A Dual-Based First-Order Algorithm for ToA Asynchronous Localization and Synchronization

Eyal Gur[*†‡] Alon Amar[§] Shoham Sabach[†‡]

March 5, 2024

Abstract

Joint ToA source localization and synchronization determines the location and time offset of a radiating source using time-of-arrival measurements collected from a time-synchronized array of sensors. Various approaches have been proposed to address this non-convex and non-smooth optimization problem, which usually transform the problem by applying convex relaxations or smooth approximations. In this paper, we focus on the original joint problem and show that it can be expressed as a sum of a quadratic function with multiple non-smooth functions. This type of problems cannot be solved using traditional proximal-based methods, and we develop a tailored dual-based first-order algorithm. We analyze the proposed method, and prove its convergence to critical points of the original problem under mild assumptions. Experimental results showcase advantages of the method in terms of convergence, RMSE, bias, and complexity.

Keywords: Time-of-Arrival; joint source localization and synchronization; non-convex optimization; global convergence; fast dual proximal gradient

1 Introduction

Classical Time-of-Arrival (ToA) Source Localization localization involves emitting a signal from the source, which is then received by an array of time-synchronized sensors. Each of these sensors logs the precise time of signal arrival, and these measurements are utilized to estimate the unknown location of the source [30]. In this paper, we focus on the more challenging scenario of ToA localization where the source is *not necessarily* time-synchronized with the array, and an unknown time offset is present [32, 36].

[*]Corresponding author. Mail: eyal.gur@campus.technion.ac.il.
[†]Faculty of Data and Decision Sciences, Technion – Israel Institute of Technology, Haifa 3200003, Israel.
[‡]The work of Eyal Gur and Shoham Sabach was supported by the Israel Science Foundation, ISF 2480-21.
[§]Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology, Haifa 3200003, Israel.

To better explain the differences between various types of ToA localization, we define the problem mathematically. This task involves an array of $N \geq 2$ sensors, each located at a known location $\mathbf{p}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$ (typically, $n = 2$ or $n = 3$), and a source at an unknown location $\mathbf{s} \in \mathbb{R}^n$. We denote by $T \in \mathbb{R}$ the unknown time offset between the source and sensors due to lack of synchronization. The measured travel time of the signal from the source to sensor i, denoted by $t_i \in \mathbb{R}$, is modeled as [36]

$$t_i = \frac{1}{c} \|\mathbf{s} - \mathbf{p}_i\| + T + \epsilon_i, \quad \forall i = 1, 2, \ldots, N, \tag{1}$$

where $c > 0$ is the constant known propagation speed of the transmitted signal, and $\epsilon_i \in \mathbb{R}$ unknown noise.

The synchronous ToA localization problem, where $T = 0$ and the focus lies solely on determining the unknown location $\mathbf{s} \in \mathbb{R}^n$, has been extensively explored [4, 7, 21, 29, 35, 36]. It is regarded as a specific case within the broader context of the asynchronous problem. Thus, in our study, we operate under the assumption that the time offset $T \in \mathbb{R}$ is unknown. In this asynchronous scenario, two primary strategies are typically employed: one strategy, called Time-Difference-of-Arrival (TDoA), eliminates the unknown time offset $T \in \mathbb{R}$ from the set of equations in (1) through their pairwise subtraction [8]. If there exists a central processor within the array that is equipped with the ability to gather ToA measurements from all sensors, then the elimination of T can be achieved and localization of the source can be accomplished. More details and up-to-date advancements on TDoA can be found in our recent paper [13] (see also [18, 41]). However, the TDoA technique requires a central processor and is also known to amplify noises [36]. Due to these limitations of TDoA, in this paper we do not focus on this approach.

The other strategy, which is the focus of this paper, involves finding a solution to the set of equations in (1) that determines *both* the unknown location and the unknown time offset. This strategy, which is considered the most general ToA approach, is also known as the *joint localization and synchronization* problem [3, 17, 19, 31, 39], as it simultaneously localizes the source and synchronizes it with the array.

Current methods addressing this joint problem typically involve two main techniques. One technique modifies the problem by considering convex relaxations of the objective function, followed by the application of commercial interior-point solvers. Another technique modifies the problem by squaring the original ToA measurements in (1), thereby transforming the non-smooth (i.e., non-differentiable) objective into a smooth one. This transformation allows for the application of standard linear least-squares methods. A detailed survey of these approaches is presented in Section 5.

Motivated by our paper [13], and in order to develop a novel algorithm for the joint problem – in this paper we will initially construct an upper bound for the objective function of the joint problem. However, as we demonstrate below, in a sharp contrast to [13], this bound comprises a sum of a quadratic function

with *multiple* (rather than just one as in [13]) non-smooth functions, each represents the Euclidean distance between the source and a sensor. This difference poses a major obstacle from the optimization point of view. Indeed, it is well-known that the proximal computation of the sum of multiple non-smooth functions is notoriously challenging, leading to the prevalence of splitting methods in optimization. In this paper, by leveraging the distinctive structure of the non-smooth functions, we are able to devise an algorithm to address the original joint asynchronous ToA problem. This algorithm utilizes first-order information and leverages the notion of duality to overcome the complexity posed by the sum of multiple non-smooth functions.

The key contributions of our proposed iterative algorithm, which is called Fast Dual Localization and Synchronization (FDLS), are summarized now.

1. FDLS is a simple iterative solver, that in contrast to existing approaches, is designed to handle both high and low levels of noise.

2. In contrast to interior-point solvers, FDLS solely relies on first-order information. As a result, it enjoys low computational and storage requirements.

3. FDLS is the first algorithm with convergence guarantees. In Section 4, we prove its convergence to first-order optimal solutions (i.e., critical points) of the original non-convex and non-smooth joint asynchronous ToA problem. This is a stronger result than function value or sub-sequence convergence.

In Section 2, we formulate the associated optimization problem. In Section 3, we develop our FDLS algorithm while in Section 4 we prove its convergence. In Section 5 we survey existing works and in Section 6 we conduct numerical experiments.

2 Problem Formulation

Throughout this paper, we assume that $\mathbf{p}_i \neq \mathbf{p}_j$ for all $i \neq j$, and that the locations of the sensors are not co-linear (if $N = 2$, then the sensors and source are not co-linear). We assume independent and Gaussian-distributed noises, and without the loss of generality that $c = 1$ [17].

The set of equations (1) can be translated to an optimization problems using the maximum-likelihood (ML) approach, which results in a task that requires to jointly determining $\mathbf{s} \in \mathbb{R}^n$ and $T \in \mathbb{R}$ [36]

$$\min_{\mathbf{s}\in\mathbb{R}^n, T\in\mathbb{R}} \quad F(\mathbf{s}, T) \equiv \frac{1}{2}\sum_{i=1}^{N} \left(\|\mathbf{s} - \mathbf{p}_i\| + T - t_i\right)^2. \tag{2}$$

This problem is non-convex and non-smooth, therefore a closed-form formula for its optimal solution cannot be derived. Hence, as criticality of a point serves as a necessary condition for its optimality, we focus

on obtaining critical points $(\mathbf{s}, T) \in \mathbb{R}^n \times \mathbb{R}$ of the objective function F of Problem (2). By critical point, we mean a point where its limiting sub-differential set, denoted as ∂, includes the zero vector [24].

Since F is continuously differentiable with respect to $T \in \mathbb{R}$, then its gradient with respect to T, denoted by $\nabla_T F$, is well-defined. Hence, from the requirement that $\nabla_T F(\mathbf{s}, T) = 0$ we get that any critical point $(\mathbf{s}, T) \in \mathbb{R}^n \times \mathbb{R}$ of F satisfies

$$T = \frac{1}{N} \sum_{i=1}^{N} (t_i - \|\mathbf{s} - \mathbf{p}_i\|) \equiv \mathcal{T}(\mathbf{s}). \tag{3}$$

Plugging (3) into (2) yields, after simple algebraic manipulations, that the objective $F(\mathbf{s}, T)$ coincides over its set of critical points with the function $\mathcal{F}\colon \mathbb{R}^n \to \mathbb{R}$ defined as

$$\mathcal{F}(\mathbf{s}) = \frac{N-1}{2} \|\mathbf{s}\|^2 + \mathbf{p}^T \mathbf{s} + \sum_{i=1}^{N} \left(\frac{t}{N} - t_i \right) \|\mathbf{s} - \mathbf{p}_i\| - \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|\mathbf{s} - \mathbf{p}_i\| \cdot \|\mathbf{s} - \mathbf{p}_j\| + C, \tag{4}$$

where we define

$$\mathbf{p} \equiv \left(\frac{1}{N} - 1 \right) \sum_{i=1}^{N} \mathbf{p}_i \in \mathbb{R}^n, \quad t \equiv \sum_{i=1}^{N} t_i > 0,$$

and $C \in \mathbb{R}$ is a constant independent of $\mathbf{s} \in \mathbb{R}^n$ and $T \in \mathbb{R}$.

To minimize $\mathcal{F}(\mathbf{s})$, and motivated by the approach in [13], we first find an upper bound of this function. To this end, for any set of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ and $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_N$ that reside in the unit ball $\mathcal{B} \subset \mathbb{R}^n$ centered at $\mathbf{0}_n$ with radius 1, we define the symmetric matrix

$$\mathbf{A} \equiv \frac{N-1}{2} \mathbf{I}_n + \frac{1}{2N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\mathbf{u}_i \mathbf{v}_j^T + \mathbf{v}_j \mathbf{u}_i^T \right), \tag{5}$$

with $\mathbf{I}_n$ the $n \times n$ identity matrix. We also define the vector

$$\mathbf{z} \equiv -\mathbf{p} + \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\mathbf{p}_j^T \mathbf{v}_j \mathbf{u}_i + \mathbf{p}_i^T \mathbf{u}_i \mathbf{v}_j \right) - \sum_{i=1}^{N} t_i \mathbf{u}_i. \tag{6}$$

Now, we define the smooth function φ as

$$\varphi(\mathbf{s}, \mathbf{u}, \mathbf{v}) \equiv \mathbf{s}^T \mathbf{A} \mathbf{s} - \mathbf{z}^T \mathbf{s}, \tag{7}$$

which is quadratic with respect to $\mathbf{s} \in \mathbb{R}^n$ for any given $\mathbf{u} \equiv (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$ and $\mathbf{v} \equiv (\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_N)$. We also define the non-smooth function ψ as

$$\psi(\mathbf{s}) \equiv \frac{t}{N} \sum_{i=1}^{N} \|\mathbf{s} - \mathbf{p}_i\|. \tag{8}$$

Simple calculations show that

$$\mathcal{F}(\mathbf{s}) \leq \Psi(\mathbf{s},\mathbf{u},\mathbf{v}) \equiv \varphi(\mathbf{s},\mathbf{u},\mathbf{v}) + \psi(\mathbf{s}) + C, \tag{9}$$

for any $(\mathbf{s},\mathbf{u},\mathbf{v})$, and equality is attained by taking the unit sphere vectors $\mathbf{u}_i = \bar{\mathbf{u}}_i(\mathbf{s})$ and $\mathbf{v}_j = \bar{\mathbf{v}}_j(\mathbf{s})$ defined as

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

and

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

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

Summarizing the derivations above, the minimization in Problem (2) is equivalent to the minimization

$$\min_{\mathbf{s}\in\mathbb{R}^n} \{\mathcal{F}(\mathbf{s}) = \Psi(\mathbf{s},\bar{\mathbf{u}},\bar{\mathbf{v}}) = \varphi(\mathbf{s},\bar{\mathbf{u}},\bar{\mathbf{v}}) + \psi(\mathbf{s})\}. \tag{12}$$

As the two Problems (2) and (12) are equivalent, we use them below interchangeably.

