a}_j\| \\ &\leq \frac{1}{2} \sum_{j=1}^{m} (\|\mathbf{x} - \mathbf{a}_j\| - d_j)^2 + \frac{m}{2} + \sum_{j=1}^{m} \|\mathbf{a}_j\| \leq \frac{\alpha}{2} + \frac{m}{2} + \sum_{j=1}^{m} \|\mathbf{a}_j\| \end{aligned}$$

where the second inequality follows from using the inequality $z \leq \frac{1+z^2}{2}$ for every $z \in \mathbb{R}$, and hence the desired result follows. □

Theorem 2.1 (Convergence of the SFP Method) *Let $\{\mathbf{x}^k\}$ be generated by (2.4) such that $\mathbf{x}^0$ satisfies*

$$f(\mathbf{x}^0) < \min_{j=1,\ldots,m} f(\mathbf{a}_j). \tag{2.13}$$

Then,

(a) $\mathbf{x}^k \notin \mathcal{A}$ for every $k \geq 0$.

(b) For every $k \geq 0$, $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$ and equality is satisfied if and only if $\mathbf{x}^{k+1} = \mathbf{x}^k$.

(c) The sequence of function values $\{f(\mathbf{x}^k)\}$ converges.

(d) The sequence $\{\mathbf{x}^k\}$ is bounded.

(e) Every convergent subsequence $\{\mathbf{x}^{k_l}\}$ satisfies $\mathbf{x}^{k_l+1} - \mathbf{x}^{k_l} \to \mathbf{0}$.

(f) Any limit point of $\{\mathbf{x}^k\}$ is a stationary point of f.

Proof. (a) and (b) follow by induction on k using Lemma 2.2.
(c) Readily follows from the monotonicity and lower boundedness (by zero) of the sequence $\{f(\mathbf{x}^k)\}$.
(d) By part (b) all the iterates $\mathbf{x}^k$ are in the level set $Lev(f, f(\mathbf{x}^0))$ which, by Lemma 2.3, establishes the boundedness of the sequence $\{\mathbf{x}^k\}$.
(e) and (f) Let $\{\mathbf{x}^{k_l}\}$ be a convergent subsequence of $\{\mathbf{x}^k\}$ with limit point $\mathbf{x}^*$. Since $f(\mathbf{x}^{k_l}) \leq f(\mathbf{x}^0) < \min_{j=1,\ldots,m} f(\mathbf{a}_j)$, it follows by the continuity of f that $f(\mathbf{x}^*) \leq f(\mathbf{x}^0) < \min_{j=1,\ldots,m} f(\mathbf{a}_j)$, proving that $\mathbf{x}^* \notin \mathcal{A}$. By (2.6):

$$\mathbf{x}^{k_l+1} = T(\mathbf{x}^{k_l}). \tag{2.14}$$

Therefore, since the subsequence $\{\mathbf{x}^{k_l}\}$ and its limit point $\mathbf{x}^*$ are not in $\mathcal{A}$, by the continuity of ∇f on $\mathbb{R}^n \setminus \mathcal{A}$, we conclude that the subsequence $\{\mathbf{x}^{k_l+1}\}$ converges to a vector $\bar{\mathbf{x}}$ satisfying

$$\bar{\mathbf{x}} = T(\mathbf{x}^*). \tag{2.15}$$

To prove (e), we need to show that $\bar{\mathbf{x}} = \mathbf{x}^*$. Since both $\mathbf{x}^*$ and $\bar{\mathbf{x}}$ are limit points of $\{\mathbf{x}^k\}$ and since the sequence of function values converges (by part (c)), then the continuity of f over $\mathbb{R}^n$ implies $f(\mathbf{x}^*) = f(\bar{\mathbf{x}})$. Invoking Lemma 2.2 for $\mathbf{y} = \mathbf{x}^*$, we conclude that $\bar{\mathbf{x}} = \mathbf{x}^*$, proving claim (e). Part (f) follows from the observation that the equality $\mathbf{x}^* = T(\mathbf{x}^*)$ is equivalent (by the definition of T) to $\nabla f(\mathbf{x}^*) = \mathbf{0}$. □

Remark 2.1 It is easy to find a vector $\mathbf{x}^0$ satisfying the condition (2.13). For example, procedure INIT that will be described at the end of this section produces a point satisfying (2.13).

Combining Claims (c) and (f) of Theorem 2.1 we immediately obtain convergence of the sequence of function values.

Corollary 2.1 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the SFP algorithm satisfying (2.13). Then $f(\mathbf{x}^k) \to f^*$, where f^* is the function value at a stationary point of f.*

We were able to prove the convergence of the function values of the sequence. The situation is more complicated for the sequence itself where we were only able to show that all limit points are stationary points. We can prove convergence of the sequence itself if we assume that all stationary points of the objective function are isolated[3]. The proof of this claim strongly relies on the following Lemma from [9].

Lemma 2.4 ([9, Lemma 4.10]) *Let $\mathbf{x}^*$ be an isolated limit point of a sequence $\{\mathbf{x}^k\}$ in $\mathbb{R}^n$. If $\{\mathbf{x}^k\}$ does not converge then there is a subsequence $\{\mathbf{x}^{k_l}\}$ which converges to $\mathbf{x}^*$ and an $\epsilon > 0$ such that $\|\mathbf{x}^{k_l+1} - \mathbf{x}^{k_l}\| \geq \epsilon$.*

We can now use the above lemma to prove a convergence result under the assumption that all stationary points of f are isolated.

[3]We say that $\mathbf{x}^*$ is an isolated stationary point of f, if there are no other stationary points in some neighborhood of $\mathbf{x}^*$.

