Distributed Scaled Proximal ADMM Algorithms for Cooperative Localization in WSNs

Mei Zhang, Zhiguo Wang, Feng Yin, Senior Member, IEEE, and Xiaojing Shen

Abstract—Distributed cooperative localization in wireless networks is a challenging problem since it typically requires solving a large-scale nonconvex and nonsmooth optimization problem. In this article, we reformulate the classic cooperative localization problem as a smooth and constrained nonconvex minimization problem while its loss function is separable over nodes. By utilizing the structure of the reformulation, we propose two novel scaled proximal alternating direction method of multipliers (SP-ADMM) algorithms, which can be implemented in a distributed manner. Compared with the classic semi-definite programming relaxation techniques, the proposed algorithms can provide more accurate position estimates with significantly lower computation complexity. The associated theoretical analysis shows that our algorithms globally converge to a KKT point of the reformulated problem and a critical point of the original problem, with a favorable sublinear $O(1/T)$ convergence rate, where $T$ is the iteration counter. Numerical experiments have consistently shown that the proposed SP-ADMM algorithms are superior to state-of-the-art methods in terms of localization accuracy and computational complexity across all tested scenarios, varying network size, number of anchors, average number of neighbors, and noise variance levels.

Index Terms—Proximal ADMM, distributed algorithm, global convergence rate, cooperative localization, wireless sensor network.

I. INTRODUCTION

Wireless sensor networks (WSNs) are widely used to deal with sensitive information in a variety of applications, including healthcare, military, Internet of Things, surveillance, and industrial [1]. In the aforementioned applications, all collected information is meaningful only when the locations of the sensor nodes are accurately known. Therefore, localization is an enabling technique for WSNs. The cooperative localization problem aims to determine multiple sensor locations with the aid of a relatively small portion of anchors whose positions are precisely known and the relative noisy range measurements of adjacent nodes [2]. The range measurements can be measured in a variety of ways, such as time-of-arrival (TOA) [3], [4], time-difference-of-arrival (TDOA) [5], angle-of-arrival (AOA) [6] and received signal strength (RSS) [7], [8]. In recent years, there has been a growing interest in estimating target positions through optimization techniques due to the faster response speed of position estimation by these methods and convergence guarantee.

Nonconvex and nonsmooth is the major difficulty of the least squares objective function in the maximum likelihood framework [9]. It is hard to find an optimal solution with low computational complexity, and in consequence, most existing methods resort to developing approximate solutions through applying, for instance, the relaxation techniques, to the original nonconvex problem. A popular class of methods is based on semi-definite programming (SDP) relaxation [10], [11], [12], which can be solved by the interior-point algorithm. Although convex relaxation techniques guarantee convergence to a global minimum point, it is not necessarily a critical point of the original nonconvex formulation. In addition, the above convex relaxation methods are implemented in a centralized framework, since all measurements are collected and processed at a processing center. Centralized methods are vulnerable to the malfunction of any single node. There have been increasing efforts in developing distributed sensor network localization methods [13], [14], [15].

Distributed methods are able to avoid the major drawbacks of the centralized counterparts for large-scale networks, thus making them well-suited for wireless sensor network localization. Distributed methods not just have lower computational and communication complexity due to scalability but are also more robust to node failures [16], [17], [18]. This inspired the use of distributed optimization approaches to solve the target positioning problem in large-scale networks. Compared with the centralized SDP methods, the work [19] proposed a more practical alternative. By leveraging the concept of convex envelope, the authors developed a method that is both scalable and simple to implement. They also analyzed the total number of iterations required by the algorithm to approach the optimal function value with a high probability for the convexified

---

1The term distributed in this article refers to an implementation that sensor nodes can locate themselves and neighboring nodes based on the local network information containing neighbors, without whole network data [9], [13].
problem. Despite these advances, it still can be difficult to infer where the solutions are the critical points of the original nonconvex problem (discussed in [20], [21]). In view of this, non-relaxed methods may be a workable viewpoint.

In this article, we focus on the first-order method that solves the original nonconvex localization problem directly. In general, the design of this kind of algorithm first reformulates the problem by exploiting the internal structure of the problem and then develops an efficient optimization algorithm to solve the reformulated problem. For example, a nonconvex sequential greedy (NCSG) optimization algorithm was proposed in [22]. It is also proved that this algorithm owns the convergence guarantee inherited from the non-linear Gauss-Seidel framework [23]. Notwithstanding, the limit point of the generated sequence converges to the KKT point of the reformulated problem, the connection with the original problem is yet not provided.

One efficient way is to reformulate the original localization problem as a two-block nonconvex optimization problem with linear equality constraints, and then a distributed ADMM method is proposed in [24], [25] to solve the resulting nonconvex reformulation. In [26], [27], they show that the ADMM algorithm may be unstable when the objective function is nonconvex and nonsmooth. In order to obtain a favorable initial point for the method developed in [24], the authors in [20] proposed a hybrid ADMM (ADMM-H) method that contains two stages. In the first stage, a reliable starting point of the target problem is given by exploiting the convex relaxation technique introduced in [19]. In the second stage, the solution will be constantly updated by adopting the method developed in [24]. While the simulations demonstrated faster convergence, there remain limitations both in the local minimum theoretical guarantee and practical implementation. This is discussed in detail by [21] and our Section V.