We mention that the function ψ in (8) exhibits an intricate structure due to the presence of a sum of *multiple* non-smooth functions. As mentioned earlier, this poses a significant challenge when employing traditional optimization tools, such as computing the proximal mapping of the sum of functions. This task is generally intractable, even when each individual function possesses a closed-form proximal mapping. The complexity of this challenge prevents us from relying on the advancements in [13]. Instead, it motivated us to devise a tailored algorithmic approach that overcome the subtle structure of the non-smooth function ψ.

Before concluding this section, it is worth noting that the derived upper bound in (9) can be applied to other potential optimization problems characterized by the minimization of objective functions structured as $\mathcal{F}(\mathbf{s})$. This structure involves a sum of a quadratic and smooth function, accompanied by a sum of non-smooth norm terms. The algorithm FDLS, which will be introduced in the subsequent section, can be adapted to address these potential optimization problems.

3 The FDLS Algorithm

Here we present our algorithm Fast Dual Localization and Synchronization (FDLS), which directly addresses the non-convex and non-smooth Problem (2). In Section 4, we prove that the generated sequence of locations

and time offsets converges to critical points of Problem (2).

To accomplish this, we address the minimization of $F(\mathbf{s}, T)$ in Problem (2) through the minimization of its equivalent function $\Psi(\mathbf{s}, \mathbf{u}, \mathbf{v})$ in (12). By focusing on Ψ, we are able to develop FDLS by leveraging the Nested Alternating Minimization (NAM) scheme of [14]. This scheme breaks down an optimization problem into smaller sub-problems and minimizes them sequentially using a chosen nested sub-algorithm.

Indeed, minimizing Ψ can be carried out in such alternating fashion: first, update all vectors $\bar{\mathbf{u}}_i$ and $\bar{\mathbf{v}}_j$, and then minimize Ψ with respect to $\mathbf{s} \in \mathbb{R}^n$ using a nested sub-algorithm, and repeat the process iteratively.

As already discussed above, given the intricate nature of Ψ, its minimization with respect to $\mathbf{s} \in \mathbb{R}^n$ presents a challenging structure for minimization, due to the presence of the sum of multiple non-smooth norm terms. However, the dual of the minimization of Ψ with respect to $\mathbf{s} \in \mathbb{R}^n$ can be derived easily, and the dual exhibits a structure that can be minimized using the well-known Accelerated Projected Gradient (APG) method of Nesterov [25].

In summary, the FDLS algorithm obeys the following alternating and iterative optimization scheme:

1. Update the vectors $\mathbf{u}$ and $\mathbf{v}$ using (10) and (11).

2. Update $\mathbf{s} \in \mathbb{R}^n$ by applying iterations of APG to minimize the dual of Ψ with respect to $\mathbf{s}$.

The exact update steps of FDLS are given below in Algorithm 1, and are developed now.

Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a sequence of iterations generated by FDLS starting with some random initial point $\mathbf{s}^0 \in \mathbb{R}^n$. For any iteration $k \geq 0$, let $\bar{\mathbf{u}}^k \equiv \left(\bar{\mathbf{u}}_1^k, \bar{\mathbf{u}}_2^k, \ldots, \bar{\mathbf{u}}_N^k\right)$ and $\bar{\mathbf{v}}^k \equiv \left(\bar{\mathbf{v}}_2^k, \bar{\mathbf{v}}_3^k, \ldots, \bar{\mathbf{v}}_N^k\right)$ be $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ evaluated at $\mathbf{s}^k$, as defined in (10) and (11), respectively. In addition, let $\mathbf{A}^k \in \mathbb{R}^{n\times n}$ and $\mathbf{z}^k \in \mathbb{R}^n$ be the matrix $\mathbf{A}$ and vector $\mathbf{z}$ (see (5) and (6), respectively) evaluated at $\bar{\mathbf{u}}^k$ and $\bar{\mathbf{v}}^k$. Therefore, minimizing $\Psi\left(\mathbf{s}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$ with respect to $\mathbf{s} \in \mathbb{R}^n$ (see (12)) at the k-th iteration takes the form

$$\begin{aligned} \min_{\mathbf{s},\mathbf{q}} \quad & \mathbf{s}^T\mathbf{A}^k\mathbf{s} - \left(\mathbf{z}^k\right)^T\mathbf{s} + \frac{t}{N}\sum_{i=1}^{N}\|\mathbf{q}_i\| \\ \text{s.t.} \quad & \mathbf{q}_i = \mathbf{s} - \mathbf{p}_i, \quad \forall i = 1, 2, \ldots, N. \end{aligned} \tag{13}$$

3.1 Finding the Dual Problem

Minimizing the problem in (13) is challenging due to the sum of norm terms. Therefore, we find its dual problem. Denoting by $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_N \in \mathbb{R}^n$ the dual variables, the Lagrangian of (13) with respect to the primal variables $\mathbf{s}, \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N \in \mathbb{R}^n$ is given as

$$\mathcal{L}(\mathbf{s}, \mathbf{q}, \boldsymbol{\zeta}) = \mathbf{s}^T\mathbf{A}^k\mathbf{s} - \left(\mathbf{z}^k + \sum_{i=1}^{N}\boldsymbol{\zeta}_i\right)^T\mathbf{s} + \sum_{i=1}^{N}\left(\frac{t}{N}\|\mathbf{q}_i\| + \boldsymbol{\zeta}_i^T\mathbf{q}_i\right) + \sum_{i=1}^{N}\boldsymbol{\zeta}_i^T\mathbf{p}_i.$$

To arrive at a dual problem, we need to minimize $\mathcal{L}$ with respect to the primal variables $\mathbf{s}, \mathbf{q}_1, \mathbf{w}_2, \ldots, \mathbf{q}_N \in \mathbb{R}^n$. Notice that $\mathcal{L}$ is quadratic with respect to $\mathbf{s} \in \mathbb{R}^n$, and its solution depends on the matrix $\mathbf{A}^k$ being positive-definite (PD). Indeed, in Appendix A we prove the following lemma which ensures that this PD property holds.

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

Remark 1. Lemma 1 necessitates the boundedness of the sequence. Due to the coupling of $\mathbf{s} \in \mathbb{R}^n$ and $T \in \mathbb{R}$ in Problem (2), which renders it non-coercive, the boundedness assumption may not hold universally. However, the failure of this assumption is only expected if the sequence of locations diverges to infinity.

Following Lemma 1, we immediately get that minimizing the Lagrangian with respect to $\mathbf{s} \in \mathbb{R}^n$ yields $\min_{\mathbf{s}\in\mathbb{R}^n} \mathcal{L}\left(\mathbf{s}, \mathbf{p}, \boldsymbol{\zeta}\right) = -\tilde{\varphi}^k\left(\boldsymbol{\zeta}\right)$, for $\tilde{\varphi}^k$ a convex and quadratic function in $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_N \in \mathbb{R}^n$, defined as

$$\tilde{\varphi}^k\left(\boldsymbol{\zeta}\right) = \frac{1}{4}\left(\mathbf{z}^k + \sum_{i=1}^{N}\boldsymbol{\zeta}_i\right)^T \left(\mathbf{A}^k\right)^{-1}\left(\mathbf{z}^k + \sum_{i=1}^{N}\boldsymbol{\zeta}_i\right) - \sum_{i=1}^{N}\boldsymbol{\zeta}_i^T\mathbf{p}_i, \tag{14}$$

and the primal-dual relation is given by

$$\mathbf{s}\left(\boldsymbol{\zeta}\right) = \frac{1}{2}\left(\mathbf{A}^k\right)^{-1}\left(\mathbf{z}^k + \sum_{i=1}^{N}\boldsymbol{\zeta}_i\right). \tag{15}$$

In addition, minimizing the Lagrangian with respect to each primal variable $\mathbf{q}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$, yields

$$\min_{\mathbf{q}_i\in\mathbb{R}^n} \mathcal{L}\left(\mathbf{s}, \mathbf{q}, \boldsymbol{\zeta}\right) = \begin{cases} 0, & \|\boldsymbol{\zeta}_i\| \leq \frac{t}{N}, \\ -\infty, & \|\boldsymbol{\zeta}_i\| > \frac{t}{N} \end{cases} = -\mathcal{I}\left(\boldsymbol{\zeta}_i\right),$$

where $\mathcal{I}\colon \mathbb{R}^n \to \{0, \infty\}$ is the indicator function of the ball with radius t/N centered at $\mathbf{0}_n$.

Summarizing the above derivations, for any iteration $k \geq 0$, the dual of (13) is given explicitly as

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^{nN}} \tilde{\Psi}^k\left(\boldsymbol{\zeta}\right) \equiv \tilde{\varphi}^k\left(\boldsymbol{\zeta}\right) + \sum_{i=1}^{N}\mathcal{I}\left(\boldsymbol{\zeta}_i\right). \tag{16}$$

Inspecting the dual problem in (16), we see that for any iteration $k \geq 0$, it comprises of a sum of a convex and quadratic smooth function $\tilde{\varphi}^k$ (see (14)), and separable ball constraints. Hence, this dual can be minimized by applying iterations of the Accelerated Projected Gradient (APG) method of Nesterov [25]. APG is an accelerated variant of the proximal gradient method, and it generates a sequence of iterations along with an auxiliary sequence. Next, we find explicit APG update steps at any iteration $k \geq 0$.

3.2 Deriving Explicit FDLS Updates

For some iteration $k \geq 0$ of FDLS, we apply $m_k \in \mathbb{N}$ iterations of APG to minimize the dual in (16). We denote by $\mathbf{y}^m \in \mathbb{R}^{nN}$, $m = 0, 1, 2 \ldots, m_k$, the sequence generated by APG with starting point $\mathbf{y}^0 = \boldsymbol{\zeta}^k$. The output vector $\boldsymbol{\zeta}^{k+1}$ is defined as $\boldsymbol{\zeta}^{k+1} = \mathbf{y}^{m_k} \in \mathbb{R}^{nN}$. In addition, we denote by $\boldsymbol{\xi}^m \in \mathbb{R}^{nN}$, $m = 0, 1, 2 \ldots, m_k$, the auxiliary sequence generated by APG.