Theorem 2.2 (Convergence of the Sequence) *Let $\{\mathbf{x}^k\}$ be generated by (2.4) such that $\mathbf{x}^0$ satisfies (2.13). Suppose further that all stationary point of f are isolated. Then the sequence $\{\mathbf{x}^k\}$ converges to a stationary point.*

Proof. Let $\mathbf{x}^*$ be a limit point of $\{\mathbf{x}^k\}$ (its existence follows from the boundedness of the sequence proved in Theorem 2.1(d)). By our assumption $\mathbf{x}^*$ is an isolated point. Suppose in contradiction that the sequence does not converge. Then by Lemma 2.4 there exists a subsequence $\{\mathbf{x}^{k_l}\}$ that converges to $\mathbf{x}^*$ satisfying $\|\mathbf{x}^{k_l+1} - \mathbf{x}^{k_l}\| \geq \epsilon$. However, this is in contradiction to part (e) of Theorem 2.1. We thus conclude that $\{\mathbf{x}^k\}$ converges to a stationary point. □

The analysis of the SFP method relies on the validity of condition (2.13) on the starting point $\mathbf{x}^0$. We will now show that thanks to the special structure of the objective function (ML) we can compute such a point through a simple procedure. This is achieved by establishing the following result.

Lemma 2.5 *Let $\mathcal{A} \equiv \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ be the given set of m sensors and let*

$$g_j(\mathbf{x}) = \sum_{i=1, i\neq j}^{m} (\|\mathbf{x} - \mathbf{a}_i\| - d_i)^2, \quad j = 1, \ldots, m.$$

Then for every $j = 1, \ldots, m$

(i) If $\nabla g_j(\mathbf{a}_j) \neq \mathbf{0}$ then $f'(\mathbf{a}_j; -\nabla g_j(\mathbf{a}_j)) < 0$. Otherwise, if $\nabla g_j(\mathbf{a}_j) = \mathbf{0}$, then $f'(\mathbf{a}_j; \mathbf{v}) < 0$ for every $\mathbf{v} \neq \mathbf{0}$. In particular, there exists a descent direction from every sensor point.

(ii) Every $\bar{\mathbf{x}} \in \mathcal{A}$ is not a local optimum for the ML problem (1.2).

Proof. (i). For convenience, for every $j = 1, \ldots, m$ we denote

$$f_j(\mathbf{x}) = (\|\mathbf{x} - \mathbf{a}_j\| - d_j)^2 \tag{2.16}$$

so that the objective function of problem (ML) can be written as

$$f(\mathbf{x}) = f_j(\mathbf{x}) + g_j(\mathbf{x}) \tag{2.17}$$

for every $\mathbf{x} \in \mathbb{R}^n$ and $j = 1, \ldots, m$. Note that f is not differentiable for every $\mathbf{x} \in \mathcal{A}$. Nonetheless, the directional derivative of f at $\mathbf{x}$ in the direction $\mathbf{v} \in \mathbb{R}^n$ always exist and is given by

$$f'(\bar{\mathbf{x}}; \mathbf{v}) = \begin{cases} \nabla f(\bar{\mathbf{x}})^T \mathbf{v} & \bar{\mathbf{x}} \notin \mathcal{A}, \\ \nabla g_j(\mathbf{a}_j)^T \mathbf{v} - 2d_j \|\mathbf{v}\| & \bar{\mathbf{x}} = \mathbf{a}_j. \end{cases} \tag{2.18}$$

Indeed, the above formula for $\bar{\mathbf{x}} \notin \mathcal{A}$ is obvious. In the other case, suppose then that $\bar{\mathbf{x}} = \mathbf{a}_j$ for some $j \in \{1, \ldots, m\}$. Noting that g_j is differentiable at $\mathbf{a}_j$ we have $g_j'(\mathbf{a}_j; \mathbf{v}) = \nabla g_j(\mathbf{a}_j)^T \mathbf{v}$, and using the definition (2.16) for f_j, we get $f_j'(\mathbf{a}_j; \mathbf{v}) = -2d_j \|\mathbf{v}\|$, and hence with (2.17), we obtain the desired formula (2.18) for $f'(\mathbf{a}_j;\mathbf{v})$. Finally, if $\nabla g_j(\mathbf{a}_j) \neq \mathbf{0}$, then using (2.18) we have

$$f'(\mathbf{a}_j; -\nabla g_j(\mathbf{a}_j)) = -\|\nabla g_j(\mathbf{a}_j)\|^2 - 2d_j\|\nabla g_j(\mathbf{a}_j)\| < 0.$$

Otherwise, if $\nabla g_j(\mathbf{a}_j) = \mathbf{0}$ then for every $\mathbf{v} \neq \mathbf{0}$:

$$f'(\mathbf{a}_j;\mathbf{v}) = -2d_j\|\mathbf{v}\| < 0.$$

(ii) By part (i) there exists a descent direction from every sensor point $\bar{\mathbf{x}} \in \mathcal{A}$. Therefore, any of the sensor points cannot be a local optimum for problem (ML). □

Using the descent directions provided by Lemma 2.5, we can compute a point $\bar{\mathbf{x}}$ satisfying

$$f(\bar{\mathbf{x}}) < \min_{j=1,\ldots,m} f(\mathbf{a}_j)$$

by the following procedure:

Procedure INIT

1. $t = 1$.
2. **Set** k to be an index for which $f(\mathbf{a}_k) = \min_{j=1,\ldots,m} f(\mathbf{a}_j)$.
3. **Set**
$$\mathbf{v}_0 = \begin{cases} -\nabla g_k(\mathbf{a}_k) & \nabla g_k(\mathbf{a}_k) \neq \mathbf{0}, \\ \mathbf{e} & \nabla g_k(\mathbf{a}_k) = \mathbf{0}, \end{cases} \tag{2.19}$$
where $\mathbf{e}$ is the vectors of all ones[4].
4. **While** $f(\mathbf{a}_k + t\mathbf{v}_0) \geq f(\mathbf{a}_k)$ **set** $t = t/2$. **End**
5. The output of the algorithm is $\mathbf{a}_k + t\mathbf{v}_0$.

The validity of this procedure stems from the fact that by Lemma 2.5, the direction $\mathbf{v}_0$ defined in (2.19) is always a descent direction.