Recently, for the single source localization problem, the authors in [25] adopted a simple variational representation of the Euclidean norm, then derived an equivalent smooth reformulation with ball constraint. Later on, it was extended for the multi-source localization problem in [21], and they proposed an alternating minimization (AM) method to solve the reformulated constrained smooth and nonconvex optimization problem. There are two versions of the AM method in [21], the fully centralized (AM-FC) and the fully distributed (AM-FD). Due to the sequential nature of the AM-FD method, it is impractical to apply it to large-scale networks. Therefore, the authors further propose a unifying AM (AM-U) algorithm to remedy this difficulty. The basic idea of AM-U is to divide the sensors in the network into several disjoint clusters and then apply the AM-FD method separately for each cluster. The convergence of the AM-FD can be guaranteed. Since the difficulty of nonconvex and nonsmooth, the convergence rate for the original problem is not provided.

In this work, we aim to exploit the advantages of [24] and [25] to develop an efficient distributed algorithm with lower computational complexity for the nonconvex and nonsmooth localization problem and establish a theoretical guarantee of its performance.

Our contributions are summarized as follows.

- By introducing an auxiliary variable for the nonsmooth Euclidean norm, we reformulate the classic cooperative localization problem as a smooth and constrained nonconvex minimization problem, whose loss function is separable over nodes and has two block variables that eventually leads to a nice optimization structure.
- To exploit the nice structure of the new reformulation, we propose a scaled proximal ADMM (SP-ADMM) algorithm, which is suitable for distributed computing and implementation. Moreover, to further reduce the storage space at each node, we proposed Algorithm 2 that is a simplified version of Algorithm 1. As shown in Table I, the proposed algorithms enjoy lower computation complexity and storage space when compared with the existing SDP relaxation method and ADMM-H method, respectively.
- By utilizing a novel potential function, we demonstrate that the global convergence of the sequence generated by the proposed algorithm, that is, the whole sequence converges to a unique KKT point of the reformulated problem

| Algorithm       | Convex Relaxations | First order Method | Step Size | Convergence Rate | Parallelized |
|-----------------|--------------------|--------------------|-----------|-----------------|--------------|
| SP-ADMM (ours)  | ✔                  | ✔                  | fixed     | ✔               | ✔            |
| AM-FD [21]      | ✗                  | ✔                  | fixed     | ✗               | ✗            |
| ADMM-H [20]     | hybrid†             | ✗                  | -         | ✗               | ✔            |
| SDP [10]        | ✔                  | ✗                  | -         | ✗               | ✗            |
| SF [19]         | ✔                  | ✔                  | fixed     | ✗               | ✗            |

| Sensor i       | Computational Complexity | Communication Cost | Storage Space |
|----------------|--------------------------|--------------------|---------------|
| SP-ADMM (ours) | O(nN)                    | nN                 | nN            |
|               | 2n + N + 1               |                    |               |
| AM-FD [21]    | O(nN)                    | nN                 | nN            |
|               | 2n + N + 1               |                    |               |
| ADMM-H [20]   | O(nN)                    | nN                 | nN            |
|               | 2n + N + 1               |                    |               |
| SF [19]       | O(nN)                    | nN                 | nN            |
|               | 2n + N + 1               |                    |               |

* is due to the convergence analysis of the SF for the convex relaxation problem rather than the original nonconvex problem.
† Here, “hybrid” means a two-stage algorithm, including a convex relaxation stage and a nonconvex stage.
‡ T refers to the total number of iterations of the algorithm.
⋆ |E| denotes the number of edges in which both nodes are anchors. The value of storage space refers to the minimum total storage space required by the SDP per step, as described in reference [28].

In this article, sequential means sensor nodes perform their update calculations in turn, that is, they have to wait for a part of neighbor nodes to complete the update before running their local update steps.
and a critical point of the original problem. Remarkably, these algorithms also exhibit sublinear convergence, with a convergence rate of \( O(1/T) \), where \( T \) represents the iteration counter. To the best of our knowledge, this is the first result that shows the sublinear convergence rate of a distributed ADMM algorithm for a nonconvex and nonsmooth localization problem.

- Numerical experiments conducted on a variety of networks have consistently demonstrated that the proposed algorithms outperform existing methods in terms of both localization accuracy and computational efficiency. The experimental data encompasses scenarios with different network sizes, number of anchors, average number of neighbor nodes, and variance levels of the measurement noise.