Following [5], at each iteration $m \geq 0$ of APG to minimize the dual $\tilde{\Psi}^k$ in (16), the next iteration $\mathbf{y}^{m+1}$ is updated by calculating the projected gradient step

$$\mathbf{y}^{m+1} = \underset{\mathbf{y} \in \mathbb{R}^{nN}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \mathcal{I}\left(\mathbf{y}_i\right) + \frac{L^k}{2} \left\| \mathbf{y} - \mathbf{c}^{k,m} \right\|^2 \right\}, \tag{17}$$

where $\mathbf{c}^{k,m} \equiv \boldsymbol{\xi}^m - \frac{1}{L^k} \nabla \tilde{\varphi}^k \left(\boldsymbol{\xi}^m\right)$, and L^k is the Lipschitz constant of the gradient of $\tilde{\varphi}^k$ given as (see [1])

$$L^k = \frac{N}{2\lambda_{\min}\left(\mathbf{A}^k\right)}. \tag{18}$$

From separability of (17) with respect to the variables $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N \in \mathbb{R}^n$, it follows for any $i = 1, 2, \ldots, N$ that

$$\mathbf{y}_i^{m+1} = \underset{\mathbf{y}_i \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \mathcal{I}\left(\mathbf{y}_i\right) + \frac{L^k}{2} \left\| \mathbf{y}_i - \mathbf{c}_i^{k,m} \right\|^2 \right\}, \tag{19}$$

where

$$\mathbf{c}_i^{k,m} = \boldsymbol{\xi}_i^m - \frac{1}{L^k} \nabla_i \tilde{\varphi}^k \left(\boldsymbol{\xi}^m\right) = \boldsymbol{\xi}_i^m - \frac{1}{L^k} \left(\mathbf{s}\left(\boldsymbol{\xi}^m\right) - \mathbf{p}_i\right) \tag{20}$$

for $\nabla_i \tilde{\varphi}^k$ the gradient of $\tilde{\varphi}^k$ with respect to $\mathbf{y}_i \in \mathbb{R}^n$, and where the equality follows from (14) and (15).

Finally, the update in (19) is explicitly given for any iteration $k \geq 0$ of FDLS as (see [2, Theorem 6.24])

$$\mathbf{y}_i^{m+1} = \frac{t \cdot \mathbf{c}_i^{k,m}}{\max\left\{ N \cdot \left\| \mathbf{c}_i^{k,m} \right\|, t \right\}}. \tag{21}$$

Summarizing the above derivations, our proposed FDLS algorithm is outlined in Algorithm 1.

Remark 2. (i) The starting point $\mathbf{s}^0 \in \mathbb{R}^n$ of FDLS can be set arbitrarily, and $\boldsymbol{\zeta}^0 \in \mathbb{R}^{nN}$ is set such that its first n coordinates are $\mathbf{s}^0$, and all other coordinates are 0 (though any other initial $\boldsymbol{\zeta}^0$ is acceptable).

(ii) For any $k \geq 0$, the number of APG iterations, denoted m_k in step 6 of Algorithm 1, can be set

Algorithm 1 FDLS (Fast Dual Localization & Synchronization)

1: **Initialization:** Set any $\mathbf{s}^0 \in \mathbb{R}^n$ and $\boldsymbol{\zeta}^0 \in \mathbb{R}^{nN}$.
2: **for** $k \geq 0$ **do**
3: For $i = 1, \ldots, N$ and $j = 2, \ldots, N$, update $\bar{\mathbf{u}}_i^k \equiv \mathbf{u}_i\left(\mathbf{s}^k\right)$ and $\bar{\mathbf{v}}_j^k \equiv \mathbf{v}_j\left(\mathbf{s}^k\right)$ (see (10), (11)).
4: Update $\mathbf{A}^k$, $\mathbf{z}^k$ and L^k (see (5), (6), (18)), by plugging $\mathbf{u} = \bar{\mathbf{u}}^k$ and $\mathbf{v} = \bar{\mathbf{v}}^k$.
5: Set $\mathbf{y}^0 = \boldsymbol{\xi}^0 = \boldsymbol{\zeta}^k$ and $\alpha_0 = 1$.
6: **for** $m = 0, 1, \ldots, m_k - 1$ **do**
7: For all $i = 1, \ldots, N$, update $\mathbf{y}_i^{m+1}$ (see (20), (21)).
8: Set $\alpha_{m+1} = \frac{1+\sqrt{1+4\alpha_m^2}}{2}$.
9: Set $\boldsymbol{\xi}^{m+1} = \mathbf{y}^{m+1} + \frac{\alpha_m - 1}{\alpha_{m+1}}\left(\mathbf{y}^{m+1} - \mathbf{y}^m\right)$.
10: **end for**
11: Set $\boldsymbol{\zeta}^{k+1} = \mathbf{y}^{m_k}$ and $\mathbf{s}^{k+1} = \mathbf{s}^k\left(\boldsymbol{\zeta}^{k+1}\right)$ (see (15)).
12: **end for**
13: Output location $\mathbf{s}^{k+1}$ and time offset $\mathcal{T}\left(\mathbf{s}^{k+1}\right)$ (see (3)).

arbitrarily. Following [14, Section 4.2.1], to obtain convergence results of FDLS in Section 4, we set

$$m_k = s + 2^{\lfloor k/r \rfloor} - 1, \tag{22}$$

for some integers $s, r \in \mathbb{N}$. This rule is quite general since there is no limitation in choosing s and r.

(iii) The updates of the auxiliary variables $\bar{\mathbf{u}}_i^k$ and $\bar{\mathbf{v}}_j^k$ in step 3 of FDLS, have a simple and computationally inexpensive closed-form formula that depends only on the previous iteration $\mathbf{s}^k$. Therefore, these vectors can be removed by substituting them in the update of $\mathbf{s}^{k+1}$, and FDLS can be written only in terms of the original variable $\mathbf{s} \in \mathbb{R}^n$ and its dual variables.

(iv) Following [13, Lemma 1], one can also set

$$L^k = \frac{N^2}{N^2 - N + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \min\left\{0, \left(\bar{\mathbf{u}}_i^k\right)^T \bar{\mathbf{v}}_j^k\right\}},$$

which is a distributed constant that does not require the calculation of the eigenvalues of $\mathbf{A}^k$.

4 Convergence Analysis of FDLS

In this section, we prove that FDLS converges to critical points (that is, first-order optimal solutions) of the original non-convex and non-smooth function F of Problem (2). This proof utilizes the methodology outlined in [14] and later extended in [15], which imposes three mathematical properties on the generated sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$: (i) a decrease in function value up to a certain error, (ii) a sub-gradient bound by the iterate gap up to a certain error, and (iii) summable errors. It was proved in [14] that when these properties

are satisfied, the sequence converges to critical points of the function under consideration. For FDLS, we state this result in Theorem 1.

Theorem 1. *Let* $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ *be a bounded sequence generated by FDLS. Then, it converges to some* $\mathbf{s}^* \in \mathbb{R}^n$ *such that* $(\mathbf{s}^*, \mathcal{T}(\mathbf{s}^*))$ *is a critical point of* F *of Problem* (2).

To prove Theorem 1, it is imperative to show that FDLS indeed satisfies the three mathematical properties mentioned above, which are stated in Propositions 1 and 2. The detailed proofs for these propositions are provided in Appendix B.

Proposition 1 (Function value decrease up to error)**.** *Let* $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ *be a bounded sequence generated by FDLS. Then, for any* $k \geq 0$ *there exists* $m_k \in \mathbb{N}$ *such that*

$$\Psi\left(\mathbf{s}^k, \mathbf{u}^k, \mathbf{v}^k\right) - \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}, \mathbf{v}^{k+1}\right) \geq \max\left\{\beta\left\|\mathbf{s}^k - \mathbf{s}^{k+1}\right\|^2 - e_1^k, 0\right\},$$

where $e_1^k \equiv M_1/m_k \geq 0$ *is an error term for some constant* $M_1 > 0$ *independent of* $k \geq 0$.

Proposition 2 (Sub-gradient bound up to error)**.** *Let* $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ *be a bounded sequence generated by FDLS. Then, for any* $k \geq 0$ *there is* $\mathbf{w}^{k+1} \in \partial_{\mathbf{s}}\Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^k, \mathbf{v}^k\right)$ *satisfying*

$$\left\|\mathbf{w}^{k+1}\right\| \leq e_2^k, \tag{23}$$

where $e_2^k \equiv M_2/m_k \geq 0$ *is an error term for some constant* $M_2 > 0$ *independent of* $k \geq 0$.

With these two propositions, we now prove Theorem 1, which is the main theoretical result of this paper.

Proof of Theorem 1. Following [14, Theorem 1], based on Propositions 1 and 2, we need to show that the error sequences $\sqrt{e_1^k}$ and e_2^k, $k \geq 0$, 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 $(\mathbf{u}^*, \mathbf{v}^*)$ is a limit point of $\left\{\left(\bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)\right\}_{k\geq 0}$, then $(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*)$ is a critical point of Ψ. To this end, notice that

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

Hence, by the definition of e_1^k in Proposition 1 we establish that the sequence $\sqrt{e_1^k}$, $k \geq 0$, is summable. Similarly, the sequence e_2^k, $k \geq 0$, is also summable by the definition of e_2^k in Proposition 2.

Now, we are left to show that for any $\mathbf{s}^* \in \mathbb{R}^n$ obtained by FDLS, then $(\mathbf{s}^*, \mathcal{T}(\mathbf{s}^*))$ is a critical point of Problem (2). As discussed in Section 2, it is enough to prove that $\mathbf{s}^*$ is a critical point of $\mathcal{F}(\mathbf{s}) \equiv F(\mathbf{s}, \mathcal{T}(\mathbf{s}))$ (see (4)).

Let $(\mathbf{u}^*, \mathbf{v}^*)$ be a limit point of the bounded sequence $\left\{\left(\bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)\right\}_{k\geq 0}$ (following (10) and (11), this sequence is bounded since its elements are all in the unit sphere). By possibly passing to a sub-sequence, we assume that $\left(\bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \to (\mathbf{u}^*, \mathbf{v}^*)$ as $k \to \infty$.

Following Assumption 1 (see Appendix B), there exits $K \geq 0$ such that $\mathbf{s}^k \neq \mathbf{p}_i$ for all $i = 1, 2, \ldots, N$ and for all $k \geq K$. By the definition of $\left(\bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$ (see step 3 in Algorithm 1) we get for any $i = 1, 2, \ldots, N$, $j = 2, 3 \ldots, N$ and $k \geq K$ that

$$\bar{\mathbf{u}}_i^{k+1} = -\frac{\mathbf{s}^k - \mathbf{p}_i}{\|\mathbf{s}^k - \mathbf{p}_i\|} \quad \text{and} \quad \bar{\mathbf{v}}_j^{k+1} = \frac{\mathbf{s}^k - \mathbf{p}_j}{\|\mathbf{s}^k - \mathbf{p}_j\|}.$$

By taking the limit $k \to \infty$ in the above equalities we get

$$\mathbf{u}_i^* = -\frac{\mathbf{s}^* - \mathbf{p}_i}{\|\mathbf{s}^* - \mathbf{p}_i\|} \quad \text{and} \quad \mathbf{v}_j^* = \frac{\mathbf{s}^* - \mathbf{p}_j}{\|\mathbf{s}^* - \mathbf{p}_j\|}. \tag{24}$$

Finally, simple calculations show that $\partial_{\mathbf{s}} \Psi\left(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*\right) = \partial \mathcal{F}\left(\mathbf{s}^*\right)$, where $(\mathbf{u}^*, \mathbf{v}^*)$ is defined in (24). Since $(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*)$ is a critical point of Ψ, then $\mathbf{0}_n \in \partial_{\mathbf{s}} \Psi\left(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*\right) = \partial \mathcal{F}\left(\mathbf{s}^*\right)$, which completes the proof. □

5 Existing Work

Approaches for the joint problem in (2) can be classified into three main categories, as specified now.

