Digital Signal Processing 139 (2023) 104074

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Direct, fast and convergent solvers for the non-convex and non-smooth TDoA localization problem

Eyal Gur [a,*], Alon Amar [b], Shoham Sabach [a]

[a] Faculty of Data and Decision Sciences, Technion – Israel Institute of Technology, Haifa, 3200003, Israel
[b] Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology, Haifa, 3200003, Israel

ARTICLE INFO

Article history:
Available online 8 May 2023

Keywords:
Global convergence
Non-convex optimization
Non-smooth optimization
Optimization methods
Source localization
Time difference of arrival

ABSTRACT

Time-Difference-of-Arrival (TDoA) Source Localization (SL) consists of determining the position of a target given differences of time measurements of several sensors. The TDoA SL problem is formulated as minimizing a non-convex and non-smooth least squares (LS) optimization problem, which is tackled in the existing literature mostly by solving convex relaxations or by taking the square of the measurements. In this work, we tackle the LS problem directly, by first showing that a standard fixed-point (FP) method can be derived. Due to numerical instability and lack of theoretical convergence guarantees of FP, we develop the T-NAM method. This easy-to-implement and novel method uses the Nested Alternating Minimization scheme together with the fast FISTA algorithm. We show that T-NAM converges to critical points of the original LS function – a result that, to the best of our knowledge, is unknown for the TDoA SL problem. Last, we show numerically the advantages of FP and T-NAM relatively to existing works.

1. Introduction

The Time-Difference-of-Arrival (TDoA) Source Localization (SL) problem appears in a diverse range of applied research fields, such as radar systems, mobile communications, unmanned vehicles, speech processing, and more [1]. The problem consists of determining the location of a radiating source (i.e., the target) given some measurements obtained by an array of sensors. This problem received a lot of attention in the past [2–5] and is still does in recent years [6–9]. We refer the reader to [1] for a description of other types of non-TDoA techniques for SL.

In TDoA localization, the source transmits a signal that is received by an array of time-synchronized sensors. Then, each sensor determines the Time-of-Arrival (ToA) at which it received the signal. Since the sensors and source are not time-synchronized, the ToA measurements depend on the unknown transmission time. To find the location of the source, the fusion center of the array selects a reference sensor from the array, and computes the differences between the ToA measurements of the reference sensor and any other sensor. As all sensors are time-synchronized, these TDoA measurements do not depend on the unknown transmission time.

The TDoA localization problem can be modeled as a system of non-linear equations, where each equation describes the estimated time-difference between the reference sensor and any other sensor. The problem of solving this system can be formulated, in various ways, as optimization problems which are all non-convex and in some cases even non-smooth – and hence are challenging to solve.

One popular formulation of the TDoA SL problem as an optimization problem is the least squares (LS) formulation, which coincides with optimizing the maximum likelihood function under some noise assumptions. This statistical interpretation is a desirable property in localization problems. Unfortunately, the LS problem is non-convex and non-smooth, and its optimal solution cannot be obtained. Hence, many works that tackle the TDoA problem use a different formulation that only approximates the LS function by using convex relaxations or by squaring the measurements (see Section 1.2). For this reason, we develop two iterative methods, which we call FP and T-NAM, that directly tackle the LS problem formulation.

This paper is organized as follows: in the remainder of Section 1 we present the LS formulation and related works. In Sections 2 and 3, we develop our FP and T-NAM algorithms, while in Section 4 we prove that T-NAM converges to critical points. In Section 5, we conduct numerical experiments, and in Section 6 we conclude this work.

1.1. Least squares problem formulation

Consider an array of N sensors, each located at the known location $\mathbf{p}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$. The true unknown location of the

[*] Corresponding author.
E-mail address: eyal.gur@campus.technion.ac.il (E. Gur).

https://doi.org/10.1016/j.dsp.2023.104074
1051-2004/© 2023 Elsevier Inc. All rights reserved.

source is denoted by $\mathbf{s} \in \mathbb{R}^n$. The unknown transmission time of the signal by the source is denoted by $\tilde{T}$. The ToA of the signal at sensor $i = 1, 2, \ldots, N$ is the total travel duration of the signal from the source to the sensor with the unknown transmission time, i.e.,

$$t_i = \frac{1}{c} \|\mathbf{s} - \mathbf{p}_i\| + \tilde{T} + \tilde{\epsilon}_i,$$

where $c > 0$ is a known propagation speed of the signal in the environment of the problem, $\tilde{\epsilon}_i \in \mathbb{R}$ is some noise, and $\|\mathbf{s} - \mathbf{p}_i\| \geq 0$ is the Euclidean distance between the source and the i-th sensor in the array.

Assuming all measurements are collected by a central processor, then time-differences with respect to some reference sensor can be obtained. For example, if the reference sensor is $j \in \{1, 2, \ldots, N\}$, then for any other sensor $i = 1, 2, \ldots, N$, $i \neq j$, we obtain the TDoA measurement

$$\tilde{d}_{i,j} \equiv \frac{1}{c} \|\mathbf{s} - \mathbf{p}_i\| - \frac{1}{c} \left\|\mathbf{s} - \mathbf{p}_j\right\| + \tilde{\epsilon}_i - \tilde{\epsilon}_j. \tag{1}$$

The TDoA performance may change due to a different selection of reference sensor [10], and several papers suggest approaches that consider the complete set of time-differences [9,11,12], or approaches that select the best $N - 1$ measurements out of this complete set [13]. Still, as the selection of the reference sensor is a degree of freedom of the problem, herein we focus on one specific selection. In this work, we choose a reference sensor $j \in \{1, 2, \ldots, N\}$ such that $\tilde{d}_{i,j} \geq 0$ for any $i \neq j$, simply by choosing $j = \operatorname{argmin}\{t_i : i = 1, 2, \ldots, N\}$. Therefore, we can assume without the loss of generality that sensor 1 is the chosen reference sensor and that $\tilde{d}_{i,1} \geq 0$ for any $i = 2, 3, \ldots, N$. Multiplying $\tilde{d}_{i,1}$ in (1) by $c > 0$, we denote by d_i the noisy time-difference measurement between sensor i and the reference sensor 1 [14,3,10]. Meaning, for any $i = 2, 3, \ldots, N$

$$d_i \equiv c\tilde{d}_{i,1} = \|\mathbf{s} - \mathbf{p}_i\| - \|\mathbf{s} - \mathbf{p}_1\| + \epsilon_i, \tag{2}$$

where $\epsilon_i \equiv c\left(\tilde{\epsilon}_i - \tilde{\epsilon}_1\right)$. The SL problem aims at finding the unknown source location $\mathbf{s} \in \mathbb{R}^n$ that satisfies the set of $N - 1$ non-convex and non-smooth equations given in (2).

To find a solution of this set of equations, we formulate a suitable optimization problem. If the distribution of each noise ϵ_i is known, a possible approach is the Maximum-Likelihood Estimator (MLE) approach, which is asymptotically unbiased, and approaches the Cramér-Rao Lower Bound (CRLB) as N increases [15]. The MLE approach coincides with the weighted least squares optimization problem, where the weighting matrix is the inverse of the covariance noise matrix. However, in practice, the distribution of the noises is unknown (since the correlation between the noises ϵ_i and ϵ_j, $i, j \in \{2, 3, \ldots, N\}$, is unknown), hence a common approach is to take the weighting matrix as the identity matrix, which results in the following least squares problem formulation [3]

$$\min_{\mathbf{s} \in \mathbb{R}^n} \quad F(\mathbf{s}) \equiv \frac{1}{2} \sum_{i=2}^{N} f_i^2(\mathbf{s}), \tag{LS}$$

where we denote $f_i(\mathbf{s}) \equiv \|\mathbf{s} - \mathbf{p}_i\| - \|\mathbf{s} - \mathbf{p}_1\| - d_i$ for any $i = 2, 3, \ldots, N$. Tackling Problem (LS) do not require any prior knowledge of the noise covariance matrix.

Problem (LS) is non-convex and non-smooth, thus finding a closed-form solution is not possible. Therefore, in this paper, we focus on developing iterative algorithms that converge to critical points of the LS objective function F. Here, by critical point we mean a point $\mathbf{s} \in \mathbb{R}^n$ satisfying $\mathbf{0}_n \in \partial F(\mathbf{s})$. Since we are dealing with non-convex and non-smooth functions, when we write ∂F we mean the limiting sub-differential set of F [16].

We mention that criticality of a point is a necessary condition for its optimality. In the world of non-convex and non-smooth optimization, the state-of-the-art in recent years is designing algorithms with proved global convergence guarantees to critical points. Here, by global convergence we mean that the entire sequence generated by an algorithm converges to a unique point, which is a stronger result than sub-sequence or function value convergence (which are the only known convergence guarantees of previous works, as discussed below in Section 1.2).

1.2. *Existing works*

Works that tackle Problem (LS) can be categorized into three: (i) convex relaxations, (ii) squaring the measurements, and (iii) directly solving Problem (LS). While (i) and (ii) can be sometimes solved explicitly, their solution is not optimal in the LS sense. Therefore, there is an advantage in tackling Problem (LS) directly. We now discuss the three categories and survey some related works.

1.2.1. *Convex relaxations*

A popular approach for tackling the non-convexity of Problem (LS), is to solve a convex relaxation of the problem. For example, in the recent paper [17], the authors add a vector variable $\mathbf{q} \in \mathbb{R}^N$ such that each unknown distance $\|\mathbf{s} - \mathbf{p}_i\|$ is replaced with $\mathbf{q}_i \in \mathbb{R}$ to obtain a quadratic problem with non-linear equality constraints. Then, the problem is relaxed into a convex semi-definite programming (SDP) problem. In [18], the authors relax these non-linear constraints into ball constraints, to obtain a second-order cone programming (SOCP) problem that is solved using a two-stage algorithm. First, the relaxed problem is solved using an SDP solver, and then this solution is used as a starting point of a non well-defined Newton-type method to solve the original non-relaxed formulation. This two-stage algorithm performs well in small arrays of 4–6 sensors, with estimates that are close to the CRLB.

Additional convex relaxation works are, for example, [19–24]. Similarly, these works consider a non-linear equality constrained problem that is relaxed into a convex SDP or SOCP problem. Even though all SDP and SOCP problems have a global minimizer that can be found using off-the-shelf SDP solvers [25], these solvers suffer from an increased computational complexity and thus are limited to small-scale array geometries.

Not all convex relaxations are solved using SDP solvers. In the well-known paper [26], the involved equations are linearized using Taylor's series keeping only terms below second-order. The resulting linear problem is minimized by simple update steps. In [27], a Levenberg-Marquardt method that is based on second-order Taylor's series approximation is developed. However, in some array geometries, the Taylor's series methods fail to converge [28].

While all convex relaxations posses a global minimum and have no locally optimal solutions, there is no guarantee that this optimal solution is even a critical point of the original LS problem (see [29]). Similar aspects and difficulties arise when dealing with other variants of localization problems. For instance, see [29] and [30] for recent papers that tackle the Sensor Network Localization problem.

1.2.2. *Squaring the measurements*

Alternative formulations of the TDoA problem overcome its non-smoothness by squaring both sides of the equations in (2). For instance, one can consider the following set of $N - 1$ non-convex and smooth equations

$$(d_i + \|\mathbf{s} - \mathbf{p}_1\|)^2 = \|\mathbf{s} - \mathbf{p}_i\|^2 + \nu_i, \quad i = 2, 3, \ldots, N, \tag{3}$$

where $\nu_i \in \mathbb{R}$ is a noise that depends on the original noise ϵ_i and on the true distance between the source and the sensor $\|\mathbf{s} - \mathbf{p}_i\|$.

This squaring idea paves the way for iterative techniques from the world of smooth optimization in order to solve (3) [31,32]. It is also possible to utilize convex relaxations, as was done, for example, in [20,33,24].

In some cases, it is possible to explicitly find an optimal solution. Notice that (3) can equivalently be written as

$$2\left(\mathbf{p}_i-\mathbf{p}_1\right)^T\mathbf{s}+2d_ir=\|\mathbf{p}_i\|^2-\|\mathbf{p}_1\|^2-d_i^2+\nu_i,$$

where $r=\|\mathbf{s}-\mathbf{p}_1\|$. Then, a globally optimal solution with respect to $\mathbf{s}\in\mathbb{R}^n$ and $r\in\mathbb{R}$ can be found using classical linear regression solvers.

In the celebrated paper [3], the authors suggest a simple two-step algorithm. The first step estimates the true covariance matrix of the measurements. In the second step, the estimated matrix is used to obtain an estimate of a globally optimal solution of (3) via weighted LS (WLS) techniques. Algorithm WLS of [3] was later extended to tackle more complex array geometries in [18,34] via constrained WLS (CWLS) techniques, and also by additionally separating the variables (SCWLS) as in [32]. Additional works are, for example, [35,6,11,5,28,36,10] which all suggest algorithms for finding optimal solutions of different squaring reformulations.

However, as mentioned above, by squaring the measurements the noise terms now depend on the original noises as well as on the unknown distances. Hence, these optimal solutions only approximate the LS solution for small enough noises [3]. In addition, there is no guarantee that these optimal solutions are even critical points of the original LS formulation, hence all lack a statistical interpretation. Moreover, as was suggested in [5], these solutions might yield a less accurate location estimate compared to the original non-smooth LS problem formulation.

1.2.3. Solving Problem (LS) directly

To date, only a few works have designed algorithms that tackle the non-convex and non-smooth Problem (LS) directly, in the sense that they do not solve some convex relaxation nor take the square of the measurements. In [12,37], the authors utilize the Majorization-Minimization (MM) scheme by introducing upper bounds for which a closed-form formula of their minimizer can be obtained. It was shown in [12] that their algorithm, called SOLVIT, uses a tighter upper bound than the one of [37]. Moreover, SOLVIT has a faster convergence rate in terms of function values. In addition, it was proved in [12] that any convergent sub-sequence generated by SOLVIT converges to critical points of Problem (LS). However, the updates of SOLVIT require calculating the inverse of a certain matrix, which increases its computational complexity and may lead to numerical instabilities. Moreover, SOLVIT is not well-defined over the sensor locations.

Some heuristic algorithms for tackling Problem (LS) also exist, such as [38,39] that suggest a hybrid method that uses the WLS method of [3] with some stochastic genetic optimization algorithm. However, these heuristic approaches have either weak or no theoretical guarantees on the convergence of the generated sequence.

1.3. Our contributions

Motivated by the limitations discussed above, in this work we aim at developing fast algorithms that directly tackle the original non-convex and non-smooth Problem (LS). The main results of this paper are summarized now.

1. We develop two novel and simple iterative solvers, called FP and T-NAM, that directly tackle Problem (LS). In contrary to some previous works, these algorithms are suitable for both high and low noises.
2. T-NAM and FP only use first-order information (as contrary to SDP solvers), and as such are fast, consist of simple updates, and enjoy low computational and storage burden.
3. The sequence of iterations generated by T-NAM is proven to globally converge to critical points of Problem (LS). To the best of our knowledge, T-NAM is the first algorithm that globally converges to critical points of the original TDoA LS formulation.

Table 1
Comparison of selected algorithms.

Method	Problem	Computational Cost	Convergence
T-NAM	(LS)	$\mathcal{O}\left(n^2+nN\right)$	✓
FP	(LS)	$\mathcal{O}\left(nN\right)$	✗
SOLVIT [12]	(LS)	$\mathcal{O}\left(n^3+n^2N\right)$	✗
WLS [3]	(3)	$\mathcal{O}\left(n^3+N^3\right)$	✗
SCWLS [32]	(3)	$\mathcal{O}\left(n^3+N^3\right)$	✗
SDP [17]	(LS) relaxation	$\mathcal{O}\left(n^2N^5+N^9\right)$	✗

In Table 1, we summarize properties of several selected algorithms. In Section 5, their performance is compared by conducting numerical experiments. The table specifies which formulation of the TDoA problem each method tackles, the total computational cost (see complexity analysis in Section 5.1), and whether or not the method is proved to converge to critical points of the problem.

As can be seen in Table 1, T-NAM is the only algorithm that enjoys a quadratic computational cost with proved convergence to critical points of the original Problem (LS). In addition, FP enjoys a very low computational burden compared to all other methods.

2. Fixed-point (FP) algorithm for TDoA

In this section, we develop a simple iterative fixed-point method that directly tackles Problem (LS). To this end, under the assumption that a minimizer $\mathbf{s}\in\mathbb{R}^n$ of Problem (LS) is not located at a sensor (i.e., $\mathbf{s}\neq\mathbf{p}_i$ for all $i=1,2,\ldots,N$), then $\nabla F\left(\mathbf{s}\right)$ exists and it holds that $\nabla F\left(\mathbf{s}\right)=\mathbf{0}_n$ (recall that F is the objective of Problem (LS)). To find an update formula, we write the gradient

$$\nabla F\left(\mathbf{s}\right)=\sum_{i=2}^{N}f_i\left(\mathbf{s}\right)\cdot\left(\frac{\mathbf{s}-\mathbf{p}_i}{\|\mathbf{s}-\mathbf{p}_i\|}-\frac{\mathbf{s}-\mathbf{p}_1}{\|\mathbf{s}-\mathbf{p}_1\|}\right).$$

After simple algebraic manipulations we see that

$$\nabla F\left(\mathbf{s}\right)=2\left(N-1\right)\mathbf{s}-\mathbf{p}-\sum_{i=1}^{N}\mathbf{w}_i\left(\mathbf{s}\right)\cdot\frac{\mathbf{s}-\mathbf{p}_i}{\|\mathbf{s}-\mathbf{p}_i\|},$$

where we define

$$\mathbf{w}_1\left(\mathbf{s}\right)\equiv\sum_{i=2}^{N}\left(\|\mathbf{s}-\mathbf{p}_i\|-d_i\right)\in\mathbb{R},$$

$$\mathbf{w}_i\left(\mathbf{s}\right)\equiv\|\mathbf{s}-\mathbf{p}_1\|+d_i\in\mathbb{R},\quad\forall i=2,3,\ldots,N,$$

and $\mathbf{p}\equiv\sum_{i=2}^{N}\left(\mathbf{p}_1+\mathbf{p}_i\right)\in\mathbb{R}^n$.