**Notation:** We use lowercase bold letters to denote vectors, \( 1_{N_i} \) and \( 0_{N_i} \) to represent \( N_i \)-dimensional column vector with all ones and all zeros, respectively. Capital bold letters represent matrices, specially, \( I_n \) and \( O_{N} \) denote \( N \times N \)-dimensional identity matrix and zero matrix, respectively. \( \| \cdot \| \) is the Euclidean norm of a real vector \( x \), \( \otimes \) denotes the Kronecker product. We use vec \( (x_i, i \in \mathcal{N}) \) to denote the concatenated vector of \( x_i \) for all \( i \in \mathcal{N} \) and \( \text{Diag} \ (z) \) to denote the diagonal matrix with the coefficients of \( z \) along the diagonal. The projection operator of set \( \mathcal{B}^N \) is defined as:

\[
\text{proj}_{\mathcal{B}^N} \left( u_i^* \right) := \arg\min_{u_i \in \mathcal{B}^N} \frac{1}{2} \| u_i - u_i^* \|^2.
\]

Lastly, let \( f: \mathcal{C} \rightarrow (\mathcal{C}, +\infty) \) be a proper closed and convex function and \( \mathbf{W} \) be a positive definite matrix. Then the scaled proximal operator of \( f \) is given by

\[
\text{prox}^\mathbf{W} (z) := \arg\min_{y \in \mathcal{C}} \frac{1}{2} \| y - z \|^2 + \frac{1}{2} \| y \|^2 \mathbf{W},
\]

where \( \| \cdot \|_{\mathbf{W}} \) is the scaled norm induced by \( \mathbf{W} \), i.e., \( \| z \|_{\mathbf{W}} := \langle z, \mathbf{W} z \rangle \) for every \( z \) in \( \mathbb{R}^{(2N_i)+1} \).

**Synopsis:** Section II introduces the reformulated smooth constrained nonconvex minimization problem. The proposed SP-ADMM algorithms are presented in Section III. Section IV presents the theoretical results of the convergence conditions and convergence rate of the SP-ADMM algorithm. The performance of the proposed SP-ADMM algorithm is illustrated in Section V, and the conclusion is given in Section VI.

## II. PROBLEM FORMULATION

### A. Problem Statement

The wireless sensor network that we consider throughout this article is represented as an undirected and connected graph, \( \mathcal{G} = (\mathcal{N}, \mathcal{E}) \), and the topology is assumed to be known. The node set \( \mathcal{N} = \{1, 2, \ldots, N\} \) consists of \( N \) nodes, some of which are anchors with known unique positions in the set \( \mathcal{A} = \{a_{N-m+1}, \ldots, a_N\} \subset \mathcal{N} \). For each node \( i \in \mathcal{N} \), we define \( \mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\} \) as the set of adjacent nodes to node \( i \), and \( \mathcal{N}_i \) as the cardinality of \( \mathcal{N}_i \). The true position of node \( i \) is represented by \( p_i \in \mathbb{R}^n \) for \( i \in \mathcal{N} \), and the collection of all node positions is denoted by \( \mathbf{p} = \text{vec}(p_i, i \in \mathcal{N}) \in \mathbb{R}^{nN} \). The available noisy range measurement between node \( i \) and its adjacent node \( j \in \mathcal{N}_i \) is denoted as \( d_{i,j} \), and we assume that \( d_{i,j} = d_{j,i} \) following the convention in [3], [19]. Specifically, the noisy range measurement \( d_{i,j} \) can be expressed as in [29], [30].

\[
d_{i,j} = \| p_i - p_j \| + w_{i,j}, \ i \in \mathcal{N}, \ j \in \mathcal{N}_i,
\]

where \( w_{i,j} \) are the zero-mean, independent, and identically-distributed Gaussian measurement noise terms.

Using these notations, the maximum likelihood estimator, as our baseline, can be obtained through solving the following nonconvex constrained optimization problem (following [24]):

\[
\begin{align}
\arg\min_{p_i \in \mathbb{R}^n} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left( \| p_i - p_j \| - d_{i,j} \right)^2 \\
\text{subject to } p_k = a_k, \ \forall k \in \mathcal{A}.
\end{align}
\]

### B. Problem Reformulation

Our first step is to derive an equivalent smooth and constrained reformulation of problem (2), which provides the key insight toward algorithm design and its convergence of this article. Notice that the objective function in problem (2) can be written explicitly as

\[
\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \left[ \frac{1}{2} \| p_i - p_j \|^2 - d_{i,j} \| p_i - p_j \| + \frac{1}{2} d_{i,j}^2 \right].
\]

Obviously, there is a nonsmooth term in the objective function. Motivated by the recent works [21], [25], we apply the Cauchy-Schwarz inequality to obtain

\[
\| p_i - p_j \| = \max_{u_{i,j} \in \mathcal{B}} u_{i,j}^T (p_i - p_j),
\]

where \( \mathcal{B} := \{ x \in \mathbb{R}^n \mid \| x \| \leq 1 \} \) is a unit ball in \( \mathbb{R}^n \) with the center at the origin and \( u_{i,j} \in \mathbb{R}^n \) is an auxiliary variable, then we have

\[
-\| p_i - p_j \| = \min_{u_{i,j} \in \mathcal{B}} -u_{i,j}^T (p_i - p_j).
\]