Convex relaxations. In [36], the authors propose the 2LS algorithm to address Problem (2). The algorithm begins by estimating the unknown time offset T, and this estimate replaces the unknown T in the objective function. To further simplify the problem, auxiliary variables $\tau_i = \|\mathbf{s} - \mathbf{p}_i\|$ are introduced. By employing the relaxation technique of [38], the problem is transformed into a convex problem, which is then solved using standard Semi-Definite Programming (SDP) solvers that utilize interior-point methods. The works [20, 23, 28, 32, 37, 42] also apply various relaxation procedures to obtain a convex relaxed problem that is then solved using SDP solvers.

Although convex relaxations have a global minimum and lack locally optimal solutions, there is no assurance that this optimal solution corresponds to a critical point of the original ML problem [27]. In addition, solving convex relaxations using SDP solvers results in an increased computational complexity, which limits their practicality.

Smooth approximations. Achieved by taking the square on both sides of the original set of ToA measurements in (1) to obtain smooth equations. However, squaring introduces additional dependency of the

noises on both the original noises and the unknown distances. Consequently, the squaring technique only approximates the ML solution when the noise levels are sufficiently small [8].

In [36], the set of N equations $(t_i - T)^2 - \frac{1}{c^2}\|\mathbf{s} - \mathbf{p}_i\|^2 = \nu_i$ is considered, where $\nu_i \in \mathbb{R}$ represents a noise term that depends on the original noise ϵ_i and on the true distance between the source and sensor i. Then, the ℓ_∞ norm of the noise terms $\nu_1, \nu_2, \ldots, \nu_N$ is minimized. To accomplish this, auxiliary variables $r_1 = \|\mathbf{s}\|^2$ and $r_2 = T^2$ are introduced to relax the problem into a convex one, which is then solved using SDP solvers. The idea of squaring the measurements and applying SDP was also explored in the recent work [22] as well as in [3, 12, 17, 19, 29, 31, 33, 39].

Solving Problem (2) Directly. To date, only a limited number of studies have focused on methods that directly address the joint asynchronous ToA problem in (2) without resorting to convex relaxations or smooth approximations.

In [40], the authors propose a standard Fixed-Point (FP) method for tackling Problem (2) by taking the gradient of the function F with respect to $\mathbf{s} \in \mathbb{R}^n$ and equating it to zero. While the well-definedness of this FP method cannot be assured in all scenarios (as the gradient may be undefined) and its convergence is not guaranteed, it offers the benefit of low computational complexity.

In [9], the authors propose the PAMP algorithm, which utilizes a proximal variant of the Alternating Minimization scheme to address Problem (2). In each iteration, the algorithm first calculates a proximal step with respect to the time offset. Subsequently, using the latest time offset, PAMP minimizes Problem (2) with respect to $\mathbf{s} \in \mathbb{R}^n$. This process continues until a stopping criterion is met. While the sub-problem with respect to T can be solved exactly, the minimization with respect to $\mathbf{s}$ is non-convex and non-smooth. To handle this, the authors employ iterations of a variant of the stochastic Barzilai-Borwein sub-gradient method with a backtracking step-size procedure. The authors claim to prove convergence of function values.

In [11], the authors address the problem directly using the Levenberg-Marquardt scheme, despite the fact that the problem is non-smooth and its Jacobian matrix is not well-defined. While this scheme demonstrates promising performance, it is important to note that these results are observed only when the scheme is initialized with a point obtained from the SDP convex relaxation that is proposed in their paper.

6 Numerical Analysis

In order to evaluate our FDLS method, we numerically compare it with the following algorithms:

1. FP of [40]: a first-order algorithm directly tackling Problem (2) by utilizing the fixed-point approach.

2. PAMP of [9]: a first-order algorithm that directly tackles Problem (2) using the stochastic Barzilai-Borwein sub-gradient method with a backtracking procedure. This method involves the tuning of seven hyper-parameters, with two of them, $0 < \alpha_{\min} \leq \alpha_{\max}$, defining the lower and upper bounds on the possible step-sizes, respectively. The code for PAMP was partially taken from [10]. In all experiments to follow we set $\alpha_{\min} = 0.001$ and $\alpha_{\max} = 1$.

3. SDP of [32]: a convex relaxation of Problem (2) solved with a commercial off-the-shelf SDP solver.

4. WLS of [22]: a convex relaxation of a smooth approximation of Problem (2), that introduces a weighted least squares problem, which is solved using a commercial off-the-shelf SDP solver.

6.1 Computational Complexity Analysis

The computational costs per iteration of the methods are summarized in Table 1 and are discussed now. The first-order methods FDLS, FP, and PAMP involve calculating N norm terms of size n. Additionally, FDLS computes the minimal eigenvalue of an $n \times n$ matrix to determine the step-size. For PAMP, its total computational cost also depends on the number of backtracking iterations performed in each inner iteration, with the unknwon average number of backtracking iterations denoted by $\bar{t}$. Consequently, the per iteration computation cost of FP, FDLS, and PAMP is $\mathcal{O}(nN)$, $\mathcal{O}\left(n^3 + nN\right)$ and $\mathcal{O}(nN\bar{t})$, respectively. It is worth mentioning that by choosing the step-size of FDLS according to the rule in Remark 2(iv), its computation complexity reduces to $\mathcal{O}\left(n^2 + nN\right)$. Notice that all three are linear in N. As for SDP and WLS, they utilize a primal-dual interior-point method with the commercial solver SeDuMi. Following [26], they have a high total computation cost of $\mathcal{O}\left(n^{4.5}N^5 + n^{3.5}N^7\right)$ and $\mathcal{O}\left(n^{6.5} + N^3\right)$, respectively.

Table 1 summarizes these algorithms. The table indicates which formulation of the problem each method addresses, their total computational cost, and whether or not they are proven to converge to critical points.

Table 1: Comparison of selected algorithms.

Method	Problem	Computational Cost	Convergence
FDLS	(2)	$\mathcal{O}\left(n^3 + nN\right)$	✓
PAMP [9]	(2)	$\mathcal{O}(nN\bar{t})$	✗
FP [40]	(2)	$\mathcal{O}(nN)$	✗
SDP [32]	(2) relaxation	$\mathcal{O}\left(n^{4.5}N^5 + n^{3.5}N^7\right)$	✗
WLS [22]	(2) approximation	$\mathcal{O}\left(n^{6.5} + N^3\right)$	✗

6.2 Generating the Data and Description of the Metrics

We investigate planar arrays with sensor locations and a true source location, denoted as $\mathbf{s}^{\text{true}} \in \mathbb{R}^n$, uniformly distributed over the box $[0, 1]^2$. Furthermore, the true time offset $T^{\text{true}} \in \mathbb{R}$ is randomly drawn

from a uniform distribution over the line segment $[-0.04, 0.04]$. We examine two scenarios: smaller-scale arrays with $N = 5$ sensors and larger-scale arrays with $N = 15$ sensors. Notice that the true source location can be either inside or outside the convex hull of the sensors.

All three iterative methods (FDLS, FP, and PAMP) were executed for a fixed total number of iterations $T_{\text{iter}} = 20000$. Here, total iterations refers to the cumulative count of both inner and outer iterations. For instance, if FDLS performed two APG iterations in the first and second outer iterations, the total iteration counter would read $T_{\text{iter}} = 4$. For PAMP, the total number includes the number of backtracking iterations performed within each inner iteration (this number cannot be predetermined).

The number of inner iterations $m_k \in \mathbb{N}$ in the k-th outer iteration of FDLS, for any $k \geq 0$, is determined by the rule (22). Below, we set $s = 15, 30$ and $r = 1000$. Consequently, the number of APG iterations is explicitly given by $m_k = s + 2^{\lfloor 0.001k \rfloor} - 1$. The integer parameter r of FDLS does not significantly affect the numerical results since it only governs the number of outer iterations after which an increase in the number of inner iterations occurs. Regarding the integer parameter s of FDLS, we will discuss below that choosing larger values of s may yield satisfactory results. As for PAMP, we fix the number of inner iterations to $s = 15, 30$ as well.

We generated a total of $K = 50$ random arrays (as described earlier). The noise is assumed to be Gaussian of the form $\epsilon_i \sim \text{Normal}\left(0, \sigma_{\text{time}}^2\right)$ for all $i = 1, 2, \ldots, N$. We considered six values of σ_{time}, logarithmically scaled in the interval $\left[10^{-4}, 10^{-1}\right]$. For each σ_{time} value, we drew $R = 50$ realizations of the noise. In other words, for each random array and each σ_{time}, we generated 50 different values of each ϵ_i. Consequently, a total of 15000 Monte-Carlo trials were performed for the larger-scale and smaller-scale scenarios. We should mention that in real-world scenarios, the value of the true σ_{time} may be unknown. Therefore, in the subsequent experiments, we employed σ_{time} solely for dataset generation purposes. For all algorithms, $\sigma_1 = \sigma_2 = \ldots = \sigma_N$ was set to 1 without variation [8].

Regarding the initial starting point $\mathbf{s}^0 \in \mathbb{R}^n$, we consistently set it to $\mathbf{s}^0 = (0.5, 0.5)$ for all iterative methods (FDLS, FP, and PAMP). Furthermore, for the PAMP method, we fixed the initial time offset $T^0 \in \mathbb{R}$ to $T^0 = 0$ in all experiments.

In all descriptions, graphs, and tables below, the values represent averages over all Monte-Carlo trials, including $K = 50$ random arrays and $R = 50$ realizations for both larger-scale and smaller-scale scenarios.

All experiments were ran on an Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz with a total of 300 GB RAM and 72 threads, using MATLAB 2022a.

6.3 Decrease in ML Function Value

The goal of all methods is to minimize the original ML function value given in Problem (2). Therefore, here we analyze how each method performs in minimizing this function. In Figure 1, we present plots of the function values along the iterations for each of the methods, considering a noise level of $\sigma_{\text{time}} = 10^{-4}$, for both the larger-scale and smaller-scale array scenarios. For SDP and WLS, which provide only a single output value at the end of their run, we included only this output value for comparison. In Table 2, we provide a summary of the results for the output function values obtained by each method (i.e., the value at the last iteration $T_{\text{iter}} = 20000$) for low, medium, and high noise levels.

Figure 1 shows that FDLS achieves the lowest values for both larger-scale and smaller-scale arrays. Notably, FDLS with $s = 15$ performs better than FDLS with $s = 30$ for the same number of iterations. However, as we will discuss in Section 6.5, FDLS with $s = 30$ exhibits slightly faster performance. Consequently, considering these factors, it is evident that FDLS with $s = 15$ and $s = 30$ yield similar results. Hence, our recommendation is to use FDLS with any sufficiently large number of inner iterations. The observation that a higher count of inner iterations facilitates quicker convergence, consequently minimizing the practical distinction between varying numbers of inner iterations, has also been substantiated in previous studies like [13, 14].