Since we require that $\nabla F\left(\mathbf{s}\right)=\mathbf{0}_n$, then a standard fixed-point algorithm for the TDoA problem, which starts at any $\mathbf{s}^0\in\mathbb{R}^n$, is given by the following update rule

$$\mathbf{s}^{k+1}=\frac{1}{2\left(N-1\right)}\left(\mathbf{p}+\sum_{i=1}^{N}\mathbf{w}_i\left(\mathbf{s}^k\right)\frac{\mathbf{s}^k-\mathbf{p}_i}{\|\mathbf{s}^k-\mathbf{p}_i\|}\right),\tag{4}$$

for any iteration $k\geq 0$. The above updates are simple with low computational complexity (see Section 5.1). However, they are not defined over $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$, and hence there is no theoretical guarantee that they are well-defined for any starting point $\mathbf{s}^0 \in \mathbb{R}^n$.

Surprisingly, such a simple FP method was not found in the existing literature. In the scope of FP algorithms that tackle localization problems, we mention the well-known Weiszfeld's FP method [40], that tackles the convex Fermat-Weber problem (see [41] for a review paper). It was shown in [42] that Weiszfeld's method is no more than a Gradient Descent (GD) method. In fact, this is also the case for our FP method in (4). Indeed, it is easy to show that the RHS of (4) can be written equivalently as

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \frac{1}{2(N-1)} \nabla F\left(\mathbf{s}^k\right), \tag{5}$$

which is a GD step with a fixed step-size $1/2(N-1)$. Iterations of the GD method can be implemented in a distributed and parallel fashion among the sensors, hence this method can be used in any array architecture.

The fact that the update step in (5) is a GD step, may lead to the wrong interpretation that the updates converge to a critical point of Problem (LS). Recall that the convergence guarantees of GD are only applicable for continuously differentiable functions, while the function F of Problem (LS) is non-smooth. Moreover, since F is additionally non-convex, in order to guarantee convergence of the GD step, the step-size should be strictly smaller than $2/L$, where L is the Lipschitz constant of ∇F. However, since the function ∇F is not continuous, the constant L is not even well-defined (see, for example, [43]).

Another work that suggests a fixed-point method to solve a localization problem is [44]. In this work, the authors address the task of source localization using ToA measurements, which is formulated as

$$\min_{\mathbf{s} \in \mathbb{R}^n} \quad \sum_{i=1}^{N} \left(\|\mathbf{s} - \mathbf{p}_i\| - d_i\right)^2. \tag{ToA}$$

Interestingly, even though Problems (LS) and (ToA) are two different optimization problems, it was shown in [44] that a standard FP method for Problem (ToA) is also a GD method with a similar step-size to the one in (5). It was also proved in [44] that for certain initialization points, the FP method for Problem (ToA) generates a bounded sequence for which any limit point is a critical point of the corresponding objective function. This convergence analysis in [44] relies on the fact that the objective in Problem (ToA) is coercive. However, since the objective of Problem (LS) is not coercive (see, for example, [12]), this convergence analysis is not applicable for Problem (LS).

In the view of the two limitations of the updates of the FP method in (5), which are lack of convergence guarantees and the fact that the iterates are not necessarily well-defined (which may lead to numerical instability), in the next section we develop the T-NAM algorithm.

3. Nested alternating minimization for TDoA

In this section, we develop the second and main algorithm of this work, which we call T-NAM. Like the FP method from Section 2, T-NAM also directly tackles the non-convex and non-smooth Problem (LS). In contrary to FP, we will prove that T-NAM always generates well-defined iterates (regardless of the initialization point), and more importantly, we will prove that T-NAM always globally converges to critical points of Problem (LS). In addition, we show that T-NAM consists of simple steps that are all given by explicit and inexpensive operations.

To achieve this, we reformulate Problem (LS) as an equivalent problem. While the reformulation remains non-convex and non-smooth, it admits a structure that allows us to develop T-NAM by utilizing the recent Nested Alternating Minimization (NAM) optimization scheme of [45].

The idea behind the NAM scheme, like the original Alternating Minimization scheme, is decomposing the problem into smaller-scale sub-problems that are minimized sequentially using some selected nested sub-algorithm. In this work, we use the accelerated and non-descent FISTA method of [46] as the nested sub-algorithm in NAM.

3.1. Equivalent reformulation of Problem (LS)

Notice that each function $f_i \colon \mathbb{R}^n \to \mathbb{R}$, $i = 2, 3, \ldots, N$, as was defined in Problem (LS), satisfies

$$\begin{aligned} f_i^2(\mathbf{s}) = {} & 2\|\mathbf{s}\|^2 - 2(\mathbf{p}_1 + \mathbf{p}_i)^T \mathbf{s} + 2d_i \|\mathbf{s} - \mathbf{p}_1\| \\ & - 2\|\mathbf{s} - \mathbf{p}_i\| \cdot \|\mathbf{s} - \mathbf{p}_1\| - 2d_i \|\mathbf{s} - \mathbf{p}_i\| \\ & + \|\mathbf{p}_i\|^2 + \|\mathbf{p}_1\|^2 + d_i^2. \end{aligned} \tag{6}$$

We denote by $\mathcal{B}$ and $\mathcal{S}$ the unit ball and sphere of $\mathbb{R}^n$ centered at $\mathbf{0}_n$ with radius 1, respectively. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ it holds from the Cauchy-Schwartz inequality that

$$-\|\mathbf{x}\| = \min_{\mathbf{u}_1 \in \mathcal{B}} \{-\|\mathbf{u}_1\| \cdot \|\mathbf{x}\|\} \le \min_{\mathbf{u}_1 \in \mathcal{B}} \left\{\mathbf{u}_1^T \mathbf{x}\right\}, \tag{7}$$

and

$$\begin{aligned} -\|\mathbf{x}\| \cdot \|\mathbf{y}\| &= \min_{\mathbf{u}_1, \mathbf{u}_i \in \mathcal{B}} \{-\|\mathbf{u}_1\| \cdot \|\mathbf{x}\| \cdot \|\mathbf{u}_i\| \cdot \|\mathbf{y}\|\} \\ &\le \min_{\mathbf{u}_1, \mathbf{u}_i \in \mathcal{B}} \left\{\mathbf{u}_1^T \mathbf{x} \cdot \mathbf{u}_i^T \mathbf{y}\right\}. \end{aligned} \tag{8}$$

Plugging $\mathbf{x} = \mathbf{s} - \mathbf{p}_i$ and $\mathbf{y} = \mathbf{s} - \mathbf{p}_1$ in (7) and (8), we obtain from (6) with any $\mathbf{u}_i \in \mathcal{B}$, $i = 1, 2, \ldots, N$, that

$$\begin{aligned} \frac{1}{2} f_i^2(\mathbf{s}) \le {} & \|\mathbf{s}\|^2 - (\mathbf{p}_1 + \mathbf{p}_i)^T \mathbf{s} + d_i \|\mathbf{s} - \mathbf{p}_1\| \\ & + \mathbf{u}_i^T (\mathbf{s} - \mathbf{p}_i) \cdot \mathbf{u}_1^T (\mathbf{s} - \mathbf{p}_1) + d_i \mathbf{u}_i^T (\mathbf{s} - \mathbf{p}_i) \\ & + \frac{1}{2}\left(\|\mathbf{p}_i\|^2 + \|\mathbf{p}_1\|^2 + d_i^2\right). \end{aligned} \tag{9}$$

Notice that equality is achieved in (9) by taking the unit sphere vectors

$$\bar{\mathbf{u}}_1(\mathbf{s}) = \begin{cases} \frac{\mathbf{s} - \mathbf{p}_1}{\|\mathbf{s} - \mathbf{p}_1\|}, & \mathbf{s} \neq \mathbf{p}_1, \\ \mathbf{e}_1, & \mathbf{s} = \mathbf{p}_1, \end{cases} \tag{10}$$

and (for any $i = 2, 3, \ldots, N$)

$$\bar{\mathbf{u}}_i(\mathbf{s}) = \begin{cases} -\frac{\mathbf{s} - \mathbf{p}_i}{\|\mathbf{s} - \mathbf{p}_i\|}, & \mathbf{s} \neq \mathbf{p}_i, \\ \mathbf{e}_1, & \mathbf{s} = \mathbf{p}_i, \end{cases} \tag{11}$$

where $\mathbf{e}_1 \equiv (1, 0, \ldots, 0) \in \mathcal{S}$ (though any other vector in the unit ball can be taken).

Now, we define the non-convex and quadratic function $\varphi \colon \mathbb{R}^n \times \mathcal{B}^N \to \mathbb{R}$ as

$$\begin{aligned} \varphi(\mathbf{s}, \mathbf{u}) = {} & (N-1)\|\mathbf{s}\|^2 - \sum_{i=2}^{N} (\mathbf{p}_1 + \mathbf{p}_i)^T \mathbf{s} \\ & + \sum_{i=2}^{N} \mathbf{u}_i^T (\mathbf{s} - \mathbf{p}_i) \left(\mathbf{u}_1^T (\mathbf{s} - \mathbf{p}_1) + d_i\right), \end{aligned} \tag{12}$$

where $\mathbf{u} \in \mathcal{B}^N$ is the concatenation of the vectors $\mathbf{u}_i \in \mathcal{B}$, $i = 1, 2, \ldots, N$, into a single column vector, and $\mathcal{B}^N$ is the Cartesian product of N unit balls. In addition, we define the convex and non-smooth function

$$\psi(\mathbf{s}) \equiv \sum_{i=2}^{N} d_i \left\| \mathbf{s} - \mathbf{p}_1 \right\|. \tag{13}$$

Summing inequality (9) for all $i = 2, 3, \ldots, N$, we get from (12) and (13) that

$$F(\mathbf{s}) \leq \Psi(\mathbf{s}, \mathbf{u}), \quad \mathbf{u} \in \mathcal{B}^N, \tag{14}$$

where $\Psi(\mathbf{s}, \mathbf{u}) \equiv \varphi(\mathbf{s}, \mathbf{u}) + \psi(\mathbf{s}) + C$, and where $C \geq 0$ is some constant independent of $\mathbf{s}$ and $\mathbf{u}$.

From (10) and (11) we get that for any $\mathbf{s} \in \mathbb{R}^n$, equality in (14) is attained for $\mathbf{u} = \bar{\mathbf{u}}(\mathbf{s}) \in \mathcal{S}^N$ (where $\bar{\mathbf{u}}(\mathbf{s})$ is the concatenation of all $\bar{\mathbf{u}}_i$, $i = 1, 2, \ldots, N$, into a single column vector). Meaning,

$$F(\mathbf{s}) = \Psi(\mathbf{s}, \bar{\mathbf{u}}(\mathbf{s})). \tag{15}$$

Finally, we obtain from (15) that Problem (LS) can be written equivalently as

$$\min_{\mathbf{s} \in \mathbb{R}^n} \left\{ F(\mathbf{s}) = \Psi(\mathbf{s}, \bar{\mathbf{u}}(\mathbf{s})) \right\}. \tag{LS}$$

In the next sub-section, we develop the T-NAM algorithm for tackling Problem (LS).

3.2. The T-NAM algorithm

In (14), we established that for any $\mathbf{s} \in \mathbb{R}^n$, the function $F(\mathbf{s})$ of Problem (LS) is bounded from above by the function $\Psi(\mathbf{s}, \mathbf{u})$, for any $\mathbf{u} \in \mathcal{B}^N$, and this upper bound is attained for $\mathbf{u} = \bar{\mathbf{u}}(\mathbf{s})$. Hence, the task of minimizing F can be viewed as minimizing Ψ in an alternating fashion: first, update the vector $\bar{\mathbf{u}}(\mathbf{s}) \in \mathcal{S}^N$, then minimize Ψ with respect to $\mathbf{s} \in \mathbb{R}^n$ using some nested sub-algorithm, and repeat the process until a stopping criterion is satisfied (this is the so-called NAM scheme of [45]).

To minimize Ψ with respect to $\mathbf{s} \in \mathbb{R}^n$, we see that the function $\mathbf{s} \mapsto \Psi(\mathbf{s}, \bar{\mathbf{u}})$ is the sum of the non-smooth norm term $\psi(\mathbf{s})$ and the smooth quadratic function $\mathbf{s} \mapsto \varphi(\mathbf{s}, \bar{\mathbf{u}})$ (see (12)–(14)). Therefore, a natural candidate to minimize $\mathbf{s} \mapsto \Psi(\mathbf{s}, \bar{\mathbf{u}})$ is the well-known FISTA method of [46]. More precisely, in any iteration $k \geq 0$ of NAM, we apply (inner) iterations of FISTA as a nested algorithm.

Now, we are ready to present our T-NAM algorithm (see Algorithm 1), which is based on the NAM scheme with FISTA as a nested sub-algorithm.

Algorithm 1 T-NAM (NAM for TDoA).

```
1: Initialization: $\mathbf{s}^0 \in \mathbb{R}^n$.
2: for $k \geq 0$ do
3:     Update $\mathbf{u}_i^k \equiv \bar{\mathbf{u}}_i(\mathbf{s}^k)$, for all $i = 1, 2, \ldots, N$, according to (10) and (11).
4:     Update $\mathbf{s}^{k+1}$ by applying $j_k \in \mathbb{N}$ iterations of FISTA to minimize the function
       $\mathbf{s} \mapsto \Psi(\mathbf{s}, \mathbf{u}^k)$ (see Algorithm 2 below), starting with $\mathbf{s}^k$.
5: end for
```

Remark 1.

(i) In step 3 of T-NAM, we see that $\mathbf{u}_i^k$, for any $i = 1, 2, \ldots, N$ and $k \geq 0$, has a simple and computationally inexpensive closed-form update formula that depends only on the previous iteration $\mathbf{s}^k$. Therefore, T-NAM can be written only in terms of the original variable $\mathbf{s} \in \mathbb{R}^n$, simply by substituting the auxiliary vectors $\mathbf{u}_i^k$ in the update of $\mathbf{s}^{k+1}$.

(ii) The point $\mathbf{s}^0 \in \mathbb{R}^n$, which is the initialization point of T-NAM, can be set arbitrarily.

(iii) For any (outer) iteration $k \geq 0$, the number of inner FISTA iterations, denoted by $j_k \in \mathbb{N}$ in step 4 of Algorithm 1, can be set arbitrarily by the user. However, in order to obtain convergence guarantees of T-NAM (see Section 4), we use the following number of inner iterations (see [45, Section 4.2.1])

$$j_k = s + 2^{\lfloor k/r \rfloor} - 1, \tag{16}$$

for any integers $s, r \in \mathbb{N}$. We point out that this update rule is quite general due to the flexibility in choosing the parameters s and r.

Developing explicit FISTA update steps.

For any (outer) iteration $k \geq 0$ of T-NAM, an explicit update formula of the FISTA iterations (see step 4 in Algorithm 1), are presented below in Algorithm 2, and are developed now. Recall that

$$\Psi\left(\mathbf{s}, \mathbf{u}^k\right) = \varphi\left(\mathbf{s}, \mathbf{u}^k\right) + \psi(\mathbf{s}) + C. \tag{17}$$

We denote by $\left\{\mathbf{s}^{k,j}\right\}_{j=0}^{j_k}$ the sequence generated by FISTA in the outer iteration $k \geq 0$ of T-NAM. Under this notation, the starting point of FISTA is set to be $\mathbf{s}^{k,0} = \mathbf{s}^k$, and the output point obtained after $j_k \in \mathbb{N}$ inner iterations is $\mathbf{s}^{k,j_k} = \mathbf{s}^{k+1}$. In addition, we denote by $\left\{\mathbf{y}^{k,j}\right\}_{j \geq 0}$ the auxiliary sequence generated by FISTA (see step 5 in Algorithm 2 below for the exact details). Following [46], at each inner iteration $j \geq 0$ of FISTA to minimize the partial function $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \mathbf{u}^k\right)$ (see (17)), the next iteration $\mathbf{s}^{k,j+1}$ is updated by calculating the prox-grad step

$$\mathbf{s}^{k,j+1} = \operatorname*{argmin}_{\mathbf{s} \in \mathbb{R}^n} \left\{ \psi(\mathbf{s}) + \frac{L^k}{2} \left\| \mathbf{s} - \mathbf{c}^{k,j} \right\|^2 \right\}, \tag{18}$$

where $\mathbf{c}^{k,j} \equiv \mathbf{y}^{k,j} - \frac{1}{L^k} \nabla_{\mathbf{s}} \varphi\left(\mathbf{y}^{k,j}, \mathbf{u}^k\right) \in \mathbb{R}^n$, and $L^k \geq 0$ is the Lipschitz constant of the gradient of the partial function $\mathbf{s} \mapsto \varphi\left(\mathbf{s}, \mathbf{u}^k\right)$.