Substituting (5) into (3), problem (2) is rewritten as a minimization problem over a ball constraint as follows

\[
\begin{align}
\arg\min_{p_i, u_{i,j}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \left[ \frac{1}{2} \| p_i - p_j \|^2 - d_{i,j} u_{i,j}^T (p_i - p_j) \right] \\
\text{subject to } u_i \in \mathcal{B}^N, \ \forall i \in \mathcal{N},
\end{align}
\]

where \( u := \text{vec}(u_i, i \in \mathcal{N}) \), \( u_i := \text{vec}(u_{i,j}, j \in \mathcal{N}_i) \), and \( \mathcal{B}^N := \{ x_{1,1}, x_{2,2}, \ldots, x_{N,N} \} \mid x_i \in \mathcal{B}, j = 1, \ldots, N_i \} \) is the Cartesian product of \( N_i \) balls \( \mathcal{B} \). To decouple the loss function of problem (6) over nodes, we introduce auxiliary variables that duplicate the positions of neighboring nodes. Specifically, we define \( z_{i,j}^+ \in \mathbb{R}^n \) as a copy of the position \( p_j \) for each neighbor \( j \in \mathcal{N}_i \), i.e.,

\[
z_{i,j}^+ := p_j, \ j \in \mathcal{N}_i.
\]
Similarly, we define $z_{i,j}^- \in \mathbb{R}^n$ as a copy of the position $p_i$ assigned to neighbor $j \in \mathcal{N}_i$, i.e.,

$$z_{i,j}^- := p_i, \quad j \in \mathcal{N}_i.$$  \hfill (8)

By collecting all variables associated with node $i$ together, we define a new variable $z_i$ as

$$z_i := \begin{bmatrix} p_i \\ z_{i,1}^- \\ \vdots \\ z_{i,n}^- \end{bmatrix} \in \mathbb{R}^{2n+1}, \quad i \in \mathcal{N}.$$  \hfill (9)

Here, $z_i^- := \text{vec}(z_{i,j}^-, j \in \mathcal{N}_i) \in \mathbb{R}^{n^2}$ is the collection of $N_i$ copies of the position $p_i$, and $z_i^+ := \text{vec}(z_{i,j}^+, j \in \mathcal{N}_i) \in \mathbb{R}^{n^2}$ represents the collection of all replicas of $p_j$ from neighboring node $j \in \mathcal{N}_i$. Using these notations, the objective function (6) can be written in a separable form as follows:

$$\sum_{i \in \mathcal{N}} \frac{1}{2} \|Q_i z_i\|^2 - u_i^T D_i Q_i z_i + \delta_{B_i}(u_i),$$  \hfill (10)

where $D_i := \text{Diag}(\text{vec}(d_{i,j}, j \in \mathcal{N}_i)) \otimes I_n \in \mathbb{R}^{n^2 \times n^2}$ is the measurement matrix of node $i$, and

$$Q_i := [I_{N_i}, O_{N_i}, -I_{N_i}] \otimes I_n \in \mathbb{R}^{n \times (1+2N_i)n}.$$  \hfill (11)

By using (9), we can rewrite (8) as a compact form

$$A_i z_i = 0,$$  \hfill (12)

where $A_i := [I_{N_i}, -I_{N_i}, O_{N_i}] \otimes I_n \in \mathbb{R}^{n \times (1+2N_i)n}$. Moreover, for a pair of connected sensors $i$ and $j$, we can deduce from equation (8) that $z_{i,j} = p_j$. Combining this equation with (7), we obtain an additional constraint that applies to all connected sensors:

$$z := \text{vec}(z_i, i \in \mathcal{N}) \in Z,$$

$$Z := \{ z | z_{i,j}^+ = p_j, j \in \mathcal{N}_i, \forall i \in \mathcal{N}, j \in \mathcal{N}_j \}. $$  \hfill (13)

Using (9), the set of anchors in (6c) can be expressed as

$$z \in \mathcal{X} := \{ z | E_i z_i = a_i, \forall i \in \mathcal{A} \},$$

$$E_i := [1, 0_{N_i}, 0_{N_i}] \otimes I_n \in \mathbb{R}^{n \times (1+2N_i)n}.$$  \hfill (14)

Finally, using (10), (12), and (13), the nonconvex optimization problem (6) can be equivalently reformulated into the compact form

$$\arg \min_{z,u} \sum_{i \in \mathcal{N}} \frac{1}{2} \|Q_i z_i\|^2 - u_i^T D_i Q_i z_i + \delta_{B_i}(u_i)$$

subject to $A_i z_i = 0, i \in \mathcal{N}$, \hfill (15a)

$$z \in \mathcal{X}, \quad z \in Z.$$  \hfill (15b)

To show these notations clearly, we give an example of a connected wireless sensor network in 2-D space in Fig. 1.