As for PAMP, it exhibits a relatively good performance in the smaller-scale arrays. However, it diverges in the larger-scale arrays, especially for $s = 30$ inner iterations. This is attributed to the stochastic nature of PAMP, where the correct configuration of its hyper-parameters, such as the lower and upper bounds of the step-size (see $\alpha_{\min}$ and $\alpha_{\max}$ above) and the number of inner iterations, should be tailored specifically for each array. Unfortunately, the authors of PAMP did not provide any rule of thumb for setting these hyper-parameters. Regarding SDP and WLS, they perform well in both instances, and FP converges after few iterations to a relatively high value.

The observations above are consistent across various values of σ_{time}, as indicated in Table 2 for selected noise factor values. However, it is important to note that the smaller-scale scenario with the highest noise level was found to be unstable for all methods. Notably, FDLS with $s = 15$ did not yield an average output value due to divergence to infinity in one realization of one of the networks (we recall that FDLS is ensured to converge to a unique critical point only when the generated sequence is bounded.) This emphasizes the recommendation to use higher values of inner iterations for FDLS when possible.

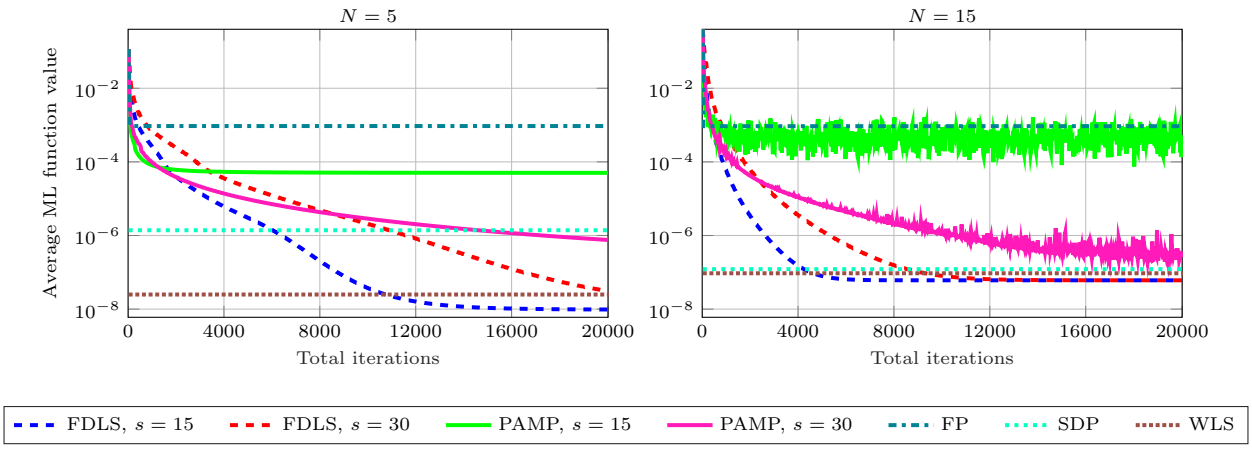

Figure 1: Average ML function value (y-axis in a logarithmic scale) of all methods is shown along the iterations for $\sigma_{\text{time}} = 10^{-4}$, separately for the smaller-scale (left) and larger-scale arrays (right). For SDP and WLS, only the output value is plotted.

Table 2: Average output (last iteration) ML function values for low, medium, and high noises, with $N = 5$ and $N = 15$. The lowest value in each instance is indicated in bold.

Method	$\sigma_{\text{time}} = 10^{-4}$		$\sigma_{\text{time}} = 1.6 \cdot 10^{-3}$		$\sigma_{\text{time}} = 10^{-1}$	
	$N = 5$	$N = 15$	$N = 5$	$N = 15$	$N = 5$	$N = 15$
FDLS $s = 15$	**9.8e-9**	**6.1e-8**	**2.6e-6**	**1.5e-5**	–	**0.0586**
FDLS $s = 30$	3.1e-8	6.1e-8	2.6e-6	1.5e-5	**0.0098**	0.0586
PAMP $s = 15$	5.0e-5	0.0005	5.3e-5	0.0008	0.0099	0.0597
PAMP $s = 30$	7.6e-7	2.4e-7	3.4e-6	1.5e-5	0.0100	0.0587
FP	0.0009	0.0009	0.0009	0.0009	0.0166	0.0661
SDP	1.4e-6	1.2e-7	0.0003	3.1e-5	1.2	0.5327
WLS	2.5e-8	9.4e-8	2.6e-6	1.5e-5	0.5243	0.4342

6.4 Decrease in Squared Norm of Bias

Here we evaluate the performance in recovering the true source location $\mathbf{s}^{\text{real}} \in \mathbb{R}^n$ and the true time offset $T^{\text{real}} \in \mathbb{R}$. To this end, we consider the following formula for the squared norm of the bias at iteration $m \geq 0$

$$\text{Bias}^m \equiv \left\|\mathbf{s}^m - \mathbf{s}^{\text{real}}\right\|^2 + \left(T^m - T^{\text{real}}\right)^2.$$

In Figure 2, we plot the bias along the iterations for each of the methods, considering a noise level of $\sigma_{\text{time}} = 10^{-4}$, for both larger-scale and smaller-scale array scenarios. For SDP and WLS, we included only the output value for comparison. In Table 3, we summarize the results for the output bias values obtained by each method (i.e., the value at the last iteration $T_{\text{iter}} = 20000$) for low, medium, and high noise levels.

Figure 2 illustrates that FDLS achieves the lowest bias. We mention that FDLS with $s = 30$ is still decreasing after 20000 iterations. Hence, considering FDLS's faster computational speed with larger number of inner iterations (see Section 6.5 below), we recommend setting FDLS with any large enough number of

inner iterations. On the other hand, PAMP with $s = 30$ exhibits the same diverging behavior discussed in Section 6.3. Additionally, FP performs poorly in both scenarios, while SDP and WLS give good results.

The observations above are consistent across various values of σ_{time}, as shown in Table 3 for selected noise factors. FDLS performs the best for all noise levels, except for the largest one where it exhibits the same behavior as discussed in Section 6.3. Remarkably, in the case of the highest σ_{time}, no discernible correlation exists between lower values of the ML function and lower biases across all methods. This observation suggests that the ML approach provides sub-optimal estimations of the true location and offset when confronted with substantial levels of noise.

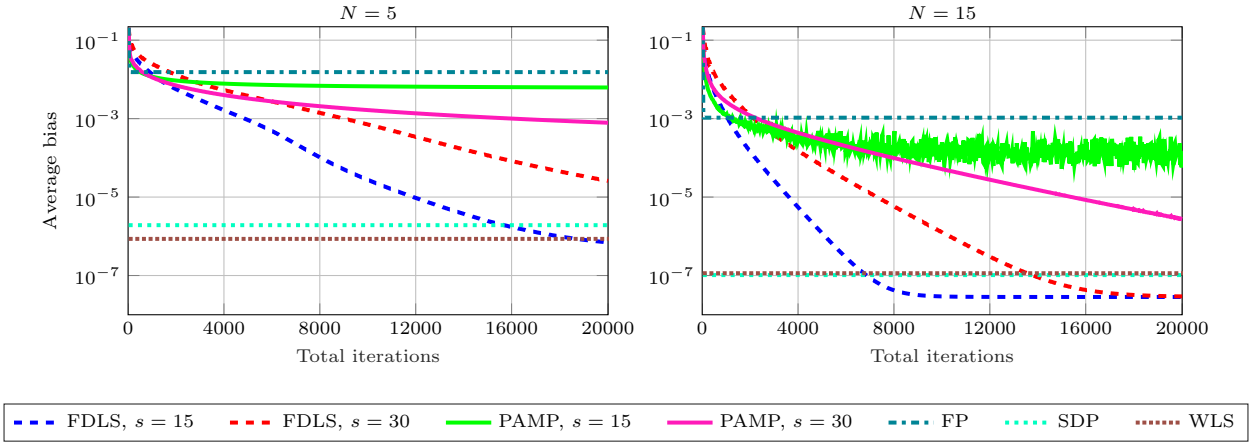

Figure 2: Average squared norm of bias (y-axis in a logarithmic scale) of all methods is shown along the iterations for $\sigma_{\text{time}} = 10^{-4}$, separately for the smaller-scale (left) and larger-scale arrays (right). For SDP and WLS, only the output value is plotted.

Table 3: Average output (last iteration) squared norm of bias for low, medium, and high noises, with $N = 5$ and $N = 15$. The lowest value in each instance is indicated in bold.

Method	$\sigma_{\text{time}} = 10^{-4}$		$\sigma_{\text{time}} = 1.6 \cdot 10^{-3}$		$\sigma_{\text{time}} = 10^{-1}$	
	$N = 5$	$N = 15$	$N = 5$	$N = 15$	$N = 5$	$N = 15$
FDLS $s = 15$	**7.0e-7**	**2.8e-8**	**0.0001**	**7.1e-6**	–	2.9643
FDLS $s = 30$	2.6e-5	2.9e-8	0.0001	7.1e-6	16.731	0.6221
PAMP $s = 15$	0.0062	0.0002	0.0063	0.0002	0.2337	0.0339
PAMP $s = 30$	0.0008	2.8e-6	0.0008	9.4e-6	0.1590	0.0353
FP	0.0153	0.0011	0.0154	0.0011	**0.0348**	0.0070
SDP	1.9e-6	1.0e-7	0.0004	2.0e-5	0.7922	0.2125
WLS	9.0e-6	1.1e-7	0.0001	7.2e-6	0.0489	**0.0023**

6.5 Run Time

Table 4 presents the average run time per iteration and the total run time (i.e., the time it took to complete 20000 iterations) for $\sigma_{\text{time}} = 10^{-4}$. Since SDP and WLS are not iterative, only the total run time is given.

It is worth noting that the noise level σ_{time} does not significantly affect the running times, but rather the quality of the solution. Hence, in Table 4, we focus on the values corresponding to $\sigma_{\text{time}} = 10^{-4}$.

As observed in Table 4, FDLS with $s = 30$ is faster than FDLS with $s = 15$. For a fixed number of total iterations, FDLS with $s = 30$ performs fewer updates of $\mathbf{u}^k$ (step 3 in Algorithm 1) compared to FDLS with $s = 15$, resulting in a lower average computation time per iteration. Therefore, despite FDLS with $s = 15$ yielding better results in terms of function value and bias (as seen in Sections 6.3 and 6.4), considering these factors, it is preferable to initialize FDLS with a larger value of inner iterations. Additionally, FP is the fastest among all methods, suggesting that it converges quickly to poor solutions. As for PAMP, it performs slowly compared to other first-order methods, likely due to the backtracking iterations that require additional calculations of function values. Finally, despite the fact that SDP and WLS employ SDP solvers known for their high computational demands, their execution time remains considerable yet on par with that of first-order methods.