In order to find an explicit formula for the update step in (18), we define the matrices $\tilde{\mathbf{A}}^k, \mathbf{A}^k \in \mathbb{R}^{n \times n}$ as

$$\begin{aligned} \tilde{\mathbf{A}}^k &\equiv (N-1)\mathbf{I}_n + \left(\sum_{i=2}^{N} \mathbf{u}_i^k\right)\left(\mathbf{u}_1^k\right)^T, \\ \mathbf{A}^k &\equiv \frac{1}{2}\left(\tilde{\mathbf{A}}^k + \left(\tilde{\mathbf{A}}^k\right)^T\right), \end{aligned} \tag{19}$$

and the vector $\mathbf{z}^k \in \mathbb{R}^n$ as

$$\mathbf{z}^k \equiv \sum_{i=2}^{N} \left(\mathbf{p}_1 + \mathbf{p}_i + \mathbf{p}_i^T \mathbf{u}_i^k \mathbf{u}_1^k + \mathbf{p}_1^T \mathbf{u}_1^k \mathbf{u}_i^k - d_i \mathbf{u}_i^k\right). \tag{20}$$

After some algebraic manipulations we see that

$$\varphi\left(\mathbf{s}, \mathbf{u}^k\right) = \mathbf{s}^T \mathbf{A}^k \mathbf{s} - \left(\mathbf{z}^k\right)^T \mathbf{s} + \sum_{i=2}^{N} \mathbf{p}_i^T \mathbf{u}_i^k \left(\mathbf{p}_1^T \mathbf{u}_1^k - d_i\right),$$

and hence

$$\nabla_{\mathbf{s}} \varphi\left(\mathbf{s}, \mathbf{u}^k\right) = 2\mathbf{A}^k \mathbf{s} - \mathbf{z}^k, \tag{21}$$

where $\nabla_{\mathbf{s}} \varphi$ denotes the gradient of φ only with respect to the variable $\mathbf{s}$. From the definition of $\psi(\mathbf{s})$ (see (13)) and by setting the auxiliary variable $\mathbf{x} = \mathbf{s} - \mathbf{p}_1$ in (18), we obtain from (21), after eliminating some constant terms, that (18) can be written as

$$\mathbf{s}^{k,j+1} = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \frac{d}{L^k} \|\mathbf{x}\| + \frac{1}{2} \left\| \mathbf{x} - \mathbf{a}^{k,j} \right\|^2 \right\} + \mathbf{p}_1,$$

where we denote $d \equiv \sum_{i=2}^{N} d_i$, and

$$\mathbf{a}^{k,j} \equiv -\frac{2}{L^k} \mathbf{A}^k \mathbf{y}^{k,j} + \mathbf{y}^{k,j} + \frac{1}{L^k} \mathbf{z}^k - \mathbf{p}_1 \in \mathbb{R}^n. \tag{22}$$

Following [47, Example 6.19], we finally obtain that

$$\mathbf{s}^{k,j+1} = \begin{cases} \frac{L^k \left\| \mathbf{a}^{k,j} \right\| - d}{L^k \left\| \mathbf{a}^{k,j} \right\|} \mathbf{a}^{k,j} + \mathbf{p}_1, & L^k \left\| \mathbf{a}^{k,j} \right\| \geq d, \\ \mathbf{p}_1, & L^k \left\| \mathbf{a}^{k,j} \right\| < d. \end{cases} \tag{23}$$

Summarizing the above derivations, the FISTA algorithm for minimizing the partial function $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \mathbf{u}^k\right)$, for any outer iteration $k \geq 0$ (see step 4 in Algorithm 1), is recorded in Algorithm 2.

Algorithm 2 FISTA.

```
1: **Input:** $\mathbf{A}^k$ and $\mathbf{z}^k$ as defined in (19) and (20). Set $L^k = 2\left\|\mathbf{A}^k\right\|$, $\mathbf{s}^{k,0} = \mathbf{y}^{k,0} = \mathbf{s}^k \in \mathbb{R}^n$ and $t_0 = 1$.
2: **for** $j = 0, 1, \ldots, j_k - 1$ **do**
3:     Set $\mathbf{a}^{k,j}$ according to (22) and then update $\mathbf{s}^{k,j+1}$ according to (23).
4:     Set $t_{j+1} = \frac{1+\sqrt{1+4t_j^2}}{2}$.
5:     Set $\mathbf{y}^{k,j+1} = \mathbf{s}^{k,j+1} + \frac{t_j - 1}{t_{j+1}}\left(\mathbf{s}^{k,j+1} - \mathbf{s}^{k,j}\right)$.
6: **end for**
```

Remark 2.

(i) For any (outer) iteration $k \geq 0$ of T-NAM, the update of $\mathbf{s}^{k+1}$ (which is the output of Algorithm 2), has a simple and computationally inexpensive closed-form update steps that depend only on the previous iteration $\mathbf{s}^k$ (which is the starting point of Algorithm 2).

(ii) As for L^k that is used in step 1 of Algorithm 2, from (21) it follows that $L^k = 2\left\|\mathbf{A}^k\right\| = 2\lambda_{\max}\left(\mathbf{A}^k\right)$ (twice the maximal eigenvalue of $\mathbf{A}^k$). In Lemma 1 below, we prove that $\lambda_{\max}\left(\mathbf{A}^k\right) \leq 2(N-1)$, and therefore one can set $L^k = 4(N-1)$ for any $k \geq 0$, which is a distributed Lipschitz constant that does not require calculating the eigenvalues of $\mathbf{A}^k$.

4. Global convergence of T-NAM

In this section, we prove that T-NAM globally converges to critical points of the original non-convex and non-smooth Problem (LS). The global convergence result is established using a proof methodology that was recently developed in [45]. This methodology requires the generated sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ to satisfy three properties (which are formulated mathematically below): (i) decrease in function value up to error, (ii) sub-gradient bound by the iterate gap up to error, and (iii) summable errors. It was proved in [45] that when these properties are satisfied, the sequence globally converges to critical points of the function at hand. In the context of T-NAM, this result is stated now in Theorem 1.

Theorem 1. *Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by T-NAM. Then, it converges to some $\mathbf{s}^* \in \mathbb{R}^n$, which is a critical point of the function F of Problem (LS).*

Remark 3. Theorem 1 requires $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ to be bounded. Since F is not coercive (see [12]), there is no guarantee that it is indeed bounded for any starting point $\mathbf{s}^0$. However, this requirement fails to hold only if the generated locations diverge to infinity – which is unlikely to happen in real-life scenarios. We should mention that a variant of T-NAM, in which the FISTA iterates are projected onto some ball that contains the environment of the problem, can be easily derived to obtain a bounded sequence with the same theoretical convergence guarantees.

In order to prove Theorem 1, throughout this section we make the following mild assumption regarding the sensor locations $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N \in \mathbb{R}^n$.

Assumption 1.

(a) $N \geq 2$.
(b) $\mathbf{p}_i \neq \mathbf{p}_j$ for all $i \neq j$.
(c) The sensors are not co-linear (and if $N = 2$, then the sensors and source are not co-linear).

In the following lemma we prove several results about the sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ generated by T-NAM, that will later enable us to prove Theorem 1. Due to the technical nature of this lemma, we provide its proof in Appendix A.

Lemma 1. *Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by T-NAM. Then, there exists $\sigma > 0$ such that $\sigma \leq \lambda_{\min}\left(\mathbf{A}^k\right) \leq \lambda_{\max}\left(\mathbf{A}^k\right) \leq 2(N-1)$ for any $k \geq 0$.*

An important property of T-NAM that directly follows from Lemma 1 and the update rule (16), is that there must exist $K \geq 0$ such that for any $k \geq K$ it holds that (recall that $\{j_k\}_{k\geq 0}$ in (16) is a non-decreasing sequence)

$$j_k \geq 2\sqrt{\frac{2(N-1)}{\sigma}} \geq 2\sqrt{\frac{\lambda_{\max}\left(\mathbf{A}^k\right)}{\lambda_{\min}\left(\mathbf{A}^k\right)}}. \tag{24}$$

Now, we are ready to state two central propositions, that will enable us to prove the global convergence result of Theorem 1. The proofs of Propositions 1 and 2 can be found in Appendix B.

Proposition 1 *(Function value decrease up to error). Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a sequence generated by T-NAM. Then, the sequence $\left\{\Psi\left(\mathbf{s}^k, \mathbf{u}^k\right)\right\}_{k\geq 0}$ is non-increasing. If, in addition, the sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ is bounded, then for all $k \geq 0$*

$$\sigma\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\|^2 - e_1^k \leq \Psi\left(\mathbf{s}^k, \mathbf{u}^k\right) - \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}\right), \tag{25}$$

where $e_1^k \geq 0$ is a non-negative error term satisfying

$$e_1^k \equiv \frac{M_1}{j_k + 1}, \tag{26}$$

for some constant $M_1 > 0$.

Proposition 2 *(Sub-gradient bound by iterate gap up to error). Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by T-NAM. Then, there exist a vector $\mathbf{w}^{k+1} \in \partial\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^k\right)$ and a positive scalar $\mathcal{L} > 0$ such that for all $k \geq 0$*

$$\left\|\mathbf{w}^{k+1}\right\| \leq \mathcal{L}\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\| + e_2^k,$$

where $e_2^k \geq 0$ is a non-negative error term satisfying

$$e_2^k \equiv \frac{M_2}{j_k - 1}, \tag{27}$$

for some constant $M_2 > 0$.

Last, before proving Theorem 1, we will need a technical result regarding limiting sub-differential sets, which is presented now and is proved in Appendix C.

Lemma 2. *Let $\bar{\mathbf{x}} \in \mathbb{R}^n$ and let $U \subseteq \mathbb{R}^n$ be an open neighborhood of $\bar{\mathbf{x}}$. Let $g: \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on U. Define $h(\mathbf{x}) \equiv g(\mathbf{x}) \cdot \|\mathbf{x} - \bar{\mathbf{x}}\|$. Then,*

$$\partial h(\bar{\mathbf{x}}) = \begin{cases} g(\bar{\mathbf{x}}) \cdot \mathcal{B}, & g(\bar{\mathbf{x}}) \geq 0, \\ g(\bar{\mathbf{x}}) \cdot \mathcal{S}, & g(\bar{\mathbf{x}}) < 0. \end{cases}$$

Now, we are ready to prove Theorem 1, which is the main theoretical result of our paper

Proof of Theorem 1. Following [45, Theorem 1], based on Propositions 1 and 2, we need to show that the error sequences $\left\{\sqrt{e_1^k}\right\}_{k\geq 0}$ and $\left\{e_2^k\right\}_{k\geq 0}$ are summable. This will guarantee that (i) $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ converges to some $\mathbf{s}^* \in \mathbb{R}^n$, and (ii) if $\mathbf{u}^* \in \mathcal{B}^N$ is some limit point of $\left\{\mathbf{u}^k\right\}_{k\geq 0}$, then $(\mathbf{s}^*, \mathbf{u}^*)$ is a critical point of Ψ. To this end, notice that

$$\sum_{k=1}^{\infty} \frac{1}{\sqrt{j_k+1}} = \sum_{k=1}^{\infty} \frac{1}{\sqrt{s+2^{\lfloor k/r \rfloor}}} < \infty.$$

Hence, by the definition of e_1^k in (26) we establish that the sequence $\left\{\sqrt{e_1^k}\right\}_{k\geq 0}$ is summable. Similarly, $\left\{e_2^k\right\}_{k\geq 0}$ is also summable by the definition of e_2^k in (27) (recall that it is safe to assume that $j_k - 1 > 0$ as follows from (24)).

Now, we are left to show that any point of convergence $\mathbf{s}^* \in \mathbb{R}^n$ obtained by T-NAM is a critical point of Problem (LS). Let $\mathbf{u}^* \in \mathcal{B}^N$ be a limit point of the bounded sequence $\left\{\mathbf{u}^k\right\}_{k\geq 0}$ (following (10) and (11), this sequence is bounded since its elements are in $\mathcal{S}$). By possibly passing to a sub-sequence, we assume that $\mathbf{u}^k \to \mathbf{u}^*$ as $k \to \infty$.

To prove that $\mathbf{s}^*$ is a critical point of F, notice that since $(\mathbf{s}^*, \mathbf{u}^*)$ is a critical point of Ψ, then $\mathbf{0}_n \in \partial_{\mathbf{s}} \Psi(\mathbf{s}^*, \mathbf{u}^*)$ ($\partial_{\mathbf{s}} \Psi$ is the limiting sub-differential of Ψ with respect to $\mathbf{s}$). Hence, it is enough to prove that $\partial_{\mathbf{s}} \Psi(\mathbf{s}^*, \mathbf{u}^*) \subseteq \partial F(\mathbf{s}^*)$.

The proof is split into three cases: $\mathbf{s}^* = \mathbf{p}_j$ for some $j = 2, 3, \ldots, N$, $\mathbf{s}^* = \mathbf{p}_1$, and $\mathbf{s}^* \neq \mathbf{p}_i$ for all $i = 1, 2, \ldots, N$. We will prove the first case in detail and skip the other two cases, which can be proved similarly.

Assume that $\mathbf{s}^* = \mathbf{p}_j$ for some $j = 2, 3, \ldots, N$. Since $\mathbf{s}^k \to \mathbf{s}^*$ as $k \to 0$, there exists some $K \geq 0$ such that $\mathbf{s}^k \neq \mathbf{p}_i$ for all $k \geq K$ and for all $i \neq j$. Hence, by the definition of $\mathbf{u}^k$ (see step 3 in Algorithm 1) we have

$$\mathbf{u}_1^{k+1} = \frac{\mathbf{s}^k - \mathbf{p}_1}{\left\|\mathbf{s}^k - \mathbf{p}_1\right\|} \quad \text{and} \quad \mathbf{u}_i^{k+1} = -\frac{\mathbf{s}^k - \mathbf{p}_i}{\left\|\mathbf{s}^k - \mathbf{p}_i\right\|},$$

for all $k \geq K$ and for all $i \neq \{1, j\}$. Taking the limit $k \to \infty$ in the above $N-1$ equalities we get

$$\mathbf{u}_1^* = \frac{\mathbf{p}_j - \mathbf{p}_1}{\left\|\mathbf{p}_j - \mathbf{p}_1\right\|} \quad \text{and} \quad \mathbf{u}_i^* = -\frac{\mathbf{p}_j - \mathbf{p}_i}{\left\|\mathbf{p}_j - \mathbf{p}_i\right\|}, \tag{28}$$

for all $i \neq \{1, j\}$. Now, we write the function F as a sum of a function that is differentiable at $\mathbf{s}^* = \mathbf{p}_j$ and a function that is not differentiable at $\mathbf{s}^* = \mathbf{p}_j$. To this end, we denote by $F_j \colon \mathbb{R}^n \to \mathbb{R}$ the function

$$F_j(\mathbf{s}) = \frac{1}{2}\sum_{i=2}^{N}\left(\|\mathbf{s}-\mathbf{p}_i\|^2 + \|\mathbf{s}-\mathbf{p}_1\|^2 + 2d_i\|\mathbf{s}-\mathbf{p}_1\|\right) - \sum_{i=2, i\neq j}^{N}\left(\|\mathbf{s}-\mathbf{p}_i\| \cdot \|\mathbf{s}-\mathbf{p}_1\| + d_i\|\mathbf{s}-\mathbf{p}_i\|\right),$$

and by $h_j \colon \mathbb{R}^n \to \mathbb{R}$ the function $h_j(\mathbf{s}) = g_j(\mathbf{s}) \cdot \left\|\mathbf{s} - \mathbf{p}_j\right\|$ where $g_j(\mathbf{s}) = -\|\mathbf{s} - \mathbf{p}_1\| - d_j$. Notice that F_j is differentiable at $\mathbf{p}_j$ while h_j is not. Simple calculations show

$$F(\mathbf{s}) = F_j(\mathbf{s}) + h_j(\mathbf{s}) + \frac{1}{2}\sum_{i=2}^{N} d_i^2.$$

Plugging $g = g_j$ and $h = h_j$ in Lemma 2, and since $g_j(\mathbf{p}_j) < 0$, then $\partial h_j(\mathbf{p}_j) = \left(\left\|\mathbf{p}_j - \mathbf{p}_1\right\| + d_j\right) \cdot \mathcal{S}$. Hence

$$\partial F(\mathbf{p}_j) = \nabla F_j(\mathbf{p}_j) + \left(\left\|\mathbf{p}_j - \mathbf{p}_1\right\| + d_j\right) \cdot \mathcal{S}. \tag{29}$$

In addition, using (28), simple calculations show that

$$\nabla F_j(\mathbf{p}_j) = 2(N-1)\mathbf{p}_j - \sum_{i=1}^{N}(\mathbf{p}_1 + \mathbf{p}_i) + d\mathbf{u}_1^* + \sum_{i=2, i\neq j}^{N}\left(\left\|\mathbf{p}_j - \mathbf{p}_1\right\|\mathbf{u}_i^* - \left\|\mathbf{p}_j - \mathbf{p}_i\right\|\mathbf{u}_1^* + d_i\mathbf{u}_i^*\right).$$

Now, recall that we assume that $j \neq 1$. Hence, the function $\mathbf{s} \mapsto \Psi(\mathbf{s}, \mathbf{u}^k)$ is differentiable in some open neighborhood of $\mathbf{p}_j$, and therefore for any $k \geq K$ we have

$$\nabla_{\mathbf{s}}\Psi\left(\mathbf{s}^k, \mathbf{u}^k\right) = 2\mathbf{A}^k\mathbf{s}^k - \mathbf{z}^k + d\frac{\mathbf{s}^k - \mathbf{p}_1}{\left\|\mathbf{s}^k - \mathbf{p}_1\right\|}. \tag{30}$$

Taking the limit $k \to \infty$ in (30), and recalling the definition of $\mathbf{A}^k$ and $\mathbf{z}^k$ (see (19) and (20)), then again by using (28) one can show after simple calculations that

$$\nabla_{\mathbf{s}}\Psi(\mathbf{p}_j, \mathbf{u}^*) = \nabla F_j(\mathbf{p}_j) + \left(\left\|\mathbf{p}_j - \mathbf{p}_1\right\| + d_j\right)\mathbf{u}_j^*. \tag{31}$$

Since $\mathbf{u}_j^k \in \mathcal{S}$ for all $k \geq 0$ by its definition, and since $\mathcal{S}$ is a closed set, then $\mathbf{u}_j^* \in \mathcal{S}$. Hence, by combining (29) and (31) we get $\nabla_{\mathbf{s}}\Psi(\mathbf{p}_j, \mathbf{u}^*) \in \partial F(\mathbf{p}_j)$, as required. □

5. Numerical analysis

In order to evaluate the performance of our suggested T-NAM and FP algorithms, in this section we numerically compare them with the following algorithms:

1. SOLVIT of [12]: a recent first-order iterative algorithm that directly tackles Problem (LS) by applying the MM approach. The exact SOLVIT updates used in this paper for the case of a single reference sensor are given in Appendix D.
2. WLS of [3]: this algorithm estimates a globally optimal solution of the set of smooth equations in (3), hence this method does not solve the original Problem (LS). This is not an iterative algorithm, and it finds a location estimation by solving a weighted LS problem. Following [3], in all experiments below, the first-step of WLS consists of three repetitions. The code of WLS was taken from [48].
3. SCWLS of [32]: extends the WLS method of [3] by solving a constrained WLS problem. This is not an iterative method and it tackles the set of smooth equations in (3), and not the original Problem (LS). The code of SCWLS was taken from [49].
4. SDP of [17]: a standard convex relaxation of Problem (LS) that is solved using a commercial off-the-shelf SDP solver. The problem formulated in [17] tackles the TDoA problem under a more general setting, hence the exact SDP formulation suitable for the setting of this paper is given in Appendix D.