**Example 1:** Fig. 1 presents a wireless sensor network with 3 (inexact) positions of node $i = 1, 2, 3$ and one anchor of node $4$. If we only focus on node 2, on the one hand, since $\mathcal{N}_2 = \{1, 4\}$ and $\mathcal{Z}_{2,1}, \mathcal{Z}_{2,4}$ are the copies of $p_4$, then we have

$$p_2 = z_{2,1}^-, \quad p_2 = z_{2,4}^-.$$  \hfill (15c)

On the other hand, Fig. 1 presents that there exit the same partial elements in $z_{2,2}, z_{1,4}$, and $z_{4,4}$. Specifically, $p_1$ is the element of both $z_{1,4}$ and $z_{4,4}$, and $z_{4,4}$ also contain $p_4$. To enforce this trivial observation, we obtain the following constraint

$$z_{2,1}^+ = p_1 = z_{1,2}^-, \quad z_{2,4}^+ = p_4 = z_{4,2}^-,$$

which is part of the constraint set $Z$ defined in (13).

**Remark 1:** The objective function $F_i(z_i, u_i)$ in problem (15) is convex with respect to $z_i$ when $u_i$ is fixed, however, it is NOT a jointly convex function for $(z_i, u_i)$. In addition, one can observe that the objective function is separable but the linear constraint $Z$ is NOT separable in $z$. In Section III, we introduce a scaled proximal term to deal with this problem.

### III. PROPOSED SP-ADMM ALGORITHMS

In this section, we present the proposed SP-ADMM algorithms for solving the nonconvex and nonsmooth optimization problem derived in (15). The proposed SP-ADMM algorithms are built upon the ADMM method [25] developed for large-scale nonconvex problems.

First, we introduce the augmented Lagrangian (AL) function of problem (15) as follows:

$$L(z, u, \lambda) := \sum_{i \in \mathcal{N}} L_i (z_i, u_i, \lambda_i),$$

where

$$L_i (z_i, u_i, \lambda_i) := F_i (z_i, u_i) + \delta_{B_i}(u_i)$$

$$+ \langle \lambda_i, A_i z_i \rangle + \frac{c}{2} \|A_i z_i\|^2, \quad i \in \mathcal{N},$$

and $\lambda_i := \text{vec}(\lambda_{i,j}, j \in \mathcal{N}_i) \in \mathbb{R}^{n^2}$ correspond to the Lagrangian multipliers, $c > 0$ is a penalty coefficient. We apply the following scaled proximal ADMM updates [31] to obtain

$$z_i^{t+1} = \arg \min_{z \in Z} \sum_{i \in \mathcal{N}} L_i (z_i, u_i, \lambda_i) + \frac{c}{2} \|A_i z_i - z_i\|_{B_i}^2$$

$$u_i^{t+1} = \arg \min_u \sum_{i \in \mathcal{N}} L_i (z_i^{t+1}, u_i, \lambda_i) + \frac{\rho}{2} \|u_i - u_i^t\|^2,$$

$$\lambda_i^{t+1} = \lambda_i^t + cA_i z_i^{t+1}, \quad i \in \mathcal{N},$$

where $\rho > 0$ is a penalty coefficient and $t$ denotes the iteration step.
The AL function $L$ is separable in each of the variables $(z_i, u_i, \lambda_i)$, $i \in \mathcal{N}$ and $L_i$, $i \in \mathcal{N}$ is convex as a function separately of $(z_i, u_i)$ when the others are fixed. These features are key to distributed implementation. In addition, we remark that the (scaled) proximal term is critical in both the implementation efficiency and the theoretical analysis. It is used to ensure the following attractive properties:

- The sequence $\{ (z^*_i, u^*_i) \}$ has lower computational complexity.

- The proximal term controlled by the algorithm’s parameters for optimal performance in practice.

- The proof is shown in the [Section 1] of the supplementary material. [32]. From (30)–(32), one can see that $z^{t+1}$ can be obtained in a distributed fashion using only $z^{t+1}, j \in \mathcal{N}_t$ from its neighbors.

- The last point for $\mathbf{u}_t$ by omitting the terms that are irrelevant with $\mathbf{u}_t$, the optimization problem in (18) reduces to

$$
\arg\min_{\mathbf{u}_t} \sum_{i \in \mathcal{N}} \frac{1}{2} \| z_i - \mathbf{D}_i \mathbf{u}_t - \mathbf{A}_i^T \lambda_i + c \mathbf{B}_i^T \mathbf{B}_i z^t_i \|^2_{\mathbf{W}_i} + \rho \| \mathbf{u}_t - \mathbf{u}_t \|^2_{\mathbf{W}_i}.
$$