Table 4: Average run time in seconds, for $N = 5$ and $N = 15$, with noise factor $\sigma_{\text{time}} = 10^{-4}$. The fastest value in each instance is indicated in bold.

Method	Run time per iteration		Total run time	
	$N = 5$	$N = 15$	$N = 5$	$N = 15$
FDLS $s = 15$	6.8e-6	8.1e-6	0.1368	0.1618
FDLS $s = 30$	6.3e-6	7.2e-6	0.1262	0.1434
PAMP $s = 15$	3.4e-5	6.5e-5	0.6598	1.3027
PAMP $s = 30$	6.5-5	0.0001	1.2950	2.6241
FP	**3.2e-6**	**4.0e-6**	**0.0643**	**0.0791**
SDP	–	–	0.2335	0.2653
WLS	–	–	0.4381	0.4157

6.6 Estimated RMSE and CRLB

The estimated Root Mean Squared Error (RMSE) is

$$\widehat{\text{RMSE}} \equiv \sqrt{\text{trace}\left(\frac{1}{R}\sum_{i=1}^{R}\left(\boldsymbol{\theta}^i - \bar{\boldsymbol{\theta}}\right)\left(\boldsymbol{\theta}^i - \bar{\boldsymbol{\theta}}\right)^T\right)},$$

where $\boldsymbol{\theta}^i \equiv \left(\mathbf{s}^i, T^i\right) \in \mathbb{R}^{n+1}$ is the output (i.e., the last iteration) of the source and time offset generated by a method in the $i = 1, 2, \ldots, R$ realization, and where $\bar{\boldsymbol{\theta}} \equiv \frac{1}{R}\sum_{i=1}^{R}\boldsymbol{\theta}^i$ is the empirical mean.

In Figure 3, we present the average (over all arrays) RMSE and Cramér–Rao Lower Bound (CRLB) [9] for all σ_{time}. Table 5 provides a summary of selected low, medium, and high noises. Since all methods have values above the CRLB, it indicates that they are not necessarily biased with respect to the true location and true offset. FP stands out with the lowest RMSE values across all configurations, suggesting that it is

relatively robust. However, FP also yields poor results in terms of function value and bias, indicating its trade-off between robustness and accuracy. On the other hand, the divergence of PAMP with $s = 30$ in the larger-scale scenario leads to less robust solutions for this method. All other methods show relatively similar performance, with SDP slightly worse. We should also mention that, as discussed earlier, FDLS with $s = 15$ did not yield a value for the largest noise in the smaller-scale scenario, indicating the need for larger enough values of inner iterations to enhance stability in such scenarios, without compromising the results.

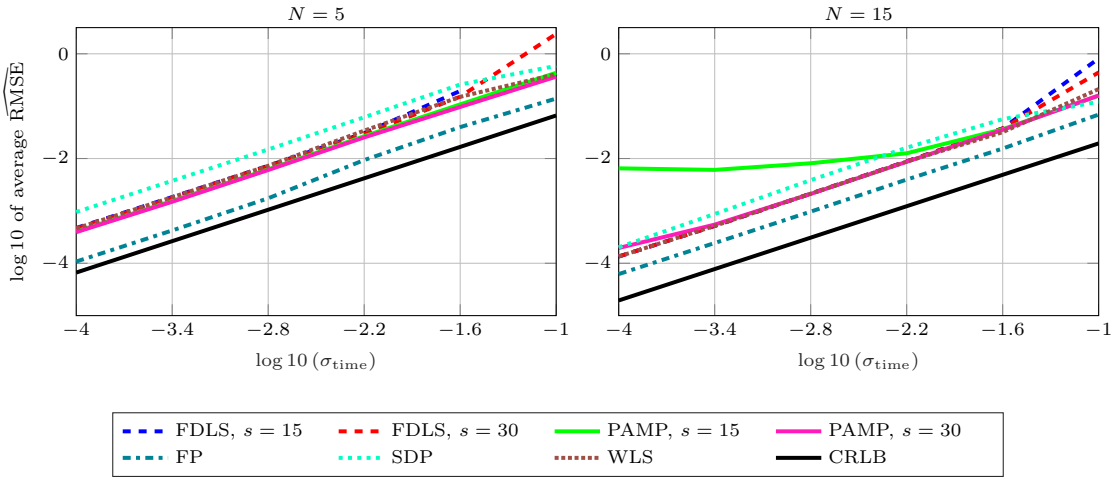

Figure 3: Average $\widehat{\text{RMSE}}$ (y-axis in a $\log_{10}$ scale) of all methods, as a function of the noise levels (x-axis in a $\log_{10}$ scale), separately for the smaller-scale (left) and larger-scale arrays (right).

Table 5: Average estimated RMSE for low, medium, and high noises, with $N = 5$ and $N = 15$, compared to the average CRLB. The lowest value in each instance is indicated in bold.

Method	$\sigma_{\text{time}} = 10^{-4}$		$\sigma_{\text{time}} = 1.6 \cdot 10^{-3}$		$\sigma_{\text{time}} = 10^{-1}$	
	$N = 5$	$N = 15$	$N = 5$	$N = 15$	$N = 5$	$N = 15$
FDLS $s = 15$	0.0005	0.0001	0.0073	0.0021	–	0.8156
FDLS $s = 30$	0.0005	0.0001	0.0070	0.0021	2.4020	0.4404
PAMP $s = 15$	0.0004	0.0065	0.0062	0.0081	0.4347	0.1586
PAMP $s = 30$	0.0004	0.0002	0.0060	0.0021	0.3656	0.1601
FP	**0.0001**	**6.2e-5**	**0.0017**	**0.0010**	**0.1398**	**0.0691**
SDP	0.0010	0.0002	0.0149	0.0161	0.5878	0.1214
WLS	0.0005	0.0001	0.0072	0.0021	0.4094	0.2122
CRLB	6.6e-5	1.9e-5	0.0011	0.0003	0.0663	0.0195

7 Conclusion

We addressed the challenging joint source localization and synchronization problem using asynchronous ToA measurements, which involves recovering both the unknown source location and time offset. To tackle this non-convex and non-smooth problem, we developed the FDLS algorithm which consists of the NAM

optimization scheme together with APG to iteratively minimize the dual functions. We provided a theoretical proof that FDLS converges to critical points of the original primal objective function, under relatively mild assumptions. This result establishes the first convergence guarantees for the joint problem, as well as for APG as a nested algorithm in NAM applied on the dual problems. Moreover, numerical experiments demonstrate the advantages of FDLS in terms of complexity, function value, bias, run time, and RMSE.

A Proof of Lemma 1

Proof. The proof that there exists $\beta > 0$ such that $\beta < \lambda_{\min}\left(\mathbf{A}^k\right)$ for any $k \geq 0$ follows by similar arguments as in [13, Lemma 1]. Now, since $\left\|\bar{\mathbf{u}}_i^k\right\| \cdot \left\|\bar{\mathbf{v}}_j^k\right\| \leq 1$ for any $k \geq 0$ and $i < j$ (see (10) and (11)), from Weyl's inequality [16, Theorem 4.3.1] we get

$$\lambda_{\max}\left(\mathbf{A}^k\right) \leq \frac{N-1}{2} + \frac{1}{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left\|\bar{\mathbf{u}}_i^k\right\| \cdot \left\|\bar{\mathbf{v}}_j^k\right\| \leq N-1,$$

which completes the proof. □

B Proofs of Propositions 1 and 2

First, we make the following assumption regarding the sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$.

Assumption 1. Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be generated by FDLS. Then, there exist $K \geq 0$ and $q > 0$ such that

$$\inf_{k\geq K}\min_{i=1,2,\ldots,N}\left\|\mathbf{s}^k - \mathbf{p}_i\right\| \geq q.$$

Assumption 1 can be expressed as follows: starting from a certain iteration, the sequence does not approach any sensor. In practical terms, this assumption is reasonable since it is highly unlikely for the true unknown source location to align precisely with any of the sensor locations. It is important to note that there is no restriction on the possibility that, in some iterations, the sequence generated by FDLS indeed coincides with one of the sensor locations. Assumption 1 can be satisfied, for example, by obtaining prior knowledge that the optimal solution of Problem (2) does not coincide with any of the sensor locations. In such cases, FDLS can be initialized with a point that has a lower function value compared to all sensor locations, ensuring the fulfillment of Assumption 1 [7].

Now, we prove inequalities concerning APG in the context of NAM that will later enable us to prove the two propositions. To this end, and following Lemma 1, the partial functions $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$ are 2β-strongly

convex for any $k \geq 0$ [2, Example 5.19]. Similarly, $\boldsymbol{\zeta}_i \mapsto \tilde{\Psi}^k(\boldsymbol{\zeta})$, for any $1 \leq i \leq N$ and $k \geq 0$, are also strongly convex. In particular, they admit a minimizer $\mathbf{s}_*^k$ and $\boldsymbol{\zeta}_{i*}^k$, respectively. We denote by $\boldsymbol{\zeta}_*^k \in \mathbb{R}^{nN}$ the concatenation of $\boldsymbol{\zeta}_{i*}^k$ into a single column, and using convexity, it follows that $\boldsymbol{\zeta}_*^k$ is a minimizer of the convex function $\tilde{\Psi}^k$.

Using the primal-dual relation (15), we denote $\mathbf{s}^{k,m} = \mathbf{s}(\mathbf{y}^m)$, $m = 0, 1, \ldots, m_k$, the sequence of primal APG iterations, and recall that $\mathbf{s}^{k+1} = \mathbf{s}^k\left(\boldsymbol{\zeta}^{k+1}\right)$, where $\boldsymbol{\zeta}^{k+1} = \mathbf{y}^{m_k}$ (step 11 in Algorithm 1), and $\mathbf{y}^0 = \boldsymbol{\zeta}^k$.

Lemma A. *Let $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ be a bounded sequence generated by FDLS. Then, there exists $M > 0$ such that for any $k \geq 0$*

(i) $\left\|\boldsymbol{\zeta}^k - \boldsymbol{\zeta}_*^k\right\| \leq M.$

(ii) $\left\|\boldsymbol{\zeta}^{k+1} - \boldsymbol{\xi}^{m_k-1}\right\| \leq M/m_k.$

(iii) $\left\|\mathbf{s}^{k,m} - \mathbf{s}_*^k\right\| \leq M/m.$

(iv) $\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\| \leq M.$

(v) $\Psi\left(\mathbf{s}^{k,m}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) - \Psi\left(\mathbf{s}_*^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \leq M/m.$

Proof. First, notice that $\mathbf{y}^m$, for any $k \geq 0$ and $m \geq 0$, is bounded. Indeed, from (21) we have $\left\|\mathbf{y}_i^{k,m}\right\| \leq \frac{t}{N}$ for any $i = 1, 2, \ldots, N$. Therefore, there exists $C_1 > 0$ such that

$$\left\|\boldsymbol{\zeta}^k - \boldsymbol{\zeta}_*^k\right\| \leq C_1. \tag{25}$$

Since $\Psi^k(\boldsymbol{\zeta})$ is a composition of a strongly convex function with a linear transformation, it follows from [34, Theorem 3.7] that there exists $\tilde{C}_2 > 0$ such that

$$\left\|\mathbf{y}^{m+1} - \mathbf{y}^m\right\| \leq \frac{\tilde{C}_2}{m}, \quad \forall m \geq 0. \tag{26}$$

Using steps 8 and 9 in Algorithm 1, triangle inequality and (26), we get that there exists $C_2 > 0$ such that

$$\left\|\mathbf{y}^{m+1} - \boldsymbol{\xi}^m\right\| \leq \left\|\mathbf{y}^{m+1} - \mathbf{y}^m\right\| + \frac{\alpha_{m-1} - 1}{\alpha_m}\left\|\mathbf{y}^m - \boldsymbol{\xi}^{m-1}\right\| \leq \frac{C_2}{m}, \tag{27}$$

where we used the fact that $\alpha_0 = 1$ and $\alpha_{m-1} \leq \alpha_m$ for any $m \geq 0$. In particular, (27) holds for $m = m_k$.