One of the advantages of the SFP scheme is its simplicity. However, the SFP method, being a gradient method, does have the tendency to converge to local minima. In the next section we will present a second and more involved algorithm to solve the ML problem. As we shall see in the numerical examples presented in Section 4, the empirical performance of this second iterative scheme is significantly better than that of the SFP both with respect to the number of required iterations, and with respect to the probability of getting stuck in a local/non-global point.

[4] We could have chosen any other nonzero vector.

3 A Sequential Weighted Least Squares Algorithm

In this section we study a different method for solving the ML problem (1.2) which we call the sequential weighted least squares (SWLS) algorithm. The SWLS algorithm is also motivated by the construction of Weiszfeld method, but from a different viewpoint, see Section 3.1. Each iteration of the method consists of solving a nonlinear least squares problem whose solution is found by the approach discussed in Section 3.2. The convergence analysis the SWLS algorithm is given in Section 3.3.

3.1 The SWLS Algorithm

To motivate the SWLS algorithm, let us first go back to the Weiszfeld scheme for solving the classical Fermat-Weber location problem, whereby we rewrite the iterative scheme (2.2) in the following equivalent but different way:

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \left\{ \sum_{j=1}^{m} \omega_j \frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{x}^k-\mathbf{a}_j\|} \right\}. \tag{3.1}$$

The strong convexity of the objective function in (3.1) implies that $\mathbf{x}^{k+1}$ is uniquely defined as a function of $\mathbf{x}^k$. Therefore, Weiszfeld method (2.2) for solving problem (2.1) can also be written as

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} q(\mathbf{x},\mathbf{x}^k),$$

where

$$q(\mathbf{x},\mathbf{y}) \equiv \sum_{j=1}^{m} \omega_j \frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{y}-\mathbf{a}_j\|} \text{ for every } \mathbf{x}\in\mathbb{R}^n, \mathbf{y}\in\mathbb{R}^n\setminus\mathcal{A}.$$

The auxiliary function q was essentially constructed from the objective function s of the Fermat-Weber location problem, by replacing the norm terms $\|\mathbf{x}-\mathbf{a}_j\|$ with $\frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{y}-\mathbf{a}_j\|}$, i.e., with $s(\mathbf{x}) = q(\mathbf{x},\mathbf{x})$. Mimicking this observation for the ML problem under study, we will use an auxiliary function in which each norm term $\|\mathbf{x}-\mathbf{a}_j\|$ in the objective function (1.2) is replaced with $\frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{y}-\mathbf{a}_j\|}$, resulting in the following auxiliary function:

$$g(\mathbf{x},\mathbf{y}) \equiv \sum_{i=1}^{m} \left(\frac{\|\mathbf{x}-\mathbf{a}_i\|^2}{\|\mathbf{y}-\mathbf{a}_i\|} - d_i \right)^2, \quad \mathbf{x}\in\mathbb{R}^n, \mathbf{y}\in\mathbb{R}^n\setminus\mathcal{A}. \tag{3.2}$$

The general step of the algorithm for solving problem (ML), termed *the sequential weighted least squares* (SWLS) method, is now given by

$$\mathbf{x}^{k+1} \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x},\mathbf{x}^k).$$

or more explicitly by

Algorithm SWLS:

$$\mathbf{x}^{k+1} \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \sum_{j=1}^{m} \left(\frac{\|\mathbf{x}-\mathbf{a}_j\|^2}{\|\mathbf{x}^k-\mathbf{a}_j\|} - d_j \right)^2. \tag{3.3}$$

The above minimization problem might have several global minima; in these cases $\mathbf{x}^{k+1}$ is arbitrary chosen as one of the global minima.

The name SWLS stems from the fact that at each iteration k we are required to solve the following weighted version of the LS problem (1.3):

$$\text{(WLS):} \quad \min_{\mathbf{x}} \sum_{j=1}^{m} \omega_j^k (\|\mathbf{x} - \mathbf{c}_j\|^2 - \beta_j^k)^2, \tag{3.4}$$

with

$$\mathbf{c}_j = \mathbf{a}_j, \beta_j^k = d_j \|\mathbf{x}^k - \mathbf{a}_j\|, \omega_j^k = \frac{1}{\|\mathbf{x}^k - \mathbf{a}_j\|^2}. \tag{3.5}$$

Note that the SWLS algorithm as presented above is not defined for iterations in which $\mathbf{x}^k \in \mathcal{A}$. In our random numerical experiments (c.f., Section 4) this situation never occurred, i.e., $\mathbf{x}^k$ did not belong to $\mathcal{A}$ for every k. However, from a theoretical point of view this issue must be resolved. Similarly to the methodology advocated in the convergence analysis of the SFP method, our approach for avoiding the sensor points $\mathcal{A}$ is by choosing a "good enough" initial vector. In Section 3.3, we introduce a simple condition on the initial vector $\mathbf{x}^0$ under which the algorithm is well defined and proven to converge.

3.2 Solving the WLS Subproblem

We will now show how the WLS subproblem (3.4) can be solved globally and efficiently by transforming it into a problem of minimizing a quadratic function subject to a single quadratic constraint. This derivation is a straightforward extension of the solution technique devised in [1] and is briefly described here for completeness.
For a given fixed k (for simplicity we omit the index k below), we first transform (3.4) into a constrained minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n, \alpha\in\mathbb{R}} \left\{ \sum_{j=1}^{m} \omega_j (\alpha - 2\mathbf{c}_j^T \mathbf{x} + \|\mathbf{c}_j\|^2 - \beta_j)^2 : \|\mathbf{x}\|^2 = \alpha \right\}, \tag{3.6}$$

which can also be written as (using the substitution $\mathbf{y} = (\mathbf{x}^T, \alpha)^T$)

$$\min_{\mathbf{y}\in\mathbb{R}^{n+1}} \left\{ \|\mathbf{A}\mathbf{y} - \mathbf{b}\|^2 : \mathbf{y}^T \mathbf{D} \mathbf{y} + 2\mathbf{f}^T \mathbf{y} = 0 \right\}, \tag{3.7}$$

where

$$\mathbf{A} = \begin{pmatrix} -2\sqrt{\omega_1}\mathbf{c}_1^T & \sqrt{\omega_1} \\ \vdots & \vdots \\ -2\sqrt{\omega_m}\mathbf{c}_m^T & \sqrt{\omega_m} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \sqrt{\omega_1}(\beta_1 - \|\mathbf{c}_1\|^2) \\ \vdots \\ \sqrt{\omega_m}(\beta_m - \|\mathbf{c}_m\|^2) \end{pmatrix}$$

and

$$\mathbf{D} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{1\times n} & 0 \end{pmatrix}, \mathbf{f} = \begin{pmatrix} 0 \\ -0.5 \end{pmatrix}.$$