Optimization problem (24) presents two challenges. Firstly, obtaining an analytic solution is difficult due to the linear constraints of both $\mathcal{Z}$ and $\mathcal{X}$. Secondly, calculating the inverse matrix $\mathbf{W}_i$ in (23) to obtain $\tilde{z}^{t+1}$ can be computationally expensive. In the following, we demonstrate that an appropriate choice of $\mathbf{B}_i^T \mathbf{B}_i$ can overcome these challenges. To facilitate the calculation of the inverse of $\mathbf{W}_i$, we set

$$
c \mathbf{B}_i^T \mathbf{B}_i = c |\mathbf{A}_i^T \lambda_i| + |\mathbf{Q}_i^T \mathbf{Q}_i|,
$$

where $| \cdot |$ takes element-wise absolute value of a matrix. From the definition of $\mathbf{A}_i$ and $\mathbf{Q}_i$ in (12) and (11) respectively, we get

$$
\mathbf{A}_i^T \mathbf{A}_i = \begin{bmatrix}
0_{N_i} & N_i^T \\
N_i & 0_{N_i}
\end{bmatrix} \otimes I_n,
$$

$$
\mathbf{Q}_i^T \mathbf{Q}_i = \begin{bmatrix}
0_{N_i} & N_i^T \\
N_i & 0_{N_i}
\end{bmatrix} \otimes I_n.
$$

Substituting (26) and (27) into (25), we obtain

$$
c \mathbf{B}_i^T \mathbf{B}_i = \begin{bmatrix}
(c + 1) N_i & c \cdot I_{N_i}^T \\
c \cdot I_{N_i} & c \cdot I_{N_i} \otimes I_n
\end{bmatrix} \otimes I_n.
$$

With the aid of (26)–(28), we can guarantee that $\mathbf{W}_i$ defined in (21) is a positive definite diagonal matrix of the form

$$
\mathbf{W}_i = 2 \cdot \text{Diag} \left( \left[ (c + 1) N_i, c \cdot I_{N_i}^T, I_{N_i} \right] \right) \otimes I_n.
$$

It makes $\tilde{z}_t^{t+1}$ in (23) easy to compute. Moreover, by denoting $\tilde{z}_i := (\tilde{p}_i)^T, (\tilde{z}_i^+)^T, (\tilde{z}_i^-)^T\}$, where its last two parts are

$$
\tilde{z}_i^- := \text{vec}(\tilde{z}_{i,j}, j \in \mathcal{N}_i), \tilde{z}_i^+ := \text{vec}(\tilde{z}_{i,j}, j \in \mathcal{N}_i),
$$

then $z^{t+1}$ can be deduced using the following remark.

Remark 2: Given $\mathbf{B}_i, i \in \mathcal{N}$ in (25), the analytic formula of the optimal solution $z^{t+1}$ in (24) is derived as follows

$$
\mathbf{p}_t^{t+1} = \tilde{p}_t^{t+1},
$$

$$
(z_i^-)^{t+1} = \text{vec} \left( \frac{c}{c + 1} (\tilde{z}_{i,j}]^t, \frac{1}{c + 1} (\tilde{z}_{i,j}]^t, j \in \mathcal{N}_i) \right),
$$

$$
(z_i^+)^{t+1} = \text{vec} \left( \frac{c}{c + 1} (\tilde{z}_{i,j}]^t, \frac{c}{c + 1} (\tilde{z}_{i,j}]^t, j \in \mathcal{N}_i) \right).
$$

The proof is shown in the [Section 1] of the supplementary material. From (30)–(32), one can see that $z^{t+1}$ can be obtained in a distributed fashion using only $\tilde{z}^{t+1}, j \in \mathcal{N}_t$ from its neighbors.

To see the last point for $\mathbf{u}_t$ by omitting the terms that are irrelevant with $\mathbf{u}_t$, the optimization problem in (18) reduces to

$$
\arg\min_{\mathbf{u}_t} \sum_{i \in \mathcal{N}} \left( -\mathbf{Q}_i^T \mathbf{D}_i \mathbf{u}_t - \mathbf{z}_i^{t+1} + \frac{\rho}{2} \| \mathbf{u}_t - \mathbf{u}_t \|^2_{\mathbf{W}_i} \right).
$$

The above objective function is separable in $\mathbf{u}_t$ and further performing projection yields

$$
\mathbf{u}_t^{t+1} = \text{proj}_{\mathcal{B}_N} \left( \mathbf{u}_t^t + \frac{1}{\rho} \mathbf{D}_i \mathbf{Q}_i \mathbf{z}_i^{t+1} \right),
$$

which represents a projection onto the unit ball that can be reduced to

$$
\mathbf{u}_t^{t+1} = \mathbf{u}_t^{t+1} \left( \max\left\{ 1, \| \mathbf{u}_t^{t+1} \|^2 \right\}, j \in \mathcal{N}_t,
$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Algorithm 1: Distributed SP-ADMM algorithm

Input: parameter $c, \rho$, and initial values of $z_0^i, u_0^i, \lambda_0^i = 0$, $i = 1, \ldots, N$.