Now, using (25) together with [6, Theorem 4.1], it follows that there exists $C_3 > 0$ such that

$$\left\|\mathbf{s}^{k,m} - \mathbf{s}_*^k\right\| \leq \frac{C_3}{m}. \tag{28}$$

In addition, since the sequence $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ is bounded then there must exist $C_4 > 0$ such that

$$\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\| \leq C_4. \tag{29}$$

To prove item (iv), we notice that since $\left\{\mathbf{s}^k\right\}_{k\geq 0}$ is bounded, then surely $\left\{\mathbf{s}^{k,m}\right\}_{m\geq 0}$, for any $k \geq 0$, are also bounded. In addition, recall that $\psi(\mathbf{s})$ (see (8)) is a sum of norms, hence $\partial\psi(\mathbf{s}) \subseteq t \cdot \mathcal{B}$, where $\mathcal{B}$ is the unit ball in $\mathbb{R}^n$ (see [2, Theorem 3.36]). In particular, there exists some $\gamma > 0$ such that

$$\max_{k,m} \max_{\mathbf{d}\in\partial_{\mathbf{s}}\Psi(\mathbf{s}^{k,m},\mathbf{u}^k,\mathbf{v}^k)} \|\mathbf{d}\| = \max_{k,m} \max_{\mathbf{g}\in\mathcal{B}} \left\|2\mathbf{A}^k\mathbf{s}^{k,m} - \mathbf{z}^k + t\mathbf{g}\right\| \leq \gamma.$$

Now, by combining Lemma 1, (25) and [6, Theorem 4.3], we derive that there exists $C_5 > 0$ such that

$$\Psi\left(\mathbf{s}^{k,m}, \mathbf{u}^k, \mathbf{v}^k\right) - \Psi\left(\mathbf{s}_*^k, \mathbf{u}^k, \mathbf{v}^k\right) \leq \frac{C_5}{m}. \tag{30}$$

Finally, the result follows from (25), (27), (28), (29) and (30) by setting $M = \max_{i\in\{1,2,3,4,5\}}\{C_i\}$. □

Proof of Proposition 1. From the 2β-strong convexity of the partial function $\mathbf{s} \mapsto \Psi\left(\mathbf{s}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$, we get

$$\Psi\left(\mathbf{s}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) - \Psi\left(\mathbf{s}_*^k, \bar{\mathbf{u}}^k, \mathbf{v}^k\right) \geq \beta\left\|\mathbf{s}^k - \mathbf{s}_*^k\right\|^2,$$

and therefore

$$\Psi\left(\mathbf{s}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) - \Psi\left(\mathbf{s}^{k+1}, \bar{\mathbf{u}}^{k+1}, \bar{\mathbf{v}}^{k+1}\right) - \left(\Psi\left(\mathbf{s}_*^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) - \Psi\left(\mathbf{s}^{k+1}, \bar{\mathbf{u}}^{k+1}, \bar{\mathbf{v}}^{k+1}\right)\right) \geq \beta\left\|\mathbf{s}^k - \mathbf{s}_*^k\right\|^2. \tag{31}$$

Now, from the Cauchy-Schwartz inequality together with Lemma A(iv) we get

$$\beta\left\|\mathbf{s}^k - \mathbf{s}_*^k\right\|^2 \geq \beta\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\|^2 - 2\beta\left\|\mathbf{s}^{k+1} - \mathbf{s}_*^k\right\| \cdot \left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\| \geq \beta\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\|^2 - 2\beta M\left\|\mathbf{s}^{k+1} - \mathbf{s}_*^k\right\|.$$

Combining the last inequality with (31), Lemma A(iii) and Lemma A(v) we get

$$\Psi\left(\mathbf{s}^k, \mathbf{u}^k\right) - \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}\right) \geq \beta\left\|\mathbf{s}^{k+1} - \mathbf{s}^k\right\|^2 - \frac{\max\left\{2\beta M^2, M\right\}}{m_k},$$

and we set $M_1 = \max\left\{2\beta M^2, M\right\} > 0$.

Last, we prove that $\Psi\left(\mathbf{s}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \geq \Psi\left(\mathbf{s}^{k+1}, \bar{\mathbf{u}}^{k+1}, \bar{\mathbf{v}}^{k+1}\right)$ for all $k \geq 0$. From Lemma A(v) we get $\Psi\left(\mathbf{s}^{k,m}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \to \Psi\left(\mathbf{s}_*^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$ as $m \to \infty$, so surely there exists $m_k \geq 0$ such that $\Psi\left(\mathbf{s}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \geq$

$\Psi\left(\mathbf{s}^{k+1}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$. Now, since $\left(\bar{\mathbf{u}}^{k+1}, \bar{\mathbf{v}}^{k+1}\right)$ is a minimizer of $(\mathbf{u}, \mathbf{v}) \mapsto \Psi\left(\mathbf{s}^{k+1}, \mathbf{u}, \mathbf{v}\right)$ (see step 3 in Algorithm 1), the required result follows. □

Remark 3. Proposition 1 establishes that for any iteration $k \geq 0$ of FDLS, there exists a number $m_k \in \mathbb{N}$ of APG iterations that leads to a decrease in the value of the primal function. The selection of the number of APG iterations follows the rule outlined in (22). It is ensured that by employing this number of APG iterations, the error terms are summable (see [14]), which typically corresponds to a decrease in the primal function in real-world scenarios. To further confirm the decrease in function value, a straightforward stopping criterion can be incorporated into step 11 of FDLS that is guaranteed to be satisfied: if $\Psi\left(\mathbf{s}^{k+1}, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right) \leq \Psi\left(\mathbf{s}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k\right)$, then continue. Otherwise, repeat step 6 for additional APG iterations.

Proof of Proposition 2. From the first-order optimality condition applied on (19) we get for any $i = 1, 2, \ldots, N$ and $k \geq 0$

$$\boldsymbol{\omega}_i^{k+1} \equiv L^k \left(\boldsymbol{\xi}_i^{m_k - 1} - \boldsymbol{\zeta}_i^{k+1}\right) - \nabla_i \tilde{\varphi}^k \left(\boldsymbol{\xi}^{m_k - 1}\right) \in \partial \mathcal{I}\left(\boldsymbol{\zeta}_i^{k+1}\right), \tag{32}$$

and therefore

$$\nabla_i \tilde{\varphi}^k \left(\boldsymbol{\zeta}^{k+1}\right) + \boldsymbol{\omega}_i^{k+1} \in \partial_i \tilde{\Psi}^k \left(\boldsymbol{\zeta}^{k+1}\right).$$

From Lipschitz continuity of $\nabla_i \tilde{\varphi}$, Lemma 1, Lemma A(ii) and the triangle inequality, there exists $D_1 > 0$ such that

$$\left\| \nabla_i \tilde{\varphi}^k \left(\boldsymbol{\zeta}^{k+1}\right) + \boldsymbol{\omega}_i^{k+1} \right\| \leq \frac{D_1}{m_k}. \tag{33}$$

As $\mathcal{I}$ is the indicator of the ball $\mathcal{B}\left[\mathbf{0}_n, t/N\right]$, its sub-differential set is (see, for example, [2, Example 3.6])

$$\partial \mathcal{I}\left(\mathbf{y}\right) = \begin{cases} \mathbf{0}_n, & \|\mathbf{y}\| < t/N, \\ \{\lambda \mathbf{y} \colon \lambda \geq 0\}, & \|\mathbf{y}\| = t/N, \\ \varnothing, & \|\mathbf{y}\| > t/N. \end{cases} \tag{34}$$

From (21) we know that $\|\mathbf{y}_i^m\| \leq t/N$ for any $k \geq 0$, $m \geq 0$ and $i = 1, 2, \ldots, N$. Therefore, combining (32) and (34), we derive that there exists some $\lambda_i^{k+1} \geq 0$ such that

$$\boldsymbol{\omega}_i^{k+1} = \lambda_i^{k+1} \boldsymbol{\zeta}_i^{k+1}, \tag{35}$$

which means that

$$\left\|\boldsymbol{\omega}_i^{k+1}\right\| = \frac{t}{N} \cdot \lambda_i^{k+1}. \tag{36}$$

We therefore get from (18), (32), (36), after using the triangle inequality and Lemma 1 that

$$-\frac{N}{2\beta}\left\|\boldsymbol{\xi}_i^{m_k-1}-\boldsymbol{\zeta}_i^{k+1}\right\|+\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\xi}_i^{m_k-1}\right)\right\|\leq\frac{t}{N}\lambda_i^{k+1}\leq\frac{N}{2\beta}\left\|\boldsymbol{\xi}_i^{m_k-1}-\boldsymbol{\zeta}_i^{k+1}\right\|+\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\xi}_i^{m_k-1}\right)\right\|.$$

Combining the last two inequalities together with Lemma A(ii), we derive that there exists some $D_2>0$ such that

$$\left|\frac{t}{N}\lambda_i^{k+1}-\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\xi}_i^{m_k-1}\right)\right\|\right|\leq\frac{D_2}{m_k}. \tag{37}$$

Now, by adding and subtracting $\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\xi}_i^{m_k-1}\right)\right\|$, we get from (37) and the triangle inequality that

$$\left|\frac{t}{N}\lambda_i^{k+1}-\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|\right|\leq\frac{D_2}{m_k}+L^k\left\|\boldsymbol{\zeta}_i^{k+1}-\boldsymbol{\xi}_i^{m_k-1}\right\|\leq\frac{D_2}{m_k}+\frac{NM}{2\beta m_k}, \tag{38}$$

where the first inequality follows from L^k-Lipschitz continuity of $\nabla\tilde{\varphi}^k$ with (18) and Lemma 1, and the second from Lemma A(ii). Hence, (38) implies that there exists $D_3>0$ such that

$$\left|\frac{t}{N}\lambda_i^{k+1}-\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|\right|\leq\frac{D_3}{m_k}. \tag{39}$$