Note that (3.7) belongs to the class of problems consisting of minimizing a quadratic function subject to a single quadratic constraint. Problems of this type are called generalized trust region subproblems (GTRS). GTRS problems possess necessary and sufficient optimality conditions from which efficient solution methods can be derived, see e.g., [6, 8].

The SWLS scheme is of course more involved than the simpler SFP scheme. However, as explained above, the additional computations required in SWLS to solve the subproblem can be done efficiently and are worthwhile, since the SWLS algorithm usually possesses a much larger region of convergence to the global minimum than the SFP scheme, which in turn implies that it has the tendency of avoiding local minima, and a greater chance to hit the global minimum. This will be demonstrated on the numerical examples given in Section 4.

3.3 Convergence Analysis of the SWLS Method

In this section we provide an analysis of the SWLS method. We begin by presenting our underlying assumptions in Section 3.3.1 and in Section 3.3.2 we prove convergence results of the method.

3.3.1 Underlying Assumptions

The following assumption will be made throughout this section

Assumption 1 *The matrix*

$$\mathbf{A} = \begin{pmatrix} 1 & \mathbf{a}_1^T \\ 1 & \mathbf{a}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{a}_m^T \end{pmatrix}$$

is of full column rank.

This assumption is equivalent to saying that $\mathbf{a}_1, \ldots, \mathbf{a}_m$ do not reside in a lower dimensional affine space (i.e., a line if $n = 2$ and a plane if $n = 3$).

To guarantee the well definiteness of the SWLS algorithm (i.e., $\mathbf{x}^k \notin \mathcal{A}$ for all k) we will make the following assumption on the initial vector $\mathbf{x}^0$:

Assumption 2 $\mathbf{x}^0 \in \mathcal{R}$ *where*

$$\mathcal{R} := \left\{ \mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) < \frac{\min_j \{d_j\}^2}{4} \right\}. \tag{3.8}$$

A similar assumption was made for the SFP (see condition (2.13)). Note that for the true source location $\mathbf{x}_{\text{true}}$ one has $f(\mathbf{x}_{\text{true}}) = \sum_{j=1}^m \varepsilon_j^2$. Therefore, $\mathbf{x}_{\text{true}}$ satisfies Assumption 2 if the errors ε_j are smaller in some sense from the range measurements d_j. This is a very reasonable assumption since in real applications the errors ε_i are often smaller in an order of magnitude than d_i. Now, if the initial point $\mathbf{x}^0$ is "good enough" in the sense that it close to the true source location, then Assumption 2 will be satisfied. We have observed through numerical experiments that the solution to the LS problem (1.3) often satisfies Assumption 2 as the following example demonstrates.

Example 3.1 Consider the source localization problem with $m = 5, n = 2$. We performed Monte-Carlo runs where in each run the sensor locations $\mathbf{a}_j$ and the source location $\mathbf{x}$ were randomly generated from a uniform distribution over the square $[-20, 20] \times [-20, 20]$. The observed distances d_j are given by (1.1) with ε_j being independently generated from a normal distribution with mean zero and standard deviation σ. In our experiments σ takes on four different values: $1, 10^{-1}, 10^{-2}$, and 10^{-3}. For each σ, N_σ denotes the number of runs for which the condition $f(\mathbf{x}_{\text{LS}}) < \frac{\min_j d_j^2}{4}$ holds, and is given in the following table. Clearly, the Assumption 2 fails only for high noise levels.

σ	1e-3	1e-2	1e-1	1e+0
N_σ	10000	10000	9927	6281

Table 1: Number of runs (out of 10000) for which Assumption 2 is satisfied for $\mathbf{x}^0 = \mathbf{x}_{\text{LS}}$

The following simple and important property will be used in our analysis.

Lemma 3.1 *Let* $\mathbf{x} \in \mathcal{R}$. *Then*

$$\|\mathbf{x} - \mathbf{a}_j\| > d_j/2, \quad j = 1, \ldots, m. \tag{3.9}$$

Proof. Suppose in contradiction that there exists j_0 for which $\|\mathbf{x} - \mathbf{a}_{j_0}\| \leq d_{j_0}/2$. Then

$$f(\mathbf{x}) = \sum_{j=1}^{m} (\|\mathbf{x} - \mathbf{a}_j\| - d_j)^2 \geq (\|\mathbf{x} - \mathbf{a}_{j_0}\| - d_{j_0})^2 \geq \frac{d_{j_0}^2}{4} \geq \frac{\min\{d_j\}^2}{4},$$

which contradicts $\mathbf{x} \in \mathcal{R}$. □

A direct consequence of the Lemma 3.1 is that any element in $\mathcal{R}$ cannot be one of the sensors.

Corollary 3.1 *If* $\mathbf{x} \in \mathcal{R}$ *then* $\mathbf{x} \notin \mathcal{A}$.

3.3.2 Convergence Analysis of the SWLS method

We begin with the following result which plays a key role in the forthcoming analysis.

Lemma 3.2 *Let δ be a positive number and let $t > \delta/2$. Then*

$$\left(\frac{s^2}{t} - \delta\right)^2 \geq 2(s - \delta)^2 - (t - \delta)^2 \tag{3.10}$$

for every $s > \sqrt{\frac{\delta t}{2}}$, and equality is satisfied if and only if $s = t$.