5.1. Computational complexity analysis

The computation costs of the methods are summarized above in Table 1 (see Section 1.3) and are discussed now. Both T-NAM and FP require the calculation of N norm terms of size n. In addition, T-NAM also performs a matrix-vector multiplication of size n. Thus, FP and T-NAM end up with a low total per iteration computation cost of $\mathcal{O}(nN)$ and $\mathcal{O}\left(n^2 + nN\right)$, respectively. The SOLVIT algorithm requires calculating the inverse of a square matrix of

Table 2
Comparison of the random array geometries.

Array	$\mathbf{s}^{\text{true}}$	Sensors	T_{iter}
Near Field	$[-0.5, 0.5]^2$	$[-0.5, 0.5]^2$	2000
Far Field	$[0.5, 1.5]^2$	$[-0.5, 0.5]^2$	5000
Circular	$[-0.5, 0.5]^2$	$\mathcal{S}$	2000

size $n \times n$, with additional summation of N matrix-vector multiplications of size n, hence its per iteration complexity is $\mathcal{O}\left(n^3 + n^2 N\right)$ (see Appendix D). Both WLS and SCWLS solve a least squares problem of size n, and both compute a matrix-matrix multiplication of size $N \times N$, hence their complexity is $\mathcal{O}\left(n^3 + N^3\right)$. As for the SDP, it runs a primal-dual interior-point method using the commercial solver SeDuMi. Following [50], SDP ends up with a high total computation cost of $\mathcal{O}\left(n^2 N^5 + N^9\right)$.

5.2. Generating the data and description of the metrics

We denote by $\mathbf{s}^{\text{true}} \in \mathbb{R}^n$ the true location of the source. In the experiments to follow, we consider three types of planar array geometries.

1. *Near field box array:* the location of each sensor and $\mathbf{s}^{\text{true}} \in \mathbb{R}^n$ are randomly drawn from a uniform distribution over the box $[-0.5, 0.5]^2$.
2. *Far field box array:* as above, but $\mathbf{s}^{\text{true}}$ is drawn from a uniform distribution over the box $[0.5, 1.5]^2$. Meaning, the true location of the source is necessarily outside the convex hull of the sensors.
3. *Circular array:* the sensors are located on the unit sphere. These positions are generated by drawing a random box array (see above), and projecting it onto the unit sphere. In addition, $\mathbf{s}^{\text{true}}$ is drawn from a uniform distribution over $[-0.5, 0.5]^2$, hence can be inside or outside the convex hull of the sensors.

All three iterative methods (which are T-NAM, FP and SOLVIT), ran for exactly $T_{\text{iter}} = 2000$ total iterations for the near field and circular array geometries, and $T_{\text{iter}} = 5000$ total iterations for the far field geometry. By total iterations we mean the total number of inner and outer iterations. For example, if in the first and second outer iterations of T-NAM it ran for two inner iterations, then the total iteration counter now reads $T_{\text{iter}} = 4$.

The properties of each random array geometry, as well as the number of total iterations used in the experiments, are summarized in Table 2.

The number of inner iterations $j_k \in \mathbb{N}$ in the k-th outer iteration of T-NAM, for any $k \geq 0$, is set according to the rule (16). In the experiments below, we set $s = 1, 2, 3, 4, 5$ and $r = 1000$. Hence, the number of inner iterations is explicitly given by $j_k = s + 2^{\lfloor 0.001k \rfloor} - 1$. As for the integer parameter r of T-NAM, its chosen value does not make very much of a numerical difference, as it only dictates the number of outer iterations after which there will be an increase in the number of inner iterations. As for the integer parameter s of T-NAM, we discuss below that taking larger values may be satisfactory.

For each of the array geometries, we considered smaller-scale arrays with $N = 4$ sensors and larger-scale arrays with $N = 15$ sensors. Due to increased computational cost of SDP, it was tested only for the smaller-scale arrays.

The data generation process is summarized in a flow chart in Fig. 1, and is described now. For each array geometry, we generated $K = 200$ random arrays. The noises are assumed to be Gaussian of the form

$$\tilde{\epsilon}_i \sim \text{Normal}\left(0, \sigma_{\text{dist}}^2\right), \quad \forall i = 1, 2, \ldots, N,$$

where, following (2), we defined $\epsilon_i = \tilde{\epsilon}_i - \tilde{\epsilon}_1$, $i = 2, 3, \ldots, N$. We consider 10 different values of σ_{dist}, logarithmically scaled in the interval $\left[10^{-4}, 10^{-1}\right]$. In addition, for each σ_{dist}, we drew $R = 100$ realizations of the noises. Meaning, for each random array, and for each σ_{dist} we drew 100 different values of each ϵ_i. In total, this sums to $200 \times 10 \times 100 = 200000$ independent Monte-Carlo trials.

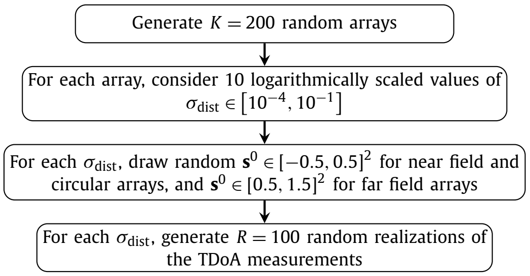
Generate $K = 200$ random arrays
For each array, consider 10 logarithmically scaled values of $\sigma_{\text{dist}} \in \left[10^{-4}, 10^{-1}\right]$
For each σ_{dist}, draw random $\mathbf{s}^0 \in [-0.5, 0.5]^2$ for near field and circular arrays, and $\mathbf{s}^0 \in [0.5, 1.5]^2$ for far field arrays
For each σ_{dist}, generate $R = 100$ random realizations of the TDoA measurements

Fig. 1. Data generation flow chart, repeated for each of the near field, far field and circular array geometries (for each geometry, the process was repeated for $N = 4$ and $N = 15$ sensors).

As for the starting point, for each of the random arrays and for each σ_{dist}, we drew a random starting point $\mathbf{s}^0 \in [-0.5, 0.5]^2$, that was used across all $R = 100$ realizations. All three iterative methods (T-NAM, FP and SOLVIT) were initialized with the same starting point (the other methods are independent of such point).

In the results below, the methods are compared using the following metrics:

(i) Decrease in the LS function value as a function of the running time for the three iterative methods T-NAM, FP and SOLVIT (Section 5.3).
(ii) Output LS function value for the non-iterative methods WLS, SCWLS and SDP (Section 5.4).
(iii) Run time (Section 5.5).
(iv) Estimated Root Mean Square Error (RMSE) compared to the CRLB (Section 5.6).

In all descriptions, graphs and tables below, the displayed values are the average over all Monte-Carlo trials (meaning, over all $K = 200$ random arrays and $R = 100$ realizations for each of the array geometries).

All experiments were executed on an Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz with a total of 300 GB RAM and 72 threads, using MATLAB 2019a.

5.3. Decrease in LS function value vs. running time

We consider the objective F of Problem (LS). For each of the iterative methods T-NAM, FP and SOLVIT, we calculated the function value as a function of the running time. A lower function value is an indication on how good a method is in minimizing F. We considered a running time interval of $[0, a]$ seconds ($a > 0$), and plotted the decrease in values along this interval.

For a fair comparison, each compared method was implemented without any loops and using only “one-liner” code updates. Once a method reached the allowed total iteration bound T_{iter} (see Table 2), the method was stopped. Since WLS, SCWLS and SDP are not iterative (and only yield an output function value, which is the value at the output point), they are treated separately in Section 5.4).

In Fig. 2, we plot the decrease in function value (the y-axis) along the running time (the x-axis) for low noise level $\sigma_{\text{dist}} = 10^{-4}$. Each plot represents the function value reached by a method after a certain amount of running time. I.e., for any point along the x-axis, a method with a lower function value means that it required the same execution time to reach this lower value. Once a

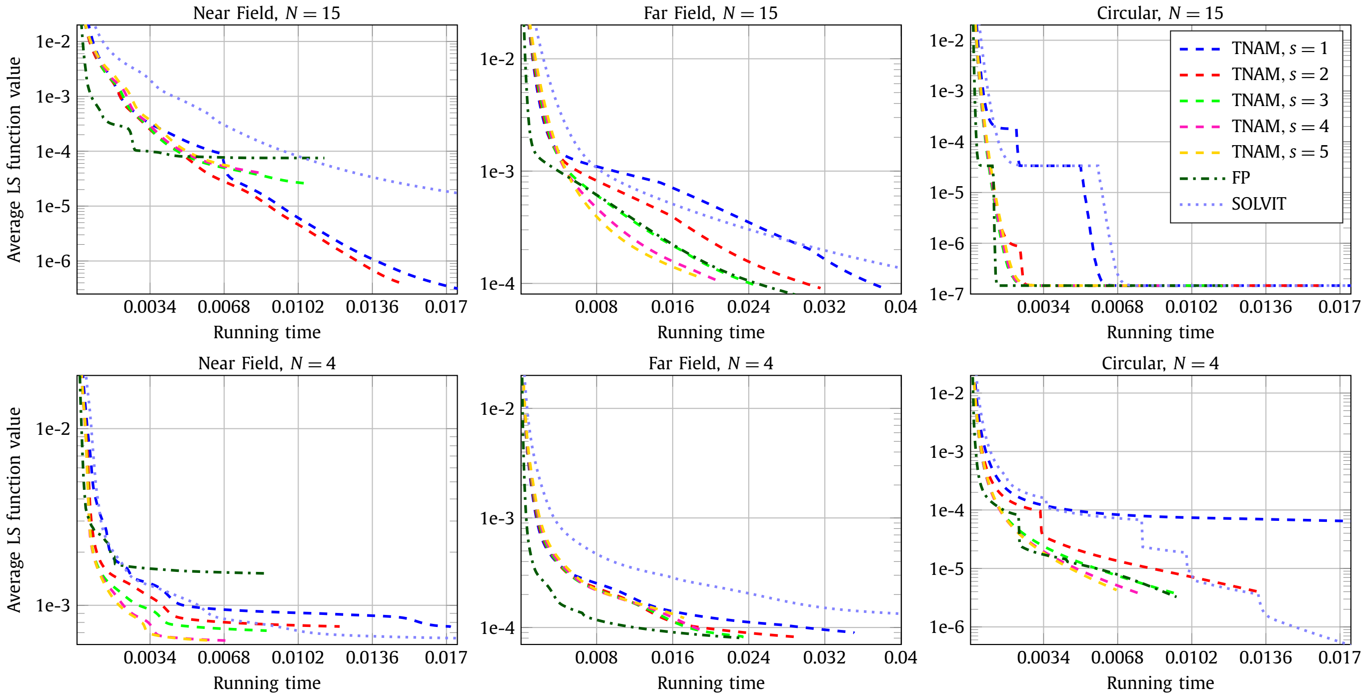

Fig. 2. Average LS function value as a function of the running time (in seconds) for $\sigma_{\text{dist}} = 10^{-4}$. Average function value (y-axis in a logarithmic scale) of the iterative methods for the three types of array geometries, as a function of the running time (x-axis in seconds), for larger-scale (top) and smaller-scale arrays (bottom). All methods start at the same value, however all graphs are truncated from above and from the right for the purpose of a clear visualization.

method reached the maximal allowed number of iterations, its line plot ends. In Table 3, we summarize the results for low, medium and high $\sigma_{\text{dist}} \in \left\{10^{-4}, 10^{-2}, 10^{-1}\right\}$. The results in Fig. 2 are discussed now.

5.3.1. Larger-scale arrays with $N = 15$ sensors

In the near field arrays, T-NAM is the fastest as it generally yielded a lower value for a given execution time. T-NAM with $s = 1, 2$ reached the lowest value given the maximal allowed number of iterations $T_{\text{iter}} = 2000$. However, T-NAM with $s = 3, 4, 5$ completed $T_{\text{iter}} = 2000$ iterations in a shorter time. Meaning, given a certain time limit, all variants of T-NAM in this array setting are roughly similar. FP was the fastest to reach lower values at the beginning of the run. However, it quickly converges (as its line plot becomes flat) to a point with a higher value compared to other methods. As for SOLVIT, though it did converge to a good value, its execution time was the slowest and the method needed to run for a longer period of time to reach a value already reached by the others.

In the far field arrays, all methods did not converge after $T_{\text{iter}} = 5000$ iterations (as none of the line plots become flat), indicating that this geometry induces a more challenging optimization problem, at least for a low noise level $\sigma_{\text{dist}} = 10^{-4}$. T-NAM with $s = 5$ was superior in this setting, followed by T-NAM with $s = 4$, $s = 3$, FP and T-NAM with $s = 2$. SOLVIT and T-NAM with $s = 1$ performed similarly slow, indicating that additional inner FISTA iterations are required in this challenging setting.

In the circular arrays, all methods converged fast to the same value, indicating that the induced problem is easier. FP and T-NAM with $s = 2, 3, 4, 5$ performed similarly fast, while SOLVIT and T-NAM with $s = 1$ were slower.

Considering all array geometries for the larger-scale networks, we conclude that in general SOLVIT was the slowest to reach the same value. This result matches the fact that the computational burden of SOLVIT is heavier than T-NAM and FP (see Table 1).

As for T-NAM, we point out that for a predefined number of total iterations T_{iter}, increasing the value of $s \in \mathbb{N}$ (meaning, applying more FISTA iterations) implies less $\mathbf{u}^k$ updates (see steps 3 and 4 in Algorithm 1), which are slower updates relatively to FISTA updates. However, more $\mathbf{u}^k$ updates might result with a lower overall function value, at least in some of the tested geometries. Therefore, based on the above results, if no prior knowledge of the geometry of the array or noise level is known, our recommendation is to initialize T-NAM with large values of $s \in \mathbb{N}$, as it would result with a fast convergence to a low function value. The advantage of such initialization is clear, as now T-NAM becomes parameter-free.

We refer the reader to Table 3 for a summary of the results obtained for different values of σ_{dist}. In general, FP and T-NAM with $s = 5$ yield better results, with an advantage to T-NAM in noisier settings. Notice that SOLVIT did not obtain a value in the noisiest setting of the near field geometry, since the involved matrix in at least one of the Monte-Carlo trials was singular (see Appendix D).

5.3.2. Smaller-scale arrays with $N = 4$ sensors

In the near field setting, T-NAM with $s = 4, 5$ outperform the others, and both yielded lower values faster. Notice that all methods yielded higher values compared to the larger-scale near field arrays, indicating that it is more challenging to reach lower values in sparse settings that posses little information. In the far field setting, FP was the fastest in reaching lower values, and roughly all variants of T-NAM performed similarly. However, examining the line trajectory of T-NAM with $s = 4, 5$, we see that applying more total iterations would result in a low function value (as FP) with a similar execution time. In the circular setting, there was a slight advantage to T-NAM with $s = 4, 5$ over FP, and T-NAM with $s = 1$ was the slowest. As in the larger-scale setting, SOLVIT was slower than other methods (though it did yield a low value in the circular setting).

Considering the results in Fig. 2 and Table 3 for the smaller-scale array geometries, we conclude that there is an advantage in applying T-NAM with a larger number of inner FISTA iterations,

Table 3
Decrease in average LS function value after 0.003 (left) and 0.006 (right) seconds from initialization, for low, medium and high noises. The lowest value in each column is in bold.

Method	$\sigma_{\text{dist}} = 10^{-4}$		$\sigma_{\text{dist}} = 10^{-2}$		$\sigma_{\text{dist}} = 10^{-1}$	
Near Field, $N = 15$						
T-NAM $s = 1$	0.0004	0.0001	0.0025	0.0019	**0.1864**	**0.1860**
T-NAM $s = 2$	0.0003	**3.9e-5**	0.0022	0.0019	0.1866	0.1863
T-NAM $s = 3$	0.0004	5.8e-5	0.0023	0.0019	0.1866	0.1863
T-NAM $s = 4$	0.0004	6.2e-5	0.0023	0.0019	0.1866	0.1864
T-NAM $s = 5$	0.0005	6.7e-5	0.0023	0.0019	0.1867	0.1864
FP	**0.0001**	7.8e-5	**0.0019**	**0.0018**	0.1914	0.1913
SOLVIT	0.0026	0.0005	0.0039	0.00231	0.1864	–
Far Field, $N = 15$						
T-NAM $s = 1$	0.0020	0.0012	0.0045	0.0035	0.2479	0.2301
T-NAM $s = 2$	0.0020	0.0010	0.0044	0.0031	0.2339	0.2191
T-NAM $s = 3$	0.0021	0.0009	0.0044	0.0029	0.2257	0.2152
T-NAM $s = 4$	0.0024	0.0007	0.0045	0.0027	0.2216	0.2139
T-NAM $s = 5$	0.0027	**0.0006**	0.0045	**0.0027**	**0.2196**	**0.2133**
FP	**0.0011**	0.0008	**0.0033**	0.0028	0.2277	0.2173
SOLVIT	0.0064	0.0016	0.0084	0.0040	0.2494	0.2296
Circular Array, $N = 15$						
T-NAM $s = 1$	3.3e-5	1.8e-7	**0.0015**	**0.0015**	**0.1742**	**0.1742**
T-NAM $s = 2$	1.4e-7	**1.4e-7**				
T-NAM $s = 3$	1.4e-7					
T-NAM $s = 4$	1.4e-7					
T-NAM $s = 5$	1.4e-7					
FP	**1.4e-7**					
SOLVIT	3.3e-5	1.3e-5				
Near Field, $N = 4$						
T-NAM $s = 1$	0.0014	0.0010	0.0017	0.0014	0.0220	0.0213
T-NAM $s = 2$	0.0012	0.0008	0.0015	0.0014	0.0213	0.0209
T-NAM $s = 3$	0.0010	0.0007	0.0014	0.0013	0.0211	0.0207
T-NAM $s = 4$	0.0008	0.0006	0.0014	0.0013	0.0210	0.0206
T-NAM $s = 5$	**0.0008**	**0.0006**	**0.0014**	**0.0013**	**0.0209**	**0.0206**
FP	0.0016	0.0015	0.0015	0.0014	0.0218	0.0216
SOLVIT	0.0013	0.0009	0.0017	0.0014	–	–
Far Field, $N = 4$						
T-NAM $s = 1$	0.0005	0.0003	0.0006	0.0005	0.0328	0.0300
T-NAM $s = 2$	0.0005	0.0003	0.0006	0.0004	0.0311	0.0284
T-NAM $s = 3$	0.0005	0.0003	0.0007	0.0004	0.0300	0.0275
T-NAM $s = 4$	0.0005	0.0003	0.0007	0.0004	0.0294	0.0270
T-NAM $s = 5$	0.0006	0.0003	0.0007	0.0004	0.0290	**0.0267**
FP	**0.0002**	**0.0001**	**0.0004**	**0.0004**	**0.0289**	0.0270
SOLVIT	0.0012	0.0006	0.0011	0.0005	0.0323	0.0287
Circular Array, $N = 4$						
T-NAM $s = 1$	0.0001	8.7e-5	0.0002	0.0001	0.0139	0.0135
T-NAM $s = 2$	9.8e-5	1.5e-5	0.0002	0.0001	0.0136	0.0134
T-NAM $s = 3$	2.8e-5	9.3e-6	0.0001	0.0001	0.0135	0.0133
T-NAM $s = 4$	2.3e-5	6.5e-6	**0.0001**	0.0001	0.0135	0.0133
T-NAM $s = 5$	2.2e-5	**5.5e-6**	0.0001	0.0001	0.0134	**0.0133**
FP	**1.8e-5**	9.4e-6	0.0001	**0.0001**	**0.0134**	0.0133
SOLVIT	0.0002	7.6e-5	0.0002	0.0001	0.0137	0.0134

i.e., with $s = 5$, compared to other methods, while FP is a good competitor in the far field setting. As in the larger-scale network setting, SOLVIT did not provide a value in the noisy near field geometry due to matrix singularity.