Let $W_i = 2 \cdot \text{Diag} \left( \left[ (c+1)N_i, c \cdot 1_{N_i}, 1_{N_i}^T \right] \right) \otimes I_n$.

for $t \in \{0, \ldots, T\}$ do

for client $i = 1, 2, \ldots, N$ in parallel do

\[ \dot{z}_{i}^{t+1} = W_i^{-1} \left( Q_i^T D_i u_i^t - A_i^T \lambda_i^t + cB_i^T B_i z_i^t \right) \]  
\[ \hat{p}_{i}^{t+1} = a_i, \text{ if } i \in A \]

end

communication:

⇒ Broadcast $(\dot{z}_{i}^{t+1}, (\dot{z}_{j}^{t+1}))$ to region $j \in N_i$.
⇐ Receive $(\dot{z}_{i}^{t+1}, (\dot{z}_{j}^{t+1}))$ from region $j \in N_i$.

for client $i = 1, 2, \ldots, N$ in parallel do

Update $z_{i}^{t+1}$ via (30)–(32)

Update $u_{i}^{t+1}$ via (34)

$\lambda_{i}^{t+1} = \lambda_{i}^{t} + cA_{i}z_{i}^{t+1}$

end

end

where

\[ \text{vec} \left( u_{i,j}^{t+1}, j \in N_i \right) = u_{i}^{t+1} + \frac{1}{\rho} D_i Q_i z_{i}^{t+1}. \]

In summary, problems (17) and (18) can be equivalently rewritten as problems (24) and (33), respectively, and the corresponding distributed solutions are given in (30)–(32) and (34). An outline of the distributed procedure is described in Algorithm 1.

Remark 3: We show that the SP-ADMM algorithm is closely related to the ADMM algorithm derived in [25] for single source localization. Specifically, let us set $p_i = p_i^i$, $a_j = p_j^i$ for all $i \geq 0$, $j \in N_i = A$ (single source localization). Then, by (30) and the definition of $E_i$ in (14), we obtain

\[ p_i^{t+1} = \hat{p}_{i}^{t+1} = E_i z_{i}^{t+1}, \ i \notin A. \]  

Multiplying $E_i$ on both sides of (23) and rearranging terms yields

\[ \dot{p}_{i}^{t+1} = \sum_{j \in N_i} \frac{1}{2(c+1)N_i} \left[ d_{ij} u_{i,j} + (2c+1) \left( p_i^t - p_j^t \right) - \lambda_{i,j}^t + (2c+1) p_j^t + (z_{j}^t - c(p_i^t - (z_{j}^t)^T) \right]. \]  

If $p_i^t = (z_{j}^t)^T$, which is $A_i z_{i}^t = 0$, $i \in N_i$, then combining (39) with (40), replacing $p_j$, $z_{j}^t$ with $a_j$, and rearranging terms again we have

\[ p_i^{t+1} = \frac{1}{m} \sum_{j=1}^{m} \left[ a_j + \frac{1}{2(c+1)} \left( d_{ij} u_{i,j}^t + (2c+1) \left( p_i^t - a_j \right) - \lambda_{i,j}^t \right] \right. \]

Here we have replaced $d_{i,j}, u_{i,j}^t, \lambda_{i,j}^t$ by $d_j, u_j^t, \lambda_j$ since $i$ is a fixed value in the single source localization problem. Hence, if $z_i^t$ satisfies the linear equality constraint (15b), then equation (41) is identical to the ADMM algorithm given in [25, Equation (3.8)] except for the constant coefficients of the variables. Therefore, the proposed SP-ADMM algorithm can be regarded as an extended version of the ADMM algorithm in [25] with extra capability to handle the multi-source localization problems.

However, in the case of multi-source localization, the ADMM algorithm can be significantly more complex than in single-source localization due to coupled variables and increased computational complexity. Coupled variables make subproblems in ADMM entangled and thus the convergence of the algorithm is hard to prove. Moreover, the increased number of variables in multi-source localization increases computational complexity, leading to slower convergence rates and reduced localization accuracy.

In Algorithm 1, we observed that in order to carry out the new round of iterations, node $i$ has to store the following values: $z_i^t, u_i^t, \lambda_i^t, W_i, B_i^T B_i, D_i, Q_i, A_i, c, \rho, N_i$. If we assume that the storage unit occupied by any real number is one, then node $i$ requires $13n^2N_i^2 + 10n^2N_i + 2n^2 + 4nN_i + n + 3$ storage units in total. Note that although the matrices and parameters involved are fixed, the rest of the vectors are updated with each iteration. To reduce the storage space required by each sensor to run Algorithm 1, we transform the update step as follows.