Proof. Rearranging (3.10) one has to prove

$$A(s,t) \equiv \left(\frac{s^2}{t} - \delta\right)^2 - 2(s-\delta)^2 + (t-\delta)^2 \geq 0.$$

Some algebra shows that the expression $A(s,t)$ can be written as follows:

$$A(s,t) = \frac{1}{t}(s-t)^2 \left(\left(\frac{s}{\sqrt{t}} + \sqrt{t} \right)^2 - 2\delta \right). \tag{3.11}$$

Using the conditions $t > \delta/2$ and $s > \sqrt{\frac{\delta t}{2}}$ we obtain

$$\left(\frac{s}{\sqrt{t}} + \sqrt{t} \right)^2 - 2\delta > \left(\sqrt{\frac{\delta}{2}} + \sqrt{\frac{\delta}{2}} \right)^2 - 2\delta = 0. \tag{3.12}$$

Therefore, from (3.11) and (3.12) it readily follows that $A(s,t) \geq 0$ and that equality holds if and only if $s = t$. □

Thanks to Lemma 3.2, we establish the next result which is essential in proving the monotonicity of the SWLS method.

Lemma 3.3 *Let* $\mathbf{y} \in \mathcal{R}$*. Then, the function* $g(\mathbf{x}, \mathbf{y})$ *given in (3.2) is well defined on* $\mathbb{R}^n \times \mathcal{R}$*, and with*

$$\mathbf{z} \in \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^n} g(\mathbf{x}, \mathbf{y}), \tag{3.13}$$

the following properties hold:

(a) $f(\mathbf{z}) \leq f(\mathbf{y})$*, and the equality is satisfied if and only if* $\mathbf{z} = \mathbf{y}$*.*

(b) $\mathbf{z} \in \mathcal{R}$*.*

Proof. By Corollary 3.1, any $\mathbf{y} \in \mathcal{R}$ implies $\mathbf{y} \notin \mathcal{A}$, and hence the function g given by (cf. (3.2))

$$g(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \left(\frac{\|\mathbf{x} - \mathbf{a}_i\|^2}{\|\mathbf{y} - \mathbf{a}_i\|} - d_i \right)^2,$$

is well defined on $\mathbb{R}^n \times \mathcal{R}$. Now, by (3.13) and $\mathbf{y} \in \mathcal{R}$ we have

$$g(\mathbf{z}, \mathbf{y}) \leq g(\mathbf{y}, \mathbf{y}) = f(\mathbf{y}) < \frac{\min\{d_j\}^2}{4}. \tag{3.14}$$

In particular,

$$\left(\frac{\|\mathbf{z} - \mathbf{a}_j\|^2}{\|\mathbf{y} - \mathbf{a}_j\|} - d_j \right)^2 < \frac{d_j^2}{4}, \quad j = 1, \ldots, m$$

from which it follows that

$$\frac{\|\mathbf{z} - \mathbf{a}_j\|^2}{\|\mathbf{y} - \mathbf{a}_j\|} \geq \frac{d_j}{2}, \quad j = 1, \ldots, m. \tag{3.15}$$

Invoking Lemma 3.2, whose conditions are satisfied by (3.15) and Lemma 3.1 we obtain

$$\left(\frac{\|\mathbf{z}-\mathbf{a}_j\|^2}{\|\mathbf{y}-\mathbf{a}_j\|}-d_j\right)^2 \geq 2(\|\mathbf{z}-\mathbf{a}_j\|-d_j)^2-(\|\mathbf{y}-\mathbf{a}_j\|-d_j)^2.$$

Summing over $j=1,\ldots,m$ we obtain

$$\sum_{j=1}^m\left(\frac{\|\mathbf{z}-\mathbf{a}_j\|^2}{\|\mathbf{y}-\mathbf{a}_j\|}-d_j\right)^2 \geq 2\sum_{j=1}^m(\|\mathbf{z}-\mathbf{a}_j\|-d_j)^2-\sum_{j=1}^m(\|\mathbf{y}-\mathbf{a}_j\|-d_j)^2.$$

Therefore, together with (3.14), we get

$$f(\mathbf{y}) \geq g(\mathbf{z},\mathbf{y}) \geq 2f(\mathbf{z})-f(\mathbf{y}),$$

showing that $f(\mathbf{z}) \leq f(\mathbf{y})$. Now, assume that $f(\mathbf{y})=f(\mathbf{z})$. Then by Lemma 3.2 it follows that the following set of equalities is satisfied:

$$\|\mathbf{y}-\mathbf{a}_j\|=\|\mathbf{z}-\mathbf{a}_j\|, \quad j=1,\ldots,m, \tag{3.16}$$

which after squaring and rearranging reads as

$$(\|\mathbf{y}\|^2-\|\mathbf{z}\|^2)-2\mathbf{a}_j^T(\mathbf{y}-\mathbf{z})=0, \quad j=1,\ldots,m.$$

Therefore,

$$\begin{pmatrix} 1 & \mathbf{a}_1^T \\ 1 & \mathbf{a}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{a}_m^T \end{pmatrix}\begin{pmatrix} \|\mathbf{y}\|^2-\|\mathbf{z}\|^2 \\ -2(\mathbf{y}-\mathbf{z}). \end{pmatrix}=0$$

Thus, by Assumption 1, $\mathbf{z}=\mathbf{y}$, and the proof of (a) is completed. To prove (b), using (a) and (3.14), we get

$$f(\mathbf{z}) \leq f(\mathbf{y}) < \min_{j=1,\ldots,m}\frac{d_j^2}{4},$$

proving that $\mathbf{z}\in\mathcal{R}$. □

We are now ready to prove the main convergence results for the SWLS method.