5.4. Output LS function value

In the previous sub-section, we compared the function value of the iterative methods. Recall that the non-iterative methods (which are WLS, SCWLS and SDP) only generate a single function value, which is the value at the output point. Therefore, we now compare the output value obtained by WLS, SCWLS and SDP with the output value of our suggested T-NAM and FP methods (which is the value at the last iteration T_{iter}, where T_{iter} is set according to Table 2). Here, for the sake of a simple representation of the results, we focus on T-NAM with $s \in \{2, 5\}$. Recall that WLS, CSWLS and SDP minimize a different objective function than T-NAM and FP (see Table 1), however since Problem (LS) is the original problem, we compare the methods in minimizing the objective of Problem (LS).

Table 4
Average output LS function value for low, medium and high noise levels, for $N = 15$ (left) and $N = 4$ (right). The lowest value in each column is in bold.

Method	$\sigma_{\text{dist}} = 10^{-4}$		$\sigma_{\text{dist}} = 10^{-2}$		$\sigma_{\text{dist}} = 10^{-1}$	
Near Field						
T-NAM $s = 2$	2.7e-7	0.0008	0.0018	0.0013	**0.1862**	0.0207
T-NAM $s = 5$	4.6e-5	0.0006	0.0019	0.0013	0.1863	**0.0206**
FP	7.5e-5	0.0015	0.0018	0.0014	0.1912	0.0215
WLS	1.5e-7	6.8e-6	0.0024	0.1005	1.6467	0.6204
SCWLS	**1.5e-7**	**1.3e-8**	**0.0015**	**0.0002**	0.3149	0.0364
SDP	–	0.0050	–	0.0063	–	0.0444
Far Field						
T-NAM $s = 2$	7.8e-5	8.0e-5	0.0020	0.0003	**0.2083**	**0.0248**
T-NAM $s = 5$	0.0002	0.0001	0.0020	0.0003	0.2091	0.0249
FP	7.2e-5	7.9e-5	0.0012	**0.0003**	0.2110	0.0250
WLS	1.8e-7	0.0104	0.0216	0.5006	5.8557	1.0929
SCWLS	**1.8e-7**	**2.0e-7**	**0.0028**	0.0017	0.3334	0.0598
SDP	–	0.0115	–	0.0120	–	0.0437
Circular Array						
T-NAM $s = 2$	**1.4e-7**	3.1e-6	**0.0015**	**0.0001**	**0.1742**	0.0133
T-NAM $s = 5$		3.4e-6		0.0001		0.0133
FP		2.6e-6		0.0001		**0.0133**
WLS	1.6e-7	0.0407	0.2071	0.6914	3.3015	1.1255
SCWLS	**1.4e-7**	**1.1e-8**	0.0015	0.0001	0.1809	0.0170
SDP	–	0.0277	–	0.0286	–	0.1179

In Fig. 3, we plot the output value obtained by the methods as a function of the ten noise levels σ_{dist}, for all array geometries. In Table 4 we give a summary of the results presented in Fig. 3 for low, medium and high noise levels. Due to a high execution time, SDP was only tested for the smaller-scale settings with $N = 4$ sensors. The results in Fig. 3 are discussed now.

5.4.1. Larger-scale arrays with $N = 15$ sensors

For all geometries and methods, the function value increased as the noise level σ_{dist} increased.

For low values of σ_{dist} in the near and far field settings, WLS and SCWLS yielded low values compared to T-NAM and FP. This is an indication that $T_{\text{iter}} = 2000$ and $T_{\text{iter}} = 5000$ for the near and far field geometries, respectively, are not enough for T-NAM and FP to reach a lower function value. Meaning, T-NAM and FP should run for *more* total iterations. Recall that the computational burden of T-NAM and FP is much less demanding than WLS and SCWLS (see Table 2), hence one can safely increase the total iteration count.

In the circular setting, all methods obtained the same value, except for WLS which yielded significantly higher values. This result is consistent with the results in [32], which claim that WLS fails to solve the TDoA problem in circular settings and in environments with high noise.

5.4.2. Smaller-scale arrays with $N = 4$ sensors

In the near field array, the output values are higher than compared to the values in the larger-scale setting, indicating that for a small number of measurements, the induced problem is more challenging. Nonetheless, T-NAM and FP generally performed better than others.

In the far field and circular arrays, we again see that the predefined T_{iter} (see Table 2) is not high enough for T-NAM and FP to reach a lower value. Meaning, these two should run for *more* total iterations. In addition, we see that unlike the near field setting, WLS performs poorly in the far field and circular settings.

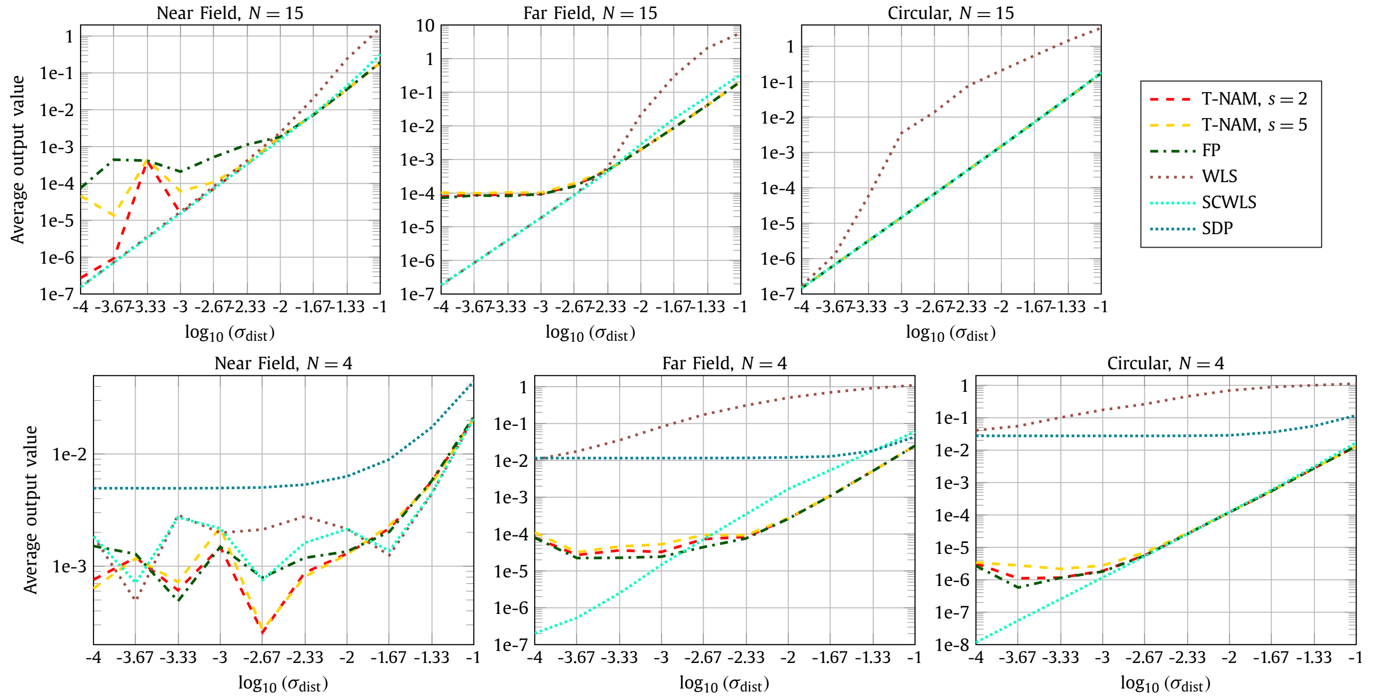

Fig. 3. Average output LS function value for all 10 noise levels σ_{dist}. Average output function value (y-axis in a logarithmic scale) of T-NAM, FP, WLS, SCWLS and SDP, as a function of the noise levels (x-axis in a $\log_{10}$ scale), for larger-scale (top) and smaller-scale arrays (bottom).

Table 5
Average run time per iteration and average total run time in seconds, for $N = 15$ (left) and $N = 4$ (right), for $\sigma_{\text{dist}} = 10^{-4}$. The fastest value in each column is in bold.

Method	Near Field		Far Field		Circular Array	
Run Time Per Iteration						
T-NAM $s = 1$	1.2e-5	1.0e-5	1.1e-5	1.0e-5	1.2e-5	1.0e-5
T-NAM $s = 2$	8.4e-6	7.6e-6	8.3e-6	7.4e-6	8.3e-6	7.2e-6
T-NAM $s = 3$	6.0e-6	5.5e-6	5.9e-6	5.3e-6	5.9e-6	5.2e-6
T-NAM $s = 4$	4.8e-6	4.4e-6	4.7e-6	4.2e-6	4.7e-6	4.2e-6
T-NAM $s = 5$	**4.0e-6**	**3.7e-6**	**4.0e-6**	**3.6e-6**	**4.0e-6**	**3.6e-6**
FP	6.4e-6	5.3e-6	6.4e-6	5.2e-6	6.5e-6	5.2e-6
SOLVIT	3.1e-5	2.8e-5	3.0e-5	2.7e-5	3.1e-5	2.7e-5
Total Run Time						
WLS	**0.0002**	**0.0001**	**0.0002**	**0.0001**	**0.0002**	**0.0001**
SCWLS	0.0017	0.0016	0.0018	0.0016	0.0017	0.0015
SDP	–	0.4637	–	0.4603	–	0.4562

As for SDP, it yields a very poor value, indicating that, if possible, solving convex relaxations of Problem (LS) should be refrained.

5.5. Run time per iteration and total run time

In Table 5, we give the average run time per iteration of the iterative methods (T-NAM, FP and SOLVIT), and the total run time of the others (WLS, SCWLS and SDP). Since the noise level σ_{dist} has no effect on the running times (only on the quality of the solution), in Table 5 we only give the values for $\sigma_{\text{dist}} = 10^{-4}$.

As seen in Table 5, T-NAM with $s = 5$ is the fastest among the iterative methods and across all array geometries. As discussed above, for a predefined number of total iterations, T-NAM with $s = 5$ performs less $\mathbf{u}^k$ updates, resulting with a lower average computation time per iteration. In addition, and as expected, the total run time of SDP is high even for smaller-scale arrays.

5.6. RMSE, CRLB and bias

Here, we compare the average estimated (or observed) RMSE, denoted by $\widehat{\text{RMSE}}$. Recall that for an unbiased estimator, the RMSE is bounded from below by the CRLB. Hence, if the RMSE value is below the CRLB, then the estimator is necessarily *biased* with respect to the true source location. Since we do not have access to the true bias (since the expectation functions of the estimators are unknown), this metric indicates if a method yields biased estimators. The formulas for $\widehat{\text{RMSE}}$ and the CRLB are given in Appendix E.

In Fig. 4, we plot the CRLB and $\widehat{\text{RMSE}}$ achieved by the methods (the y-axis), as a function of the noise level σ_{dist} (the x-axis). In Table 6, we summarize the results given in Fig. 4 for four selected values of σ_{dist}. Here, we only consider the larger-scale arrays with $N = 15$ sensors, as the smaller-scale arrays posses little information and hence the results were heavily biased for all compared methods (especially for WLS, SCWLS and SDP). The results in Fig. 4 and Table 6 are discussed now.

In the near field setting, all methods give $\widehat{\text{RMSE}}$ values above the CRLB, indicating that the estimates are not necessarily biased. The values of FP, SOLVIT, WLS, SCWLS and T-NAM with $s = 1, 2$ are closer to the CRLB, indicating that these methods might yield estimates with smaller variance (see Appendix E). For T-NAM with $s = 3, 4, 5$, the results suggest that their variance is somewhat higher, indicating that the allowed number of total iterations T_{iter} (as set in Table 2), can be increased for obtaining estimates with lower variance. In this context, recall that the computational cost and run time of T-NAM are smaller compared to other methods (see Table 1), hence one can safely increase the total iteration count of T-NAM. As for SOLVIT, it did not obtain an $\widehat{\text{RMSE}}$ value for a high noise level of $\sigma_{\text{dist}} = 10^{-1}$. The reason is that in at least one trial, the involved matrix (see Appendix D) was singular, resulting with no output and hence $\widehat{\text{RMSE}}$ was undefined. As mentioned in Section 1.2, SOLVIT is not always well-defined, which might result with a failure to yield an estimate (especially for high noise levels).

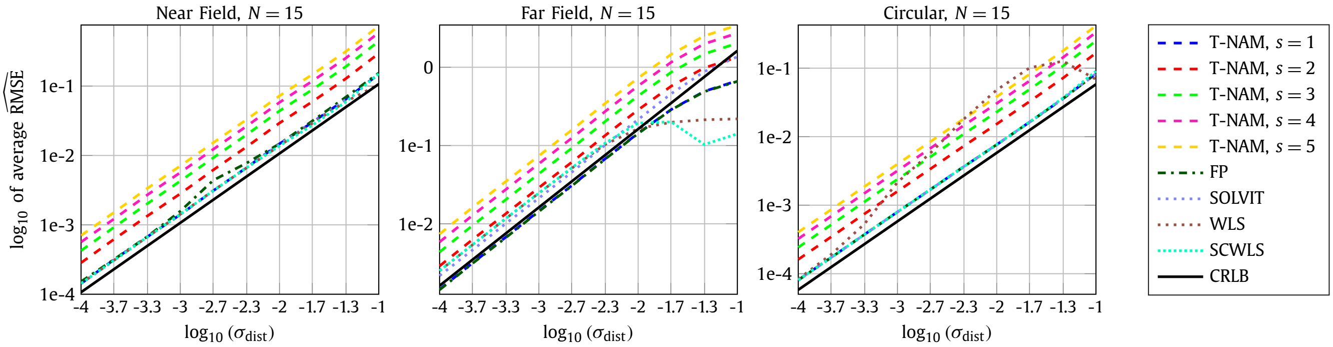

Fig. 4. Average $\widehat{\text{RMSE}}$ for all 10 noise levels σ_{dist}. Average $\widehat{\text{RMSE}}$ (y-axis in a $\log_{10}$ scale) of T-NAM, FP, SOLVIT, WLS and SCWLS for the three types of array geometries, as a function of the noise levels (x-axis in a $\log_{10}$ scale), for larger-scale networks with $N = 15$ sensors.

Table 6
Average $\widehat{\text{RMSE}}$ and average CRLB, for four selected noises, for $N = 15$. In red color are values below the CRLB.

Method	Near Field	Far Field	Circular
$\sigma_{\text{dist}} = 10^{-4}$			
T-NAM $s = 1$	1.4e-4	0.0015	8.0e-5
T-NAM $s = 2$	2.8e-4	0.0029	1.6e-4
T-NAM $s = 3$	4.2e-4	0.0044	2.4e-4
T-NAM $s = 4$	5.7e-4	0.0059	3.2e-4
T-NAM $s = 5$	7.0e-4	0.0074	4.0e-4
FP	1.5e-4	0.0014	8.0e-5
SOLVIT	1.4e-4	0.0022	8.0e-5
WLS	1.4e-4	0.0025	8.3e-5
SCWLS	1.4e-4	0.0025	8.0e-5
CRLB	1.0e-4	0.0016	5.8e-5
$\sigma_{\text{dist}} = 10^{-3}$			
T-NAM $s = 1$	1.4e-3	0.0146	0.0008
T-NAM $s = 2$	2.8e-3	0.0289	0.0016
T-NAM $s = 3$	4.2e-3	0.0436	0.0024
T-NAM $s = 4$	5.6e-3	0.0586	0.0032
T-NAM $s = 5$	7.0e-3	0.0739	0.0040
FP	1.5e-3	0.0144	0.0008
SOLVIT	1.4e-3	0.0213	0.0008
WLS	1.4e-3	0.0247	0.0022
SCWLS	1.4e-3	0.0247	0.0008
CRLB	1.0e-3	0.0162	0.0006
$\sigma_{\text{dist}} = 10^{-2}$			
T-NAM $s = 1$	0.0147	0.1437	0.0077
T-NAM $s = 2$	0.0294	0.2857	0.0153
T-NAM $s = 3$	0.0439	0.4304	0.0230
T-NAM $s = 4$	0.0582	0.5781	0.0306
T-NAM $s = 5$	0.0724	0.5781	0.0383
FP	0.0147	0.1423	0.0077
SOLVIT	0.0146	0.2146	0.0077
WLS	0.0138	0.1643	0.0482
SCWLS	0.0138	0.1937	0.0077
CRLB	0.0108	0.1622	0.0058
$\sigma_{\text{dist}} = 10^{-1}$			
T-NAM $s = 1$	0.1483	0.6728	0.0833
T-NAM $s = 2$	0.2951	1.3434	0.1665
T-NAM $s = 3$	0.4414	2.0276	0.2498
T-NAM $s = 4$	0.5871	2.7281	0.3331
T-NAM $s = 5$	0.7321	3.4469	0.4163
FP	0.1541	0.6606	0.0833
SOLVIT	–	1.3649	0.0833
WLS	0.1053	0.2190	0.0689
SCWLS	0.1506	0.1412	0.0906
CRLB	0.1069	1.6217	0.0581

In the far field setting, WLS and SCWLS are well below the CRLB for high noises, indicating that these methods are heavily biased with respect to the true source location. FP and T-NAM with $s = 1$ are slightly below the CRLB, indicating that these two methods are somewhat biased.