Now, notice that using the primal-dual relation in (15), simple calculations show that

$$\nabla_{\mathbf{s}}\varphi\left(\mathbf{s}^{k+1},\bar{\mathbf{u}}^k,\bar{\mathbf{v}}^k\right)=\sum_{i=1}^N\boldsymbol{\zeta}_i^{k+1}. \tag{40}$$

and that for any $i=1,2,\ldots,N$ it holds that

$$\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)=\mathbf{s}^{k+1}-\mathbf{p}_i. \tag{41}$$

From (40) and the definition of Ψ we get

$$\partial_{\mathbf{s}}\Psi\left(\mathbf{s}^{k+1},\bar{\mathbf{u}}^k,\bar{\mathbf{v}}^k\right)=\sum_{i=1}^N\boldsymbol{\zeta}_i^{k+1}+\partial\psi\left(\mathbf{s}^{k+1}\right). \tag{42}$$

Following Assumption 1, the sequence does not approach any sensor location. Hence, from (42) with (8)

$$\partial_{\mathbf{s}}\Psi\left(\mathbf{s}^{k+1},\bar{\mathbf{u}}^k,\bar{\mathbf{v}}^k\right)=\sum_{i=1}^N\left(\boldsymbol{\zeta}_i^{k+1}+\frac{t}{N}\frac{\mathbf{s}^{k+1}-\mathbf{p}_i}{\|\mathbf{s}^{k+1}-\mathbf{p}_i\|}\right).$$

We denote the above LHS vector as $\mathbf{w}^{k+1}\in\mathbb{R}^n$ and we will bound its norm. From Assumption 1 we

have $\left\|\mathbf{s}^{k+1}-\mathbf{p}_i\right\| \geq q > 0$ for all i, and from the triangle inequality and (41)

$$\left\|\mathbf{w}^{k+1}\right\| \leq \frac{1}{q}\sum_{i=1}^{N}\left\|\left\|\mathbf{s}^{k+1}-\mathbf{p}_i\right\|\boldsymbol{\zeta}_i^{k+1}+\frac{t}{N}\left(\mathbf{s}^{k+1}-\mathbf{p}_i\right)\right\| = \frac{1}{q}\sum_{i=1}^{N}\left\|\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|\boldsymbol{\zeta}_i^{k+1}+\frac{t}{N}\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|.$$

Focusing on one of the elements in the summation in the RHS of the last inequality, we get from the triangle inequality by adding and subtracting $\frac{t}{N}\boldsymbol{\omega}_i^{k+1} = \frac{t}{N}\lambda_i^{k+1}\boldsymbol{\zeta}_i^{k+1}$ (see (35)) that

$$\begin{aligned}\left\|\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|\boldsymbol{\zeta}_i^{k+1}+\frac{t}{N}\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\| \leq{}& \left\|\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)\right\|\boldsymbol{\zeta}_i^{k+1}-\frac{t}{N}\lambda_i^{k+1}\boldsymbol{\zeta}_i^{k+1}\right\| \\ &+\frac{t}{N}\cdot\left\|\nabla_i\tilde{\varphi}^k\left(\boldsymbol{\zeta}^{k+1}\right)+\boldsymbol{\omega}_i^{k+1}\right\|.\end{aligned}$$

Now, since $\left\|\boldsymbol{\zeta}_i^{k+1}\right\| \leq \frac{t}{N}$ (see (21)), we get from the last inequality together with (33) and (39) that

$$\left\|\mathbf{w}^{k+1}\right\| \leq \frac{Nt}{N}\cdot\frac{D_2}{m_k}+\frac{Nt}{N}\cdot\frac{D_1}{m_k},$$

and the result follows with $M_2 \equiv t\cdot\max\left\{D_1,D_2\right\} > 0$. □

References

[1] Amir Beck. *Introduction to nonlinear optimization: theory, algorithms, and applications with MATLAB*, volume 19. SIAM, 2014.

[2] Amir Beck. *First-Order Methods in Optimization*, volume 25. SIAM, 2017.

[3] Amir Beck and Dror Pan. On the solution of the gps localization and circle fitting problems. *SIAM Journal on Optimization*, 22(1):108–134, 2012.

[4] Amir Beck, Petre Stoica, and Jian Li. Exact and approximate solutions of source localization problems. *IEEE Transactions on signal processing*, 56(5):1770–1778, 2008.

[5] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[6] Amir Beck and Marc Teboulle. A fast dual proximal gradient algorithm for convex minimization and applications. *Operations Research Letters*, 42(1):1–6, 2014.

[7] Amir Beck, Marc Teboulle, and Zahar Chikishev. Iterative minimization schemes for solving the single source localization problem. *SIAM Journal on Optimization*, 19(3):1397–1416, 2008.

[8] Yiu-Tong Chan and KC Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on signal processing*, 42(8):1905–1915, 1994.

[9] Yuanpeng Chen, Zhiqiang Yao, and Zheng Peng. A novel method for asynchronous time-of-arrival-based source localization: algorithms, performance and complexity. *Sensors*, 20(12):3466, 2020.

[10] Yuanpeng Chen, Zhiqiang Yao, and Zheng Peng. The PAMP method. Online: https://github.com/LeslieChan0513/localization-method, Jun. 2020, visited Jul. 13, 2023.

[11] Dalia El Badawy, Viktor Larsson, Marc Pollefeys, and Ivan Dokmanić. Localizing unsynchronized sensors with unknown sources. *IEEE Transactions on Signal Processing*, 71:641–654, 2023.

[12] Shangchao Gao, Fan Zhang, and Gang Wang. NLOS error mitigation for TOA-based source localization with unknown transmission time. *IEEE Sensors Journal*, 17(12):3605–3606, 2017.

[13] Eyal Gur, Alon Amar, and Shoham Sabach. Direct, fast and convergent solvers for the non-convex and non-smooth TDoA localization problem. *Digital Signal Processing*, 139:104074, 2023.

[14] Eyal Gur, Shoham Sabach, and Shimrit Shtern. Convergent nested alternating minimization algorithms for nonconvex optimization problems. *Mathematics of Operations Research*, 2022.

[15] Eyal Gur, Shoham Sabach, and Shimrit Shtern. Nested alternating minimization with FISTA for non-convex and non-smooth optimization problems. *Journal of Optimization Theory and Applications*, 199(3):1130–1157, 2023.

[16] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge university press, 2012.

[17] Jun Huang, Yanbo Xue, and Le Yang. An efficient closed-form solution for joint synchronization and localization using TOA. *Future Generation Computer Systems*, 29(3):776–781, 2013.

[18] R Jyothi and Prabhu Babu. SOLVIT: A reference-free source localization technique using majorization minimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2661–2673, 2020.

[19] Yimei Kang, Qingyang Wang, Jiawei Wang, and Renhai Chen. A high-accuracy toa-based localization method without time synchronization in a three-dimensional space. *IEEE Transactions on Industrial Informatics*, 15(1):173–182, 2018.

[20] Mayur Katwe, Pradnya Ghare, and Prabhat Kumar Sharma. Robust NLOS bias mitigation for hybrid RSS-TOA based source localization under unknown transmission parameters. *IEEE Wireless Communications Letters*, 10(3):542–546, 2020.

[21] D Russell Luke, Shoham Sabach, and Marc Teboulle. Optimization on spheres: models and proximal algorithms with computational performance comparisons. *SIAM Journal on Mathematics of Data Science*, 1(3):408–445, 2019.

[22] Xiao Ma, Benjian Hao, Hailin Zhang, and Pengwu Wan. Semidefinite relaxation for source localization by TOA in unsynchronized networks. *IEEE Signal Processing Letters*, 29:622–626, 2022.

[23] Zemene Walle Mekonnen and Armin Wittneben. Robust TOA based localization for wireless sensor networks with anchor position uncertainties. In *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pages 2029–2033. IEEE, 2014.

[24] Boris S Mordukhovich. *Variational Analysis and Generalized Differentiation I: Basic Theory*, volume 330. Springer Science & Business Media, 2006.

[25] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[26] Dimitri Peaucelle, Didier Henrion, Yann Labit, and Krysten Taitz. User's guide for SeDuMi interface 1.04. *LAAS-CNRS, Toulouse*, 2002.

[27] Nicola Piovesan and Tomaso Erseghe. Cooperative localization in WSNs: A hybrid convex/nonconvex solution. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):162–172, 2018.

[28] Hong Shen, Zhi Ding, Soura Dasgupta, and Chunming Zhao. Multiple source localization in wireless sensor networks based on time of arrival measurement. *IEEE Transactions on Signal Processing*, 62(8):1938–1949, 2014.

[29] Jiong Shi, Gang Wang, and Liping Jin. Moving source localization using TOA and FOA measurements with imperfect synchronization. *Signal Processing*, 186:108113, 2021.

[30] H. C. So. *Source Localization: Algorithms and Analysis*, chapter 3, pages 59–106. John Wiley & Sons, Ltd, 2018.

[31] Ming Sun and Le Yang. On the joint time synchronization and source localization using toa measurements. *International Journal of Distributed Sensor Networks*, 9(2):794805, 2013.

[32] Reza Monir Vaghefi and R Michael Buehrer. Asynchronous time-of-arrival-based source localization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4086–4090. IEEE, 2013.

[33] Gang Wang, Shu Cai, Youming Li, and Ming Jin. Second-order cone relaxation for TOA-based source localization with unknown start transmission time. *IEEE transactions on vehicular technology*, 63(6):2973–2977, 2013.

[34] Bo Wen, Xiaojun Chen, and Ting Kei Pong. Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems. *SIAM Journal on Optimization*, 27(1):124–145, 2017.

[35] Wenxin Xiong, Sneha Mohanty, Christian Schindelhauer, Stefan Johann Rupitsch, and Hing Cheung So. Convex relaxation approaches to robust RSS-TOA based source localization in NLOS environments. *IEEE Transactions on Vehicular Technology*, 2023.

[36] Enyang Xu, Zhi Ding, and Soura Dasgupta. Source localization in wireless sensor networks from signal time-of-arrival measurements. *IEEE Transactions on Signal Processing*, 59(6):2887–2897, 2011.

[37] Ge Yang, Yongsheng Yan, Haiyan Wang, and Xiaohong Shen. Improved robust TOA-based source localization with individual constraint of sensor location uncertainty. *Signal Processing*, 196:108504, 2022.

[38] Kehu Yang, Gang Wang, and Zhi-Quan Luo. Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals. *IEEE transactions on signal processing*, 57(7):2775–2784, 2009.

[39] Shouhong Zhu and Zhiguo Ding. Joint synchronization and localization using toas: A linearization based wls solution. *IEEE Journal On Selected areas in communications*, 28(7):1017–1025, 2010.

[40] Yanbin Zou, Jingna Fan, Liehu Wu, and Huaping Liu. Fixed point iteration based algorithm for asynchronous TOA-based source localization. *Sensors*, 22(18):6871, 2022.

[41] Yanbin Zou and Huaping Liu. Semidefinite programming methods for alleviating clock synchronization bias and sensor position errors in TDOA localization. *IEEE Signal Processing Letters*, 27:241–245, 2020.

[42] Yanbin Zou and Qun Wan. Asynchronous time-of-arrival-based source localization with sensor position uncertainties. *IEEE Communications Letters*, 20(9):1860–1863, 2016.