Theorem 3.1 (Convergence of the SWLS Method) *Let $\{\mathbf{x}^k\}$ be the sequence generated by the SWLS method. Suppose that Assumptions 1 and 2 hold true. Then*

(a) $\mathbf{x}^k \in \mathcal{R}$ *for* $k \geq 0$.

(b) For every $k \geq 0$, $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$ *and equality holds if and only if* $\mathbf{x}^{k+1}=\mathbf{x}^k$.

(c) The sequence of function values $\{f(\mathbf{x}^k)\}$ *converges.*

(d) The sequence $\{\mathbf{x}^k\}$ *is bounded.*

(e) Every convergent subsequence $\{\mathbf{x}^{k_l}\}$ *satisfies* $\mathbf{x}^{k_l+1}-\mathbf{x}^{k_l}\rightarrow\mathbf{0}$

(f) Any limit point of $\{\mathbf{x}^k\}$ *is a stationary point of* f.

Proof. (a) and (b) follow by induction on k using Lemma 3.3.
(c) Follows from the fact that $\{f(\mathbf{x}^k)\}$ is bounded below (by zero) and a nonincreasing sequence.
(d) By part (b) all the iterates $\mathbf{x}^k$ are in the level set $Lev(f, f(\mathbf{x}^0))$ which, by Lemma 2.3, establishes the boundedness of the sequence $\{\mathbf{x}^k\}$.
(e) Let $\{\mathbf{x}^{k_l}\}$ be a convergent subsequence and denote its limit by $\mathbf{x}^*$. By claims (a) and (b) we have for every k

$$f(\mathbf{x}^k) \le f(\mathbf{x}^0) < \min_{j=1,\ldots,m} \frac{d_j^2}{4},$$

which combined with the continuity of f imply $\mathbf{x}^* \in \mathcal{R}$ and hence $\mathbf{x}^* \notin \mathcal{A}$, by Corollary 3.1. Now, recall that

$$\mathbf{x}^{k_l+1} \in \operatorname{argmin} g(\mathbf{x}, \mathbf{x}^{k_l}).$$

To prove the convergence of $\{\mathbf{x}^{k_l+1}\}$ to $\mathbf{x}^*$, we will show that every subsequence converges to $\mathbf{x}^*$. Let $\{\mathbf{x}^{k_{l_p}+1}\}$ be a convergent subsequence and denote its limit by $\mathbf{y}^*$. Since

$$\mathbf{x}^{k_{l_p}+1} \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x}, \mathbf{x}^{k_{l_p}}),$$

the following holds:

$$g(\mathbf{x}, \mathbf{x}^{k_{l_p}}) \ge g(\mathbf{x}^{k_{l_p}+1}, \mathbf{x}^{k_{l_p}}) \text{ for every } \mathbf{x} \in \mathbb{R}^n.$$

Taking the limits of both sides in the last inequality and using the continuity of the function f we have

$$g(\mathbf{x}, \mathbf{x}^*) \ge g(\mathbf{y}^*, \mathbf{x}^*) \text{ for every } \mathbf{x} \in \mathbb{R}^n,$$

and hence,

$$\mathbf{y}^* \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x}, \mathbf{x}^*). \tag{3.17}$$

Since the sequence of function values converges, it follows that $f(\mathbf{x}^*) = f(\mathbf{y}^*)$. Invoking Lemma 3.3 with $\mathbf{y} = \mathbf{x}^*$ and $\mathbf{z} = \mathbf{y}^*$ we obtain $\mathbf{x}^* = \mathbf{y}^*$, establishing claim (e).
(f) To prove the claim, note that (3.17) and $\mathbf{x}^* = \mathbf{y}^*$ imply

$$\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x}, \mathbf{x}^*).$$

Thus, by the first order optimality conditions we obtain:

$$0 = \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{x}^*)|_{\mathbf{x}=\mathbf{x}^*} = 4\sum_{j=1}^{m} (\|\mathbf{x}^* - \mathbf{a}_j\| - d_j)\frac{\mathbf{x}^* - \mathbf{a}_j}{\|\mathbf{x}^* - \mathbf{a}_j\|} = 2\nabla f(\mathbf{x}^*).$$

□

As a direct consequence of Theorem 3.1 we obtain convergence in function values.

Corollary 3.2 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the algorithm. Then $f(\mathbf{x}^k) \to f^*$, where f^* is the function value at some stationary point $\mathbf{x}^*$ of f.*

As was shown for the SFP algorithm, global convergence of the sequence generated by the SWLS algorithm can also be established under the same condition i.e., assuming that f admits isolated stationary points.

Theorem 3.2 (Convergence of the Sequence) *Let $\{\mathbf{x}^k\}$ be generated by (3.3) such that Assumptions 1 and 2 hold. Suppose further that all stationary point of f are isolated. Then the sequence $\{\mathbf{x}^k\}$ converges to a stationary point.*

Proof. The same as the proof of Theorem 2.2. □

4 Numerical Examples

In this section we present numerical simulations illustrating the performance of the SFP and SWLS schemes, as well as numerical comparisons with the Least Squares approach, and with the semidefinite relaxation (SDR) of the ML problem. The simulations were performed in MATLAB and the semidefine programs were solved by SeDuMi [14].

Before describing the numerical results, for the reader's convenience we first recall the semidefinite relaxation (SDR) proposed in [4], and which will be used in our numerical experiments comparisons. The first stage is to rewrite problem (ML) given in (1.2) as

$$\begin{array}{ll} \min_{\mathbf{x},\mathbf{g}} & \sum_{j=1}^m (g_j - d_j)^2 \\ \text{s.t.} & g_j^2 = \|\mathbf{x} - \mathbf{a}_j\|^2, \quad j = 1, \ldots, m. \end{array}$$

Making the change of variables

$$\mathbf{G} = \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{g}^T & 1 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^T & 1 \end{pmatrix},$$

problem (1.2) becomes

$$\begin{array}{ll} \min_{\mathbf{X},\mathbf{G}} & \sum_{j=1}^m (G_{jj} - 2d_j G_{m+1,j} + d_j^2) \\ \text{s.t.} & G_{jj} = \mathrm{Tr}(\mathbf{C}_j \mathbf{X}), \quad j = 1, \ldots, m, \\ & \mathbf{G} \succeq \mathbf{0}, \mathbf{X} \succeq \mathbf{0}, \\ & G_{m+1,m+1} = X_{n+1,n+1} = 1, \\ & \mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\mathbf{G}) = 1, \end{array}$$

where

$$\mathbf{C}_j = \begin{pmatrix} \mathbf{I} & -\mathbf{a}_j \\ -\mathbf{a}_j^T & \|\mathbf{a}_j\|^2 \end{pmatrix}, \quad j = 1, \ldots, m.$$

Dropping the rank constraints in the above problem, we obtain the desired SDR of problem (1.2). The SDR is not guaranteed to provide an accurate solution to the ML problem, but it can always be considered as an approximation of the ML problem.