In the circular setting, all methods are with values above the CRLB (hence are not biased). WLS has a varying variance, as its $\widehat{\text{RMSE}}$ is not proportional to the CRLB.

6. Conclusion

In this paper, we introduced a novel algorithm, called T-NAM, that directly tackles the non-convex and non-smooth least squares formulation of the TDoA source localization problem. This algorithm uses the NAM scheme together with the fast FISTA method as a nested algorithm. Using a recent proof technique, we proved that T-NAM globally converges to critical points. To the best of our knowledge, this is the first algorithm to enjoy the global convergence result for this non-convex and non-smooth problem. We also introduced a simple fixed-point (FP) method that tackles the same problem formulation. We compared FP and T-NAM with other methods that solve the TDoA problem, and we numerically showed the advantages of the algorithms for low and high noise levels by inspecting the function values, running times, estimated RMSE and bias.

CRediT authorship contribution statement

Eyal Gur: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Alon Amar:** Conceptualization, Investigation, Methodology, Software, Validation, Writing – review & editing. **Shoham Sabach:** Conceptualization, Formal analysis, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Proof of Lemma 1

We first prove the following result that will later be useful in proving Lemma 1.

Lemma A. *Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by T-NAM. Then,*

(i) *For any $k \geq 0$ there exists $i = 2, 3, \ldots, N$ such that $\left(\mathbf{u}_i^k\right)^T \mathbf{u}_1^k > -1$.*
(ii) *If $\tilde{\mathbf{u}} \in \mathcal{B}^N$ is a limit point of $\left\{\mathbf{u}^k\right\}_{k\geq 0}$, then there exists $i = 2, 3, \ldots, N$ such that $\tilde{\mathbf{u}}_i^T \tilde{\mathbf{u}}_1 > -1$.*

Proof. First, for any $(\mathbf{s}, \mathbf{c}, \gamma) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$, we define the vector $\mathbf{v}(\mathbf{s}, \mathbf{c}, \gamma) \equiv \mathbf{s} + \gamma \left\|\mathbf{s} - \mathbf{p}_1\right\| \mathbf{c} \in \mathbb{R}^n$.

(i): Let $k \geq 0$. Assume on the contrary that for all $i = 2, 3, \ldots, N$ it holds that $\left(\mathbf{u}_i^k\right)^T \mathbf{u}_1^k = -1$. Hence $-1 = \left(\mathbf{u}_i^k\right)^T \mathbf{u}_1^k \geq -\left\|\mathbf{u}_i^k\right\| \cdot \left\|\mathbf{u}_1^k\right\| = -1$ (the second equality holds true since $\mathbf{u}_1^k, \mathbf{u}_i^k \in \mathcal{S}$). Thus, there exist scalars $\gamma_2^k, \gamma_3^k \ldots, \gamma_N^k \in \mathbb{R}$ (all depend on $k \geq 1$) such that

$$\mathbf{u}_1^k = \gamma_i^k \mathbf{u}_i^k, \quad \forall i = 2, 3, \ldots, N. \tag{A.1}$$

If $\gamma_i^k = 0$ for some $i = 2, 3, \ldots, N$, then $\mathbf{u}_1^k = \mathbf{0}_n$ and we arrive at a contradiction since $\mathbf{u}_1^k \in \mathcal{S}$. Hence, $\gamma_i^k \neq 0$ for all $i = 2, 3, \ldots, N$. Using step 4 of Algorithm 1 we arrive at two possible cases that follow from (A.1):

(a) If $\mathbf{s}^k = \mathbf{p}_j$ for some $j = 2, 3, \ldots, N$, then $\mathbf{s}^k \neq \mathbf{p}_i$ for any $i \neq j$, and $\mathbf{p}_1 = \mathbf{v}\left(\mathbf{s}^k, \mathbf{s}^k - \mathbf{p}_i, \gamma_i^k / \left\|\mathbf{s}^k - \mathbf{p}_i\right\|\right)$.
(b) If $\mathbf{s}^k = \mathbf{p}_1$, then $\mathbf{p}_1 = \mathbf{v}\left(\mathbf{p}_i, \mathbf{e}_1, 1/\gamma_i^k\right)$ for all $i \neq 1$.

In both cases, the points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$ are co-linear, in contrary to Assumption 1(c).

(ii): Since $\tilde{\mathbf{u}}$ is a limit point of $\left\{\mathbf{u}^k\right\}_{k\geq 0}$ (which exists since this sequence is bounded by its definition in (10) and (11)), there exists a sub-sequence $\left\{\mathbf{u}^{k_j}\right\}_{j\geq 0}$ that converges to $\tilde{\mathbf{u}}$. Since $\left\{\mathbf{s}^{k_j}\right\}_{j\geq 0}$ is bounded, it has a sub-sequence that converges to some point $\tilde{\mathbf{s}} \in \mathbb{R}^n$. Hence, for simplicity, by possibly passing to sub-sequences, we assume that $\left\{\left(\mathbf{s}^k, \mathbf{u}^k\right)\right\}_{k\geq 0}$ converges to $\left(\tilde{\mathbf{s}}, \tilde{\mathbf{u}}\right)$.

Assume on the contrary that $\tilde{\mathbf{u}}_i^T \tilde{\mathbf{u}}_1 = -1$ for all $i \neq 1$. Using similar arguments, we derive that there exist non-zero scalars $\gamma_2, \gamma_3, \ldots, \gamma_N$ (independent of $k \geq 0$) such that $\tilde{\mathbf{u}}_1 = \gamma_i \tilde{\mathbf{u}}_i$ for all $i \neq 1$. Again, we arrive at two cases:

(a) If $\tilde{\mathbf{s}} = \mathbf{p}_j$ for some $j = 2, 3, \ldots, N$, then $\tilde{\mathbf{s}} \neq \mathbf{p}_i$ for any $i \neq j$ and $\mathbf{u}_i^k \to \tilde{\mathbf{u}}_i = \left(\tilde{\mathbf{s}} - \mathbf{p}_i\right) / \left\|\tilde{\mathbf{s}} - \mathbf{p}_i\right\|$ as $k \to \infty$. Therefore, $\mathbf{p}_1 = \mathbf{v}\left(\tilde{\mathbf{s}}, \tilde{\mathbf{s}} - \mathbf{p}_i, \gamma_i^k / \left\|\tilde{\mathbf{s}} - \mathbf{p}_i\right\|\right)$.
(c) If $\tilde{\mathbf{s}} = \mathbf{p}_1$, then $\mathbf{p}_1 = \mathbf{v}\left(\mathbf{p}_i, \tilde{\mathbf{u}}_1, 1/\gamma_i\right)$ for all $i \neq 1$.

In both cases, the points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$ are co-linear, in contrary to Assumption 1(c). □

Proof of Lemma 1. First, we prove that $\mathbf{A}^k$ is positive-definite (PD) for all $k \geq 0$. To this end, notice that for any two non-zero vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ we have $\mathbf{x}\mathbf{y}^T\mathbf{x} = \mathbf{x}^T\mathbf{y}\mathbf{x}$, and therefore $\mathbf{x}^T\mathbf{y}$ is an eigenvalue of the matrix $\mathbf{x}\mathbf{y}^T \in \mathbb{R}^{n\times n}$. Moreover, for any $\mathbf{v} \in \mathbb{R}^n$ we have $\mathbf{x}\mathbf{y}^T\mathbf{v} = \mathbf{v}^T\mathbf{y}\mathbf{x} \in \operatorname{span}(\mathbf{x})$ and hence $\operatorname{rank}\left(\mathbf{x}\mathbf{y}^T\right) = 1$. Therefore, all other eigenvalues are 0. In particular, for the matrices $\mathbf{u}_i^k\left(\mathbf{u}_1^k\right)^T$, $i = 2, 3, \ldots, N$ and $k \geq 0$, it holds that

$$\begin{aligned}\lambda_{\min}\left(\mathbf{u}_i^k\left(\mathbf{u}_1^k\right)^T\right) &= \min\left\{0, \left(\mathbf{u}_i^k\right)^T\mathbf{u}_1^k\right\} \\ &\leq \max\left\{0, \left(\mathbf{u}_i^k\right)^T\mathbf{u}_1^k\right\} = \lambda_{\max}\left(\mathbf{u}_i^k\left(\mathbf{u}_1^k\right)^T\right).\end{aligned} \tag{A.2}$$

Now, using Weyl's inequality [53, Theorem 4.3.1], we derive from (A.2) that for all $k \geq 0$

$$\begin{aligned}\lambda_{\min}\left(\mathbf{A}^k\right) &\geq (N-1) + \sum_{i=2}^{N} \lambda_{\min}\left(\mathbf{u}_i^k\left(\mathbf{u}_1^k\right)^T\right) \\ &= N - 1 + \sum_{i=2}^{N} \min\left\{0, \left(\mathbf{u}_i^k\right)^T\mathbf{u}_1^k\right\}.\end{aligned} \tag{A.3}$$

From Lemma A(i) we have that $\left(\mathbf{u}_j^k\right)^T\mathbf{u}_1^k > -1$ for some $j = 2, 3, \ldots, N$, and thus from (A.3) it follows that

$$\begin{aligned}\lambda_{\min}\left(\mathbf{A}^k\right) &> N - 2 - \sum_{i=2, i\neq j}^{N} \min\left\{0, \left\|\mathbf{u}_i^k\right\| \cdot \left\|\mathbf{u}_1^k\right\|\right\} \\ &\geq N - 2 + (N-2)\cdot\min\{0, -1\} = 0,\end{aligned}$$

and hence $\mathbf{A}^k$ is PD.

Now, we will prove that there exists $\sigma > 0$ such that $\lambda_{\min}\left(\mathbf{A}^k\right) \geq \sigma$ for any $k \geq 0$. Assume on the contrary that there exists a sub-sequence $\left\{\mathbf{A}^{k_j}\right\}_{j\geq 0}$ that satisfies $\lambda_{\min}\left(\mathbf{A}^{k_j}\right) \to 0$ as $j \to \infty$. In this case, from (A.3) we get

$$\lim_{l\to\infty}\left(\mathbf{u}_i^{k_j}\right)^T\mathbf{u}_1^{k_j} = -1, \quad \forall i = 2, 3, \ldots, N. \tag{A.4}$$

Since the sequence $\left\{\left(\mathbf{s}^{k_j}, \mathbf{u}^{k_j}\right)\right\}_{j\geq 0}$ is bounded, then by possibly passing to a convergent sub-sequence, we can assume that $\left(\mathbf{s}^{k_j}, \mathbf{u}^{k_j}\right) \to \left(\tilde{\mathbf{s}}, \tilde{\mathbf{u}}\right)$ as $j \to \infty$, for some $\tilde{\mathbf{s}} \in \mathbb{R}^n$ and $\tilde{\mathbf{u}} \in \mathcal{B}^N$. Therefore, (A.4) contradicts Lemma A(ii).

Finally, we prove that $\lambda_{\max}\left(\mathbf{A}^k\right) \leq 2(N-1)$. By similar arguments to those presented above it holds that

$$\begin{aligned}\lambda_{\max}\left(\mathbf{A}^k\right) &\leq (N-1) + \sum_{i=2}^{N}\lambda_{\max}\left(\mathbf{u}_i^k\left(\mathbf{u}_1^k\right)^T\right) \\ &\leq N - 1 + \sum_{i=2}^{N}\max\{0, 1\} = 2(N-1),\end{aligned}$$

and the proof is now completed. □

Appendix B. Proof of Propositions 1 and 2

A direct consequence of Lemma 1 is that the partial function $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \mathbf{u}^k\right)$ is 2σ-strongly convex for any $k \geq 0$ (see [47, Example 5.19]), and as such it admits a unique minimizer $\mathbf{s}_*^k \in \mathbb{R}^n$.

For simplicity, we assume that (24) holds true starting from $K = 0$. Moreover, from now on we denote $\kappa \equiv 2(N-1)/\sigma$ and it follows from (24) that $j_k \geq 2\sqrt{\kappa}$.

In Lemma B below, we first prove some properties of the sequence $\left\{\left(\mathbf{s}^k, \mathbf{u}^k\right)\right\}_{k\geq 0}$ generated by T-NAM.

Lemma B. *Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by T-NAM and let $\left\{\mathbf{y}^{k,j}\right\}_{j=0}^{j_k-1}$, $k \geq 0$, be the auxiliary sequences generated by FISTA (see Algorithm 2). Then, for any $k \geq 0$, it holds that*

(i) $\left\|\mathbf{s}^{k,j} - \mathbf{s}_*^k\right\| \leq \frac{2\sqrt{\kappa}}{j+1}\left\|\mathbf{s}^k - \mathbf{s}_*^k\right\|$ *for any $j \geq 0$.*

In addition, there exists $M > 0$ such that

(ii) $\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\| \le M$.
(iii) $\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| \le M$.
(iv) $\left\|\mathbf{s}^{k+1} - \mathbf{y}^{k,j_k-1}\right\| \le \frac{8M\sqrt{\kappa}}{j_k-1}$.

Proof. Let $k \ge 0$. From the 2σ-strong convexity of the function $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \mathbf{u}^{k+1}\right)$ we have for any $j \ge 0$ that

$$\Psi\left(\mathbf{s}^{k,j}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) \ge \sigma \left\|\mathbf{s}^{k,j} - \mathbf{s}^{k}_{*}\right\|^2 . \tag{B.1}$$

In addition, from a classical rate of convergence result of FISTA (see [46]) we have

$$\Psi\left(\mathbf{s}^{k,j}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) \le \frac{2L^k}{(j+1)^2}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 \le \frac{8\,(N-1)}{(j+1)^2}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 , \tag{B.2}$$

where the last inequality follows from the fact that $L^k \le 4\,(N-1)$, which follows from Lemma 1. Combining (B.1) and (B.2) yields (recall that $\kappa = 2\,(N-1)/\sigma$)

$$\left\|\mathbf{s}^{k,j} - \mathbf{s}^{k}_{*}\right\| \le \frac{2\sqrt{\kappa}}{j+1}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| ,$$

and item (i) is established. For items (ii) and (iii), since the sequence $\left\{\mathbf{s}^{k}\right\}_{k\ge 0}$ is bounded, there exists $C_1 > 0$ such that for any $k \ge 0$ it holds that

$$\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\| \le C_1 . \tag{B.3}$$

Recalling that $\mathbf{s}^{k+1} \equiv \mathbf{s}^{k,j_k}$ by its definition (see Algorithm 2), then plugging $j = j_k \ge 2\sqrt{\kappa}$ (see (22)) in item (i) yields

$$\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| = \left\|\mathbf{s}^{k,j_k} - \mathbf{s}^{k}_{*}\right\| \le \frac{2\sqrt{\kappa}}{2\sqrt{\kappa}+1}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| . \tag{B.4}$$

From the triangle inequality with (B.3) and (B.4) we get

$$\begin{aligned}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| &\le \left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| + \left\|\mathbf{s}^{k} - \mathbf{s}^{k+1}\right\| \\ &\le \frac{2\sqrt{\kappa}}{2\sqrt{\kappa}+1}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| + C_1 ,\end{aligned}$$

which implies that (recall that $\sqrt{\kappa} \ge 1$)

$$\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\| \le C_1\left(2\sqrt{\kappa}+1\right) \equiv M . \tag{B.5}$$

Now, since $M > C_1$, we establish items (ii) and (iii) from (B.3) and (B.5), respectively.

Last, we prove item (iv). Recall that for any $j \ge 1$ it holds that (see step 6 in Algorithm 2)

$$\mathbf{y}^{k,j} = \mathbf{s}^{k,j} + \frac{t_{j-1}-1}{t_j}\left(\mathbf{s}^{k,j} - \mathbf{s}^{k,j-1}\right) . \tag{B.6}$$

Now, since $\left(t_{j_k-2}-1\right)/t_{j_k-1} \le 1$ for any $j_k \ge 2$ (see (22)), then by adding and subtracting $2\mathbf{s}^{k}_{*}$ from the RHS of (B.6), we obtain from the triangle inequality that

$$\begin{aligned}\left\|\mathbf{s}^{k+1} - \mathbf{y}^{k,j_k-1}\right\| &\le \left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| \\ &+ 2\left\|\mathbf{s}^{k,j_k-1} - \mathbf{s}^{k}_{*}\right\| + \left\|\mathbf{s}^{k,j_k-2} - \mathbf{s}^{k}_{*}\right\| .\end{aligned} \tag{B.7}$$

Recalling that $\mathbf{s}^{k+1} \equiv \mathbf{s}^{k,j_k}$, then by combining item (i) with item (iii) and (B.7) we get

$$\left\|\mathbf{s}^{k+1} - \mathbf{y}^{k,j_k-1}\right\| \le \frac{8M\sqrt{\kappa}}{j_k-1},$$

which is the required result. □

Proof of Proposition 1. Let $k \ge 0$. First, we prove that $\left\{\Psi(\mathbf{s}^{k}, \mathbf{u}^{k})\right\}_{k\ge 0}$ is non-increasing by first proving that

$$\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k}\right) \le \Psi\left(\mathbf{s}^{k}, \mathbf{u}^{k}\right) . \tag{B.8}$$

From the 2σ-strong convexity of $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \mathbf{u}^{k}\right)$ we get

$$\Psi\left(\mathbf{s}^{k}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) \ge \sigma\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 . \tag{B.9}$$

In addition, from a classical rate of convergence result of FISTA (see [46]) we have

$$\begin{aligned}\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) &\le \frac{2L^k\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2}{(j_k+1)^2} \\ &\le \frac{8\,(N-1)}{\left(2\sqrt{\kappa}+1\right)^2}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 ,\end{aligned} \tag{B.10}$$

where the last inequality follows from (24) and the fact that $L^k \le 4\,(N-1)$ (see Lemma 1). Combining (B.9) and (B.10) yields

$$\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) \le \frac{4\kappa}{\left(2\sqrt{\kappa}+1\right)^2}\left(\Psi\left(\mathbf{s}^{k}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right)\right),$$

and the required inequality (B.8) follows. Now, since $\mathbf{u}^{k+1}_{i}$, $i = 1, 2, \ldots, N$, is a minimizer of $\mathbf{u}_i \mapsto \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}\right)$, then

$$\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}\right) \le \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k}\right) . \tag{B.11}$$

Combining (B.8) with (B.11), we obtain that the sequence $\left\{\Psi(\mathbf{s}^{k}, \mathbf{u}^{k})\right\}_{k\ge 0}$ is non-increasing, as required.

Now, we prove the required inequality (25). From (B.9) it holds that

$$\begin{aligned}\sigma\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 &\le \Psi\left(\mathbf{s}^{k}, \mathbf{u}^{k}\right) - \Psi\left(\mathbf{s}^{k}_{*}, \mathbf{u}^{k}\right) \\ &+ \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}\right) - \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}\right) .\end{aligned} \tag{B.12}$$

From the triangle and Cauchy-Schwartz inequalities we get

$$\begin{aligned}\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 &\ge \left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\|^2 \\ &- 2\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| \cdot \left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\| \\ &\ge \left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\|^2 - 2M\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| ,\end{aligned} \tag{B.13}$$

where the last inequality follows from Lemma B(ii). Multiplying both sides of (B.13) by $\sigma > 0$ and using the fact that $2\,(N-1) \ge \sigma$ (see Lemma 1), we get

$$\begin{aligned}\sigma\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 &\ge \sigma\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\|^2 \\ &- 4\,(N-1)\,M\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}_{*}\right\| .\end{aligned} \tag{B.14}$$

Applying Lemma B(i) and then Lemma B(iii) on the RHS of (B.14) we get

$$\begin{aligned}\sigma\left\|\mathbf{s}^{k} - \mathbf{s}^{k}_{*}\right\|^2 &\ge \sigma\left\|\mathbf{s}^{k+1} - \mathbf{s}^{k}\right\|^2 \\ &- \frac{8M^2\sqrt{\kappa}\,(N-1)}{j_k+1} .\end{aligned} \tag{B.15}$$

Combining (B.10), (B.11), (B.12) and (B.15) we get

$$\Psi\left(\mathbf{s}^k,\mathbf{u}^k\right)-\Psi\left(\mathbf{s}^{k+1},\mathbf{u}^{k+1}\right)$$
$$\geq \sigma\left\|\mathbf{s}^{k+1}-\mathbf{s}^k\right\|^2-\frac{8M^2\sqrt{\kappa}\left(N-1\right)}{j_k+1}-\frac{8M^2\left(N-1\right)}{\left(j_k+1\right)^2},$$

where we used Lemma B(iii). Since $\sqrt{\kappa}\geq 1$ and since $\left(j_k+1\right)^2\geq\left(j_k+1\right)$ we finally get

$$\Psi\left(\mathbf{s}^k,\mathbf{u}^k\right)-\Psi\left(\mathbf{s}^{k+1},\mathbf{u}^{k+1}\right)\geq\sigma\left\|\mathbf{s}^{k+1}-\mathbf{s}^k\right\|^2$$
$$-\frac{16M^2\sqrt{\kappa}\left(N-1\right)}{j_k+1}.$$

Now the result follows with $M_1\equiv 16M^2\sqrt{\kappa}\left(N-1\right)$. □

Proof of Proposition 2. Using (21) and the first-order optimality condition applied on (18), we obtain that the following inclusion holds for any $k\geq 0$

$$L^k\left(\mathbf{y}^{k,j_k-1}-\mathbf{s}^{k+1}\right)-\left(2\mathbf{A}^k\mathbf{y}^{k,j_k-1}-\mathbf{z}^k\right)\in\partial\psi\left(\mathbf{s}^{k+1}\right).$$

Therefore, we derive that the following inclusion holds

$$\mathbf{w}_{\mathbf{s}}^{k+1}\equiv L^k\left(\mathbf{y}^{k,j_k-1}-\mathbf{s}^{k+1}\right)+2\mathbf{A}^k\left(\mathbf{s}^{k+1}-\mathbf{y}^{k,j_k-1}\right)$$
$$\in\partial_{\mathbf{s}}\Psi\left(\mathbf{s}^{k+1},\mathbf{u}^k\right).$$

Recall that $L^k\leq 4\left(N-1\right)$ (see Lemma 1). Hence,

$$\left\|\mathbf{w}_{\mathbf{s}}^{k+1}\right\|\leq 8\left(N-1\right)\left\|\mathbf{s}^{k+1}-\mathbf{y}^{k,j_k-1}\right\|. \tag{B.16}$$

Now, notice that minimizing Ψ over its domain is the same as minimizing the unconstrained function $\Psi\left(\mathbf{s},\mathbf{u}\right)+\sum_{i=1}^N\delta\left(\mathbf{u}_i\right)$, where $\delta\colon\mathbb{R}^n\to\mathbb{R}$ is the indicator function of the ball defined as $\delta\left(\mathbf{v}\right)=0$ if $\mathbf{v}\in\mathcal{B}$ and $+\infty$ otherwise. Hence, since $\mathbf{u}^k$ is a minimizer of the $\mathbf{u}\mapsto\Psi\left(\mathbf{s}^k,\mathbf{u}\right)$, then from the first-order optimality condition we have

$$-\nabla_{\mathbf{u}}\varphi\left(\mathbf{s}^k,\mathbf{u}^k\right)\in\partial\sum_{i=1}^N\delta\left(\mathbf{u}_i^k\right).$$

Therefore, we derive that the following inclusion holds

$$\mathbf{w}_{\mathbf{u}}^{k+1}\equiv\nabla_{\mathbf{u}}\varphi\left(\mathbf{s}^{k+1},\mathbf{u}^k\right)-\nabla_{\mathbf{u}}\varphi\left(\mathbf{s}^k,\mathbf{u}^k\right)\in\partial_{\mathbf{u}}\Psi\left(\mathbf{s}^{k+1},\mathbf{u}^k\right).$$

Notice that since φ is a quadratic function, then it has an $\mathcal{L}$-Lipschitz continuous gradient with some $\mathcal{L}>0$. Hence,

$$\left\|\mathbf{w}_{\mathbf{u}}^{k+1}\right\|\leq\mathcal{L}\left\|\mathbf{s}^{k+1}-\mathbf{s}^k\right\|. \tag{B.17}$$

Now, denote $\mathbf{w}^{k+1}\equiv\left(\mathbf{w}_{\mathbf{s}}^{k+1},\mathbf{w}_{\mathbf{u}}^{k+1}\right)\in\partial\Psi\left(\mathbf{s}^{k+1},\mathbf{u}^k\right)$ for all $k\geq 0$. Combining (B.16) and (B.17) we have

$$\left\|\mathbf{w}^{k+1}\right\|\leq\left\|\mathbf{w}_{\mathbf{s}}^{k+1}\right\|+\left\|\mathbf{w}_{\mathbf{u}}^{k+1}\right\|$$
$$\leq\frac{64M\left(N-1\right)\sqrt{\kappa}}{j_k-1}+\mathcal{L}\left\|\mathbf{s}^{k+1}-\mathbf{s}^k\right\|,$$

where the last inequality follows from Lemma B(iv). Now, the result follows by setting $M_2\equiv 64M\left(N-1\right)\sqrt{\kappa}$. □

Appendix C. Proof of Lemma 2

Here we prove technical results on the Fréchet and limiting sub-differential sets that may be used also in other contexts. We refer the reader to [16] for the exact definition of the Fréchet sub-differential, denoted here by $\hat{\partial}$, and the limiting sub-differential, denoted by ∂.

We first prove the following result that will later be useful in proving Lemma 2.

Lemma C. *Let $\bar{\mathbf{x}}\in\mathbb{R}^n$ and let $g\colon\mathbb{R}^n\to\mathbb{R}$ be a continuous function and set $h\left(\mathbf{x}\right)\equiv g\left(\mathbf{x}\right)\cdot\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|$. Then,*

$$\hat{\partial}h\left(\bar{\mathbf{x}}\right)=\begin{cases}g\left(\bar{\mathbf{x}}\right)\cdot\mathcal{B}, & g\left(\bar{\mathbf{x}}\right)\geq 0,\\ \varnothing, & g\left(\bar{\mathbf{x}}\right)<0.\end{cases}$$

Proof. We first prove that if $\mathbf{v}\in\mathbb{R}^n$ satisfies $\left\|\mathbf{v}\right\|\leq g\left(\bar{\mathbf{x}}\right)$, then $\mathbf{v}\in\hat{\partial}h\left(\bar{\mathbf{x}}\right)$. Using the definition of h together with the Cauchy-Schwartz inequality we have

$$\liminf_{\mathbf{x}\to\bar{\mathbf{x}}}\frac{h\left(\mathbf{x}\right)-h\left(\bar{\mathbf{x}}\right)-\mathbf{v}^T\left(\mathbf{x}-\bar{\mathbf{x}}\right)}{\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|}$$
$$\geq\liminf_{\mathbf{x}\to\bar{\mathbf{x}}}\frac{g\left(\mathbf{x}\right)\cdot\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|-\left\|\mathbf{v}\right\|\cdot\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|}{\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|}=g\left(\bar{\mathbf{x}}\right)-\left\|\mathbf{v}\right\|.$$

Hence, by the definition of $\partial\hat{h}\left(\bar{\mathbf{x}}\right)$ we get that $\mathbf{v}\in\partial\hat{h}\left(\bar{\mathbf{x}}\right)$. Conversely, let $\mathbf{v}\in\partial\hat{h}\left(\bar{\mathbf{x}}\right)$. By the definition of $\partial\hat{h}\left(\bar{\mathbf{x}}\right)$, for any $\left\{\mathbf{x}^k\right\}_{k\geq 1}$ such that $\mathbf{x}^k\to\bar{\mathbf{x}}$ as $k\to\infty$, we have

$$0\leq\liminf_{k\to\infty}\frac{h\left(\mathbf{x}^k\right)-h\left(\bar{\mathbf{x}}\right)-\mathbf{v}^T\left(\mathbf{x}^k-\bar{\mathbf{x}}\right)}{\left\|\mathbf{x}^k-\bar{\mathbf{x}}\right\|}$$
$$=\liminf_{k\to\infty}\frac{g\left(\mathbf{x}^k\right)\cdot\left\|\mathbf{x}^k-\bar{\mathbf{x}}\right\|-\mathbf{v}^T\left(\mathbf{x}^k-\bar{\mathbf{x}}\right)}{\left\|\mathbf{x}^k-\bar{\mathbf{x}}\right\|}.$$

In particular, we take $\mathbf{x}^k=\mathbf{v}/k+\bar{\mathbf{x}}$ and then $\mathbf{x}^k\to\bar{\mathbf{x}}$ as $k\to\infty$. Therefore, we obtain that

$$0\leq\liminf_{k\to\infty}\frac{g\left(\mathbf{v}/k+\bar{\mathbf{x}}\right)\cdot\left\|\mathbf{v}\right\|-\left\|\mathbf{v}\right\|^2}{\left\|\mathbf{v}\right\|}$$
$$=\liminf_{k\to\infty}g\left(\mathbf{v}/k+\bar{\mathbf{x}}\right)-\left\|\mathbf{v}\right\|=g\left(\bar{\mathbf{x}}\right)-\left\|\mathbf{v}\right\|,$$

where the last equality follows from the continuity of g. Thus, we derive that $\left\|\mathbf{v}\right\|\leq g\left(\bar{\mathbf{x}}\right)$.

The two implications imply the desired result when $g\left(\bar{x}\right)\geq 0$. If $g\left(\bar{x}\right)<0$, then $0\leq\left\|\mathbf{v}\right\|\leq g\left(\bar{\mathbf{x}}\right)<0$, which is a contradiction. Therefore, $\partial\hat{h}\left(\bar{\mathbf{x}}\right)=\varnothing$, as required. □

Proof of Lemma 2. For any $\mathbf{x}\in U$ such that $\mathbf{x}\neq\bar{\mathbf{x}}$, the function h is differentiable at $\mathbf{x}$ and

$$\nabla h\left(\mathbf{x}\right)=\nabla g\left(\mathbf{x}\right)\cdot\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|+g\left(\mathbf{x}\right)\cdot\frac{\mathbf{x}-\bar{\mathbf{x}}}{\left\|\mathbf{x}-\bar{\mathbf{x}}\right\|}. \tag{C.1}$$

Let $\mathbf{v}\in\partial h\left(\bar{\mathbf{x}}\right)$, and consider the sequences $\left\{\mathbf{x}^k\right\}_{k\geq 1}\subseteq U$ and $\left\{\mathbf{v}^k\right\}_{k\geq 1}$, $\mathbf{v}^k\in\hat{\partial}h\left(\mathbf{x}^k\right)$, such that $\mathbf{x}^k\to\bar{\mathbf{x}}$ and $\mathbf{v}^k\to\mathbf{v}\in\mathbb{R}^n$ as $k\to\infty$.

First, assume that $\mathbf{x}^k\neq\bar{\mathbf{x}}$ for all $k\geq K$ for some $K\in\mathbb{N}$. We will prove that in this case $\mathbf{v}\in g\left(\bar{\mathbf{x}}\right)\cdot\mathcal{S}$. Since h is differentiable at any $\mathbf{x}^k$, then $\hat{\partial}h\left(\mathbf{x}^k\right)=\left\{\nabla h\left(\mathbf{x}^k\right)\right\}$ and it follows from (C.1) that

$$\lim_{k\to\infty}\mathbf{v}^k=\lim_{k\to\infty}\nabla h\left(\mathbf{x}^k\right)$$
$$=\lim_{k\to\infty}\left(\nabla g\left(\mathbf{x}^k\right)\left\|\mathbf{x}^k-\bar{\mathbf{x}}\right\|+g\left(\mathbf{x}^k\right)\frac{\mathbf{x}^k-\bar{\mathbf{x}}}{\left\|\mathbf{x}^k-\bar{\mathbf{x}}\right\|}\right). \tag{C.2}$$

Since the set $\mathcal{S}$ is closed, there exists $\mathbf{g} \in \mathcal{S}$ such that

$$\lim_{k\to\infty} \frac{\mathbf{x}^k - \bar{\mathbf{x}}}{\left\|\mathbf{x}^k - \bar{\mathbf{x}}\right\|} = \mathbf{g} \in \mathcal{S}. \tag{C.3}$$

Combining (C.2) and (C.3), it follows from the continuity of g and ∇g that $\mathbf{v} = g\left(\bar{\mathbf{x}}\right) \cdot \mathbf{g} \in g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{S} \subset g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B}$ (recall that $\mathbf{v}^k \to \mathbf{v}$ as $k \to \infty$), which is the required inclusion.

Now, assume that $\mathbf{x}^k = \bar{\mathbf{x}}$ for infinitely many indices $k \geq 1$. Hence, $\mathbf{v}^k \in \hat{\partial} h\left(\bar{\mathbf{x}}\right)$ infinitely many times. If $g\left(\bar{\mathbf{x}}\right) < 0$, then from Lemma C we have $\mathbf{v}^k \in \varnothing$, which contradicts the fact that $\left\{\mathbf{v}^k\right\}_{k\geq 1}$ converges to $\mathbf{v} \in \mathbb{R}^n$. Therefore, $g\left(\bar{\mathbf{x}}\right) \geq 0$, and from Lemma C it follows that $\mathbf{v}^k \in \hat{\partial} h\left(\bar{\mathbf{x}}\right) = g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B}$ infinitely many times. Since the set $g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B}$ is closed, we conclude that $\mathbf{v} \in g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B}$.

Now we prove the other inclusion. If $g\left(\bar{\mathbf{x}}\right) \geq 0$, then Lemma C implies $g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B} = \hat{\partial} h\left(\bar{\mathbf{x}}\right) \subseteq \partial h\left(\bar{\mathbf{x}}\right)$ (see [54, Chapter 6]) and hence $g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{B} = \partial h\left(\bar{\mathbf{x}}\right)$, as required. Assume that $g\left(\bar{\mathbf{x}}\right) < 0$ and let $\mathbf{w} \in g\left(\bar{\mathbf{x}}\right) \cdot \mathcal{S}$. We will prove that $\mathbf{w} \in \partial h\left(\bar{\mathbf{x}}\right)$. By the definition of $\partial h\left(\bar{\mathbf{x}}\right)$, we need to show that there exist sequences $\left\{\mathbf{x}^k\right\}_{k\geq 1}$ and $\left\{\mathbf{w}^k\right\}_{k\geq 1}$, $\mathbf{w}^k \in \hat{\partial} h\left(\mathbf{x}^k\right)$, such that $\mathbf{x}^k \to \bar{\mathbf{x}}$ and $\mathbf{w}^k \to \mathbf{w}$ as $k \to \infty$. To this end, we define $\mathbf{x}^k = \bar{\mathbf{x}} + \mathbf{w}/\left(k \cdot g\left(\bar{\mathbf{x}}\right)\right)$, and notice that $\mathbf{x}^k \neq \bar{\mathbf{x}}$ for all $k \geq 1$ (since $\mathbf{w} \neq \mathbf{0}_n$). Therefore, $\hat{\partial} h\left(\mathbf{x}^k\right) = \left\{\nabla h\left(\mathbf{x}^k\right)\right\}$. Plugging $\mathbf{x} = \mathbf{x}^k$ in (C.1) yields (recall that $\|\mathbf{w}\| = \left|g\left(\bar{\mathbf{x}}\right)\right|$)

$$\lim_{k\to\infty} \nabla h\left(\mathbf{x}^k\right) = \lim_{k\to\infty} \left(\frac{1}{k}\nabla g\left(\mathbf{x}^k\right) - g\left(\mathbf{x}^k\right)\frac{\mathbf{w}}{\left|g\left(\bar{\mathbf{x}}\right)\right|}\right) = \mathbf{w},$$

and we derive that $\mathbf{w} \in \partial h\left(\bar{\mathbf{x}}\right)$, as required. □

Appendix D. SOLVIT and SDP

SOLVIT algorithm Following the notations in [12] for the case of a single reference sensor, we denote for any $i = 2, 3, \ldots, N$ and $k \geq 0$ the matrix

$$\mathbf{M}_i^k \equiv \left(2 + d_i\right)\mathbf{I}_n - \mathbf{Q}_i^k - \left(\mathbf{Q}_i^k\right)^T \in \mathbb{R}^{n\times n},$$

where

$$\mathbf{Q}_i^k \equiv \frac{\left(\mathbf{s}^k - \mathbf{p}_1\right)\left(\mathbf{s}^k - \mathbf{p}_i\right)^T}{\left\|\mathbf{s}^k - \mathbf{p}_1\right\|\left\|\mathbf{s}^k - \mathbf{p}_i\right\|} \in \mathbb{R}^{n\times n}.$$