In the first example, we show that the SWLS scheme usually possesses a larger region of convergence to the global minimum than the scheme SFP. This last property is further demonstrated in the second example which compares the SFP and SWLS methods, and also demonstrates the superiority of the SWLS scheme. The last example illustrates the attractiveness of the solution obtained by the SWLS method over the SDR and the LS approaches.

Example 4.1 (**Region of Convergence of the SFP and SWLS methods**) In this example we show a typical behavior of the SFP and SWLS methods. Consider an instance of the source localization problem in the plane ($n = 2$) with three sensors ($m = 3$). Figure 1 and Figure 2 describe the results produced by the iterative schemes SFP and SWLS for three initial trial points. The global minimum is $(0.4285, 0.9355)$ and there exists one additional local minimum at $(0.1244, 0.3284)$. As demonstrated in Figure (2), the SWLS method might converge to a local minimum; however, it seems to have a greater chance than the SFP algorithm to avoid local minima; for example the SWLS converged to the (relatively far) global minimum from the initial starting point (0.5,0.1), while the SFP converged to the local minimum. The region of convergence to the local minima of both methods is denoted by red points in Figures 3 and 4. Obviously, the SWLS method has a much wider region of convergence to the global minimum. This was our observation in many other examples that we ran, and which suggest that the SWLS has the tendency to converge to the global minimum.

Remark 4.1 *As shown in Proposition 2.1, the SFP scheme is just a gradient method with a fixed step size. Thanks to Lemma 2.5, which as shown in Section 2.2 can be used in order to avoid the nonsmoothness, we can of course use more sophisticated smooth unconstrained minimization methods. Indeed, we also tested a gradient method with an Armijo stepsize rule, and a trust region method [9], which uses second order information. Our observation was that while these methods usually posses an improved rate of convergence in comparison to the SFP method, they essentially have the same region of convergence to the global minimum as the SFP algorithm.*

Example 4.2 (**Comparison of the SFP and SWLS Methods**) We performed Monte-Carlo runs where in each run the sensor locations $\mathbf{a}_j$ and the true source location were randomly generated from a uniform distribution over the square $[-1000, 1000] \times [-1000, 1000]$. The observed distances d_j are given by (1.1) with ε_j being generated from a normal distribution with mean zero and standard deviation 20. Both the SFP and SWLS methods were employed with (the same) initial point which was also uniformly randomly generated from the square $[-1000, 1000] \times [-1000, 1000]$. The stopping rule for both the SWLS and SFP methods was $\|\nabla f(\mathbf{x}^k)\| < 10^{-5}$.

The results of the runs are summarized in Table 5 below. For each value of m, 1000 realizations were generated. The numbers in the first column are the number of sensors, and in the second column we give the number of runs out of 1000 in which the SDR of the ML problem was tight. The third column contains the number of runs out of 1000 in

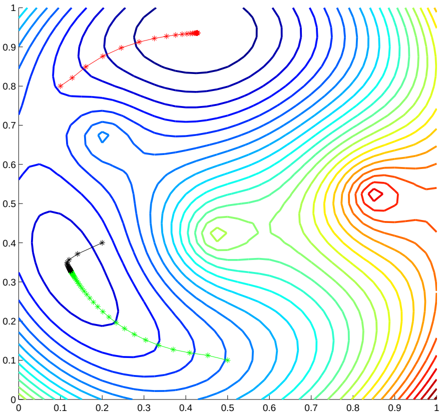

Fig. 1: The SFP method for three initial points.

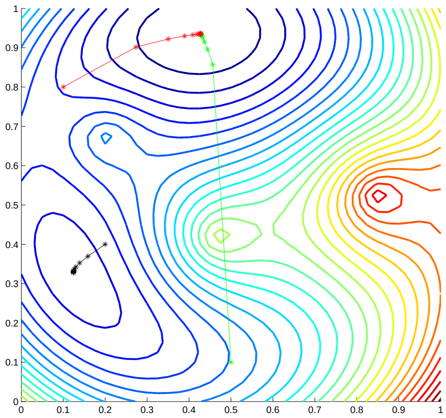

Fig. 2: The SWLS method for three initial points.

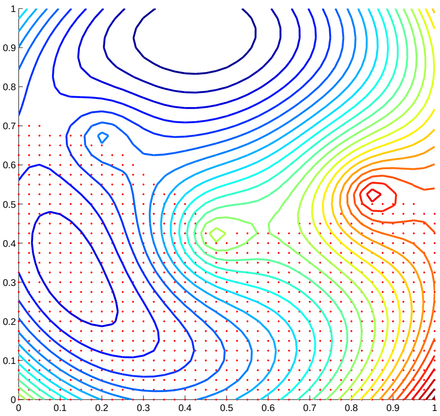

Fig. 3: Region of convergence to the local minimum marked by red points for SFP.

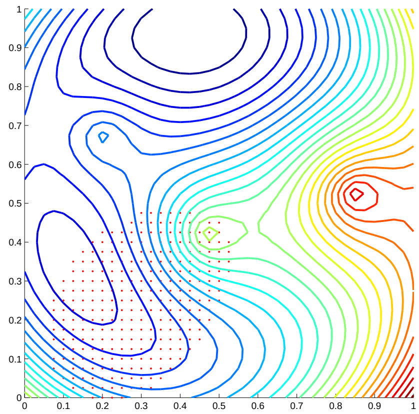

Fig. 4: Region of convergence to the local minimum marked by red points for SWLS.

which the solution produced by the SFP method was worse than the SWLS method. In all the remaining runs the two methods converge to the same point; thus, there were no runs in which the SWLS produced worse results. The last two columns contain the mean and standard deviation of the number of iterations of each of the method in the form "mean (standard deviation)".

m	#tight	$\#(f(\hat{\mathbf{x}}_{\text{SFP}}) > f(\hat{\mathbf{x}}_{\text{SWLS}}))$	Iter – SFP	Iter – SWLS
3	314	152	207(500.2)	26.2 (5)
4	325	96	124(192.6)	29.9(1.8)
5	259	83	93.6(96.2)	30.9(3.1)
10	278	23	66.5 (35.3)	31.6 (1.3)

Table 5: Comparison between the SFP and SWLS methods

As can be clearly seen from the table, the SWLS method requires much less iterations than the SFP method and in addition it is more robust in the sense that the number of iterations are more or less constant. In contrast, the standard deviations of the number of iterations of the SFP method are quite large. For example, the huge standard deviation 500.2 in the first row stems from the fact that in some of the runs the SFP algorithm required thousands of iterations!

From the above examples we conclude that the SWLS method does tend to converge to the global minimum. Of course, we can always construct an example in which the method converges to a local minimum (as was demonstrated in Example 4.1), but it seems that for random instances this convergence to a non-global solution is not likely.
We should also note that we also compared the SFP and SWLS methods with initial point chosen as the solution of the LS problem (1.3). For this choice of initial point, the SFP and SWLS always converged to the same point (which is probably the global minimum); however, with respect to the number of iterations, the SWLS method was still significantly superior than the SFP algorithm. We have also compared the SWLS solution with the SDR solution for the runs in which the SDR solution is tight (about a quarter of the runs, (cf. column 1 in Table 5)). In all of these runs, the SWLS and SDR solutions coincided, i.e., the SWLS method produced the exact ML solution.

The last example shows the attractiveness of SWLS over the LS and SDR approaches.

Example 4.3 (**Comparison with the LS and SDR Estimates**) Here we compare the solution of (1.3) and the solution of SDR with the SWLS solution. The stopping rule for the SWLS method was $\|\nabla f(\mathbf{x}_k)\| < 10^{-5}$. We generated 100 random instances of the source localization problem with five sensors where in each run the sensor locations $\mathbf{a}_j$ and the source location $\mathbf{x}$ were randomly generated from a uniform distribution over the square $[-10, 10]\times[-10, 10]$. The observed distances d_j are given by (1.1) with ε_j being independently generated from a normal distribution with mean zero and standard deviation σ. In our experiments σ takes four different values: $1, 10^{-1}, 10^{-2}$ and 10^{-3}. The numbers in the three right columns of Table 6 are the average of the squared position error $\|\hat{\mathbf{x}} - \mathbf{x}\|^2$ over 100 realizations, where $\hat{\mathbf{x}}$ is the solution by the corresponding method. The best result for each possible value of σ is marked by boldface. From the table below, it is clear that the SWLS algorithm outperforms the LS and SDR methods for all four values of σ.

σ	SDR	LS	SWLS
1e-3	2.4e-6	2.7e-6	**1.5e-6**
1e-2	2.2e-4	1.6e-4	**1.3e-4**
1e-1	2.2e-2	1.9e-2	**1.3e-2**
1e+0	2.2e+0	2.7e+0	**2.0e+0**

Table 6: Mean squared position error of the SDR, LS and SWLS methods

References

[1] A. Beck, P. Stoica, and J. Li. Exact and approximate solution of source localization problems. 2007. To appear in *IEEE Trans. Signal Processing*.

[2] D. P. Bertsekas. *Nonlinear Programming.* Belmont MA: Athena Scientific, second edition, 1999.

[3] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sen. Netw.*, 2(2):188–220, 2006.

[4] K. W. Cheung, W. K. Ma, and H. C. So. Accurate approximation algorithm for TOA-based maximum likelihood mobile location using semidefinite programming. In *Proc. ICASSP*, volume 2, pages 145–148, 2004.

[5] K. W. Cheung, H. C. So, W. K. Ma, and Y. T. Chan. Least squares algorithms for time-of-arrival-based mobile location. *IEEE Trans. Signal Processing*, 52(4):1121–1228, April 2004.

[6] C. Fortin and H. Wolkowicz. The trust region subproblem and semidefinite programming. *Optim. Methods Softw.*, 19(1):41–67, 2004.

[7] H. W. Kuhn. A note on Fermat's problem. *Math. Programming*, 4:98–107, 1973.

[8] J. J. Moré. Generalizations of the trust region subproblem. *Optim. Methods Softw.*, 2:189–209, 1993.

[9] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4(3):553–572, 1983.

[10] L. M. Ostresh. On the convergence of a class of iterative methods for solving the Weber location problem. *Operations Research*, 26(4):597–609, 1978.

[11] J. G. Morris R. F. Love and G. O. Wesolowsky. *Facilities location: Models and Methods*. North-Holland Publishing Co., New York, 1988.

[12] A. H. Sayed, A. Tarighat, and N. Khajehnouri. Network-based wireless location. *IEEE Signal Processing Mag.*, 22(4):24–40, July 2005.

[13] P. Stoica and J. Li. Source localization from range-difference measurements. *IEEE Signal Processing Mag.*, 23:63–65, 69, Nov. 2006.

[14] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11-12:625–653, 1999.

[15] Y. Vardi and C. H. Zhang. A modified weiszfield algorithm for the fermat-weber location problem. *Mathematical Programming Series A*, 90:559–566, 2001.

[16] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43:355–386, 1937.