We also denote the vector

$$\mathbf{b}_i \equiv \mathbf{p}_i + \mathbf{p}_1 + d_i\frac{\mathbf{s}^k - \mathbf{p}_i}{\left\|\mathbf{s}^k - \mathbf{p}_i\right\|} + \frac{d_i}{\left\|\mathbf{s}^k - \mathbf{p}_1\right\|}\mathbf{p}_1$$
$$- \mathbf{Q}_i^k\mathbf{p}_i - \left(\mathbf{Q}_i^k\right)^T\mathbf{p}_1 \in \mathbb{R}^n.$$

Following [12], the next iteration is given by the formula

$$\mathbf{s}^{k+1} = \left(\sum_{i=2}^{N}\mathbf{M}_i^k\right)^{-1}\left(\sum_{i=2}^{N}\mathbf{b}_i\right).$$

SDP formulation Following the derivations in [17], an SDP problem formulation that is suitable for the setting of the TDoA problem discussed in this paper is given by

$$\begin{aligned}
\min_{\mathbf{s},\mathbf{q},\mathbf{Q},Y} \quad & \frac{1}{2}\sum_{i=1}^{N}\left(\mathbf{q}_i - \mathbf{q}_1 - d_i\right)^2 \\
\text{s.t.} \quad & \left\|\mathbf{s} - \mathbf{p}_i\right\| \leq \mathbf{q}_i, \quad \forall i, \\
& \mathbf{Q}_{ii} = Y - 2\mathbf{p}_i^T\mathbf{s} + \left\|\mathbf{p}_i\right\|^2, \quad \forall i, \\
& \mathbf{Q}_{ij} \geq \left|Y - \left(\mathbf{p}_i + \mathbf{p}_j\right)^T\mathbf{s} + \mathbf{p}_i^T\mathbf{p}_j\right|, \quad \forall i, j, \\
& Y \geq \|\mathbf{s}\|^2, \\
& \mathbf{Q} \succeq \mathbf{q}\mathbf{q}^T.
\end{aligned}$$

Appendix E. RMSE and CRLB

The estimated RMSE Recall that for an unbiased estimator $\hat{\mathbf{s}} \in \mathbb{R}^n$ of $\mathbf{s}^{\text{true}} \in \mathbb{R}^n$, the CRLB states that the matrix $\operatorname{cov}\left(\hat{\mathbf{s}}\right) - \mathbf{F}^{-1}\left(\mathbf{s}^{\text{true}}\right)$ is positive semi-definite, where $\mathbf{F}\left(\mathbf{s}^{\text{true}}\right) \in \mathbb{R}^{n\times n}$ is the Fisher Information Matrix (FIM), and where the covariance matrix is given by

$$\operatorname{cov}\left(\hat{\mathbf{s}}\right) \equiv \mathbb{E}\left[\left(\hat{\mathbf{s}} - \mathbb{E}\left[\hat{\mathbf{s}}\right]\right)\left(\hat{\mathbf{s}} - \mathbb{E}\left[\hat{\mathbf{s}}\right]\right)^T\right] \in \mathbb{R}^{n\times n}.$$

In order to obtain a single scalar for comparison, we use the following (looser) inequality

$$\sqrt{\operatorname{trace}\left(\operatorname{cov}\left(\hat{\mathbf{s}}\right)\right)} \geq \sqrt{\operatorname{trace}\left(\mathbf{F}^{-1}\left(\mathbf{s}^{\text{true}}\right)\right)} \equiv \text{CRLB},$$

where the LHS is called the RMSE. Since the expectation function is unknown, we consider instead the estimated RMSE given by

$$\widehat{\text{RMSE}} \equiv \sqrt{\operatorname{trace}\left(\frac{1}{R}\sum_{r=1}^{R}\left(\left(\hat{\mathbf{s}}^r - \bar{\mathbf{s}}\right)\left(\hat{\mathbf{s}}^r - \bar{\mathbf{s}}\right)^T\right)\right)},$$

where $\hat{\mathbf{s}}^r \in \mathbb{R}^n$ is a solution obtained by the algorithm (i.e., its output point) in the r-th trial, and $\bar{\mathbf{s}} \equiv (1/R)\sum_{r=1}^{R}\hat{\mathbf{s}}^r \in \mathbb{R}^n$ is the empirical mean.

Calculating the CRLB For obtaining the matrix $\mathbf{F}\left(\mathbf{s}\right)$, we notice that since $\tilde{\epsilon}_i \sim \text{Normal}\left(0, \sigma_{\text{dist}}^2\right)$, then simple calculations yield for any $i, j = 2, 3, \ldots, N$, $i \neq j$, that $\operatorname{cov}\left(\epsilon_i\epsilon_j\right) = \sigma_{\text{dist}}^2$ and $\operatorname{cov}\left(\epsilon_i^2\right) = 2\sigma_{\text{dist}}^2$. Hence, if $\mathbf{d} \in \mathbb{R}^{N-1}$ is the vector of all measurements, we obtain that $\mathbf{d} \sim \text{Normal}\left(\boldsymbol{\mu}\left(\mathbf{s}\right), \boldsymbol{\Sigma}\right)$, where each coordinate of $\boldsymbol{\mu}\left(\mathbf{s}\right) \in \mathbb{R}^{N-1}$ satisfies $\left(\boldsymbol{\mu}\left(\mathbf{s}\right)\right)_i = \left\|\mathbf{s} - \mathbf{p}_i\right\| - \left\|\mathbf{s} - \mathbf{p}_1\right\|$, $i \neq 1$. Moreover, $\boldsymbol{\Sigma} = \sigma_{\text{dist}}^2\mathbf{1}_{N-1} + \sigma_{\text{dist}}^2\mathbf{I}_{N-1}$ where $\mathbf{1}_{N-1}$ is the square matrix of all ones of size $N-1$ and $\mathbf{I}_{N-1}$ is the identity matrix of the same size. Hence, it holds that [51]

$$\mathbf{F}\left(\mathbf{s}\right) = \mathbf{J}\left(\mathbf{s}\right)^T\boldsymbol{\Sigma}^{-1}\mathbf{J}\left(\mathbf{s}\right),$$

where $\mathbf{J}\left(\mathbf{s}\right) \in \mathbb{R}^{(N-1)\times n}$ is the Jacobian of $\boldsymbol{\mu}\colon \mathbb{R}^n \to \mathbb{R}^{N-1}$, assuming that $\mathbf{s} \neq \mathbf{p}_i$ for all $i = 1, 2, \ldots, N$. We mention that the CRLB above considers the true covariance matrix $\boldsymbol{\Sigma}$, while in Problem (LS) it was assumed to be unknown and was estimated as the identity.

Recall that we require all measurements d_i, $i \neq 1$, to be nonnegative, which might result with a different selection of reference sensor in each realization (notice that $\boldsymbol{\mu}$ depends on the selected reference sensor). Following [52], selection of reference sensor does not affect the FIM, and hence the CRLB is independent of the realizations.

References

[1] H.C. So, Source Localization: Algorithms and Analysis, John Wiley & Sons, Ltd, 2018, pp. 59–106.

[2] J. Smith, J. Abel, Closed-form least-squares source location estimation from range-difference measurements, IEEE Trans. Acoust. Speech Signal Process. 35 (1987) 1661–1669.

[3] Y.-T. Chan, K. Ho, A simple and efficient estimator for hyperbolic location, IEEE Trans. Signal Process. 42 (1994) 1905–1915.

[4] R. Schmidt, Least squares range difference location, IEEE Trans. Aerosp. Electron. Syst. 32 (1996) 234–242.

[5] A. Beck, P. Stoica, J. Li, Exact and approximate solutions of source localization problems, IEEE Trans. Signal Process. 56 (2008) 1770–1778.

[6] J. Velasco, D. Pizarro, J. Macias-Guarasa, A. Asaei, TDOA matrices: algebraic properties and their application to robust denoising with missing data, IEEE Trans. Signal Process. 64 (2016) 5242–5254.

[7] M.R. Bai, S.-S. Lan, J.-Y. Huang, Time difference of arrival (TDOA)-based acoustic source localization and signal extraction for intelligent audio classification, in: 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), IEEE, 2018, pp. 632–636.

[8] Y. Zhu, B. Deng, A. Jiang, X. Liu, Y. Tang, X. Yao, ADMM-based TDOA estimation, IEEE Commun. Lett. 22 (2018) 1406–1409.

[9] J.A. Apolinário, H. Yazdanpanah, A. Nascimento, M.L. de Campos, A data-selective LS solution to TDOA-based source localization, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 4400–4404.

[10] B. Friedlander, A passive localization algorithm and its accuracy analysis, IEEE J. Ocean. Eng. 12 (1987) 234–245.

[11] A. Amar, G. Leus, A reference-free time difference of arrival source localization using a passive sensor array, in: 2010 IEEE Sensor Array and Multichannel Signal Processing Workshop, IEEE, 2010, pp. 157–160.

[12] R. Jyothi, P. Babu, SOLVIT: a reference-free source localization technique using majorization minimization, IEEE/ACM Trans. Audio Speech Lang. Process. 28 (2020) 2661–2673.

[13] H.C. So, Y.T. Chan, F.K.W. Chan, Closed-form formulae for time-difference-of-arrival estimation, IEEE Trans. Signal Process. 56 (2008) 2614–2620.

[14] D.J. Torrieri, Statistical theory of passive location systems, IEEE Trans. Aerosp. Electron. Syst. (1984) 183–198.

[15] B. Yang, J. Scheuing, Cramér-Rao bound and optimum sensor array for source localization from time differences of arrival, in: 2005 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, IEEE, 2005, pp. 961–964.

[16] B.S. Mordukhovich, Variational Analysis and Generalized Differentiation I: Basic Theory, vol. 330, Springer Science & Business Media, 2006.

[17] Y. Zou, H. Liu, TDOA localization with unknown signal propagation speed and sensor position errors, IEEE Commun. Lett. 24 (2020) 1024–1027.

[18] K. Yang, G. Wang, Z.-Q. Luo, Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals, IEEE Trans. Signal Process. 57 (2009) 2775–2784.

[19] Y. Zou, H. Liu, Semidefinite programming methods for alleviating clock synchronization bias and sensor position errors in TDOA localization, IEEE Signal Process. Lett. 27 (2020) 241–245.

[20] G. Wang, K. Ho, Convex relaxation methods for unified near-field and far-field TDOA-based localization, IEEE Trans. Wirel. Commun. 18 (2019) 2346–2360.

[21] Y. Wang, K. Ho, TDOA positioning irrespective of source range, IEEE Trans. Signal Process. 65 (2016) 1447–1460.

[22] G. Wang, Y. Li, N. Ansari, A semidefinite relaxation method for source localization using TDOA and FDOA measurements, IEEE Trans. Veh. Technol. 62 (2012) 853–862.

[23] K. Yang, L. Jiang, Z.-Q. Luo, Efficient semidefinite relaxation for robust geolocation of unknown emitter by a satellite cluster using TDOA and FDOA measurements, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 2584–2587.

[24] F. Quo, K. Ho, A quadratic constraint solution method for TDOA and FDOA localization, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 2588–2591.

[25] X. Shi, B.D. Anderson, G. Mao, Z. Yang, J. Chen, Z. Lin, Robust localization using time difference of arrivals, IEEE Signal Process. Lett. 23 (2016) 1320–1324.

[26] W.H. Foy, Position-location solutions by Taylor-series estimation, IEEE Trans. Aerosp. Electron. Syst. (1976) 187–194.

[27] C. Mensing, S. Plass, Positioning algorithms for cellular networks using TDOA, in: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 4, IEEE, 2006, pp. IV–IV.

[28] K. Ho, W. Xu, An accurate algebraic solution for moving source location using TDOA and FDOA measurements, IEEE Trans. Signal Process. 52 (2004) 2453–2463.

[29] N. Piovesan, T. Erseghe, Cooperative localization in WSNs: a hybrid convex/non-convex solution, IEEE Trans. Signal Inf. Process. Netw. 4 (2018) 162–172.

[30] E. Gur, S. Sabach, S. Shtern, Alternating minimization based first-order method for the wireless sensor network localization problem, IEEE Trans. Signal Process. 68 (2020) 6418–6431.

[31] X. Qu, L. Xie, An efficient convex constrained weighted least squares source localization algorithm based on tdoa measurements, Signal Process. 119 (2016) 142–152.

[32] L. Lin, H.-C. So, F.K. Chan, Y.T. Chan, K. Ho, A new constrained weighted least squares algorithm for tdoa-based localization, Signal Process. 93 (2013) 2872–2878.

[33] G. Wang, A.M.-C. So, Y. Li, Robust convex approximation methods for TDOA-based localization under NLOS conditions, IEEE Trans. Signal Process. 64 (2016) 3281–3296.

[34] K.W. Cheung, H.-C. So, W.-K. Ma, Y.-T. Chan, A constrained least squares approach to mobile positioning: algorithms and optimality, EURASIP J. Adv. Signal Process. 2006 (2006) 1–23.

[35] Q. Li, B. Chen, M. Yang, Improved two-step constrained total least-squares TDOA localization algorithm based on the alternating direction method of multipliers, IEEE Sens. J. 20 (2020) 13666–13673.

[36] P. Stoica, J. Li, Lecture notes-source localization from range-difference measurements, IEEE Signal Process. Mag. 23 (2006) 63–66.

[37] N. Ono, S. Sagayama, R-means localization: a simple iterative algorithm for range-difference-based source localization, in: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2010, pp. 2718–2721.

[38] P. Wu, S. Su, Z. Zuo, X. Guo, B. Sun, X. Wen, Time difference of arrival (TDoA) localization combining weighted least squares and firefly algorithm, Sensors 19 (2019) 2554.

[39] M. Rosić, M. Simić, P. Pejović, Hybrid genetic optimization algorithm for target localization using tdoa measurements, in: Proceedings of the IcETRAN, 2017.

[40] E. Weiszfeld, Sur le point pour lequel la somme des distances de n points donnés est minimum, Tohoku Math. J., First Ser. 43 (1937) 355–386.

[41] A. Beck, S. Sabach, Weiszfeld's method: old and new results, J. Optim. Theory Appl. 164 (2015) 1–40.

[42] H.W. Kulin, R.E. Kuenne, An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics, J. Reg. Sci. 4 (1962) 21–33.

[43] A. Beck, Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB, vol. 19, SIAM, 2014.

[44] A. Beck, M. Teboulle, Z. Chikishev, Iterative minimization schemes for solving the single source localization problem, SIAM J. Optim. 19 (2008) 1397–1416.

[45] E. Gur, S. Sabach, S. Shtern, Convergent nested alternating minimization algorithms for nonconvex optimization problems, Math. Oper. Res. (2022).

[46] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009) 183–202.

[47] A. Beck, First-Order Methods in Optimization, vol. 25, SIAM, 2017.

[48] M. Sun, K. Ho, TDOA localization using closed-form solution, Computational Intelligence Signal Processing Laboratory, University of Missouri, Columbia, MO 65211, USA. Online: https://cisp.ece.missouri.edu/code.html, Oct. 2009, visited Nov. 30, 2022.

[49] L. Lin, H.-C. So, F.K. Chan, Y.T. Chan, K. Ho, So, Hing Cheung, Department of Electrical Engineering, City University of Hong, Kong. Online: https://www.ee.cityu.edu.hk/~hcso/publication.html, Nov. 2013, visited Jan. 22, 2023.

[50] D. Peaucelle, D. Henrion, Y. Labit, K. Taitz, User's Guide for SeDuMi Interface 1.04, LAAS-CNRS, Toulouse, 2002.

[51] L. Malagò, G. Pistone, Information geometry of the Gaussian distribution in view of stochastic optimization, in: Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, 2015, pp. 150–162.

[52] Y. Xie, J. Lee, T. Song, Analysis for reference sensor selection in time difference of arrival-based localisation, Electron. Lett. 54 (2018) 1454–1456.

[53] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 2012.

[54] J.-P. Penot, Calculus Without Derivatives, vol. 266, Springer Science & Business Media, 2012.

Eyal Gur received the B.Sc. degree in mathematics and economics from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2016 and the M.Sc. degree in applied mathematics (cum laude) from the Technion - Israel Institute of Technology, Haifa, Israel, in 2019. He is currently working as Ph.D. candidate researcher with the Faculty of Data and Decision Sciences in the Technion. His current research interest is non-convex optimization and its applications in engineering.

Alon Amar received the B.Sc. degree in electrical engineering from the Technion - Israel Institute of Technology, Haifa, Israel, in 1997 and the M.Sc. degrees in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 2003, and 2009, respectively. From 2009 to 2010, he was a Postdoctoral Research Associate with the Circuits and Systems group, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft, The Netherlands. In 2011, he joined the Israeli National Research Center, Haifa, as a Research Scientist. Since 2016, he is an Adjunct Instructor at the Faculty of Electrical and Computer Engineering, Technion - Israel Institute of Technology, Haifa, Israel. His main research interests include statistical and array signal processing, wireless communication, and sensor networks.

Shoham Sabach received the B.Sc. degree in mathematics (summa cum laude) and the M.Sc. degree in mathematics (magna cum laude) from University of Haifa, Haifa, Israel, in 2004 and 2008, respectively, and the Ph.D. degree in mathematics from the Technion - Israel Institute of Technology, Haifa, Israel, in 2012. From October 2012 to September 2013, he was a Postdoctoral Fellow with the School of Mathematical Sciences of Tel-Aviv University, Tel-Aviv, Israel. From October 2013 to September 2014, he was a Humboldt Postdoctoral Research Fellow with the Institute for Numerical and Applied Mathematics, University Göttingen, Göttingen, Germany. He is currently an Assistant Professor with the Faculty of Industrial Engineering and Management at the Technion-Israel Institute of Technology, Haifa, Israel. He has given invited lectures at international conferences, and was awarded the Cooper Award for Excellence in Research and the SIAM prize for the most outstanding paper on a topic in optimization published between 2013 and 2017. His research interests include the area of continuous optimization, including theory, algorithmic analysis, and its applications.