Let us start by giving an explicit formula for the parts of the update variables in Algorithm 1. Using the form of $A_i, Q_i$, and $D_i$, defined in (12), (11), and (10), respectively, we have

\[ A_i^T \lambda_i^t = \left[ \sum_{j \in N_i} (\lambda_{i,j}^t)^T, \text{vec} \left( -\lambda_{i,j}^t, j \in N_i \right) \right]^T, \]

\[ Q_i^T D_i u_i^t = \left[ \sum_{j \in N_i} (d_{i,j} u_{i,j}^t)^T, \text{vec} \left( -d_{i,j} u_{i,j}^t, j \in N_i \right) \right]^T. \]

Also, it follows from the definition (28) that

\[ cB_i^T B_i z_i^t = \left[ \left( (c+1)N_i p_i^t + c \sum_{j \in N_i} (z_{i,j}^t)^T + \sum_{j \in N_i} (z_{i,j}^t)^T \right)^T, \right. \]

By substituting (42) and (43) into (35) yields the following update

\[ p_i^{t+1} = \frac{1}{2(c+1)} \sum_{j \in N_i} \left[ d_{i,j} u_{i,j} - \lambda_{i,j}^t \right. \]

\[ + c \left( p_i^t + (z_{i,j}^t)^T \right) + p_i^t + (z_{i,j}^t)^T \right], \]

\[ (z_{i,j}^t)^{t+1} = \frac{1}{2c} \lambda_{i,j}^t + \frac{1}{2} \left( p_i^t + (z_{i,j}^t)^T \right), \]

\[ (z_{i,j}^t)^{t+1} = -\frac{1}{2} d_{i,j} u_{i,j}^t + \frac{1}{2} \left( p_i^t + (z_{i,j}^t)^T \right). \]
Using once again the form of $Q_i$, $D_i$ and $A_i$, it follows from the (34) and (38) that
\[
\begin{align*}
\hat{u}_{i,j}^{t+1} &= u_{i,j}^t + \frac{d_{i,j}}{\rho} (p_{i,j}^t - (z_{i,j}^-)^{t+1}), \\
u_{i,j}^{t+1} &= \max\{1, \|u_{i,j}^{t+1}\|\} u_{i,j}^{t+1},
\end{align*}
\]
(47)
\[
\lambda_{i,j}^{t+1} = \lambda_{i,j}^t + c (p_{i,j}^t - (z_{i,j}^-)^{t+1}).
\] (48)

Next, we intend to remove $\tilde{z}_{i,j}^{t+1}$ to reduce the storage space and computation required by Algorithm 1. Combining (30) with (36) and (44), we have
\[
p_i^{t+1} = \left\{ \begin{array}{ll}
\sum_{j \in N_i} (2d_{i,j} u_{i,j}^t - 2\lambda_{i,j}^t + \alpha_{i,j}^t + \beta_{i,j}^t), & \text{if } i \notin A, \\
a_i, & \text{if } i \in A.
\end{array} \right.
\] (49)

where
\[
\alpha_{i,j}^t := \lambda_{i,j}^t + c (p_{i,j}^t + (z_{i,j}^-)^t),
\] (50)
\[
\beta_{i,j}^t := -d_{i,j} u_{i,j}^t + p_{i}^t + (z_{i,j}^-)^t.
\] (51)

By adding $\frac{1}{c} (\tilde{z}_{i,j}^{t+1})^2$ to the both side of (45) and using (31), we get
\[
\frac{c+1}{c} (z_{i,j}^-)^{t+1} = \frac{1}{2c} \lambda_{i,j}^t + \frac{1}{2} (p_{i,j}^t + (z_{i,j}^-)^t) + \frac{1}{c} (\tilde{z}_{i,j}^{t+1})^2.
\] (52)

Then, substituting (46) into (52) and using (50) and (51), it yields
\[
(z_{i,j}^-)^{t+1} = \frac{1}{2(c+1)} (\alpha_{i,j}^t + \beta_{i,j}^t).
\] (53)

Similarly, we add $c(z_{i,j}^-)^{t+1}$ to (46), then using (32) and (50) and (51) to obtain
\[
(z_{i,j}^-)^{t+1} = \frac{1}{2(c+1)} (\beta_{i,j}^t + \alpha_{i,j}^t).
\] (54)

Applying (54) to (47) and (51), we get
\[
\begin{align*}
\hat{u}_{i,j}^{t+1} &= u_{i,j}^t + \frac{d_{i,j}}{\rho} p_{i,j}^t - \frac{d_{i,j}}{2\rho (c+1)} (\beta_{i,j}^t + \alpha_{i,j}^t), \\
u_{i,j}^{t+1} &= \max\{1, \|u_{i,j}^{t+1}\|\} u_{i,j}^{t+1},
\end{align*}
\]
(55)
\[
\beta_{i,j}^{t+1} = -d_{i,j} u_{i,j}^{t+1} + p_{i}^{t+1} + \frac{1}{2(c+1)} (\beta_{i,j}^t + \alpha_{i,j}^t).
\] (56)

Substituting (53) into (48) and (50), we have
\[
\begin{align*}
\lambda_{i,j}^{t+1} &= \lambda_{i,j}^t + c p_{i,j}^{t+1} - \frac{c}{2(c+1)} (\alpha_{i,j}^t + \beta_{i,j}^t), \\
\alpha_{i,j}^{t+1} &= \alpha_{i,j}^t + 2c p_{i,j}^{t+1}.
\end{align*}
\] (57)
(58)

The simplified version of the algorithm is summarized in Algorithm 2. From the updating steps of Algorithm 2, it can be seen that each node only needs to store the following values: $\alpha_{i,j}^t, \beta_{i,j}^t, \hat{u}_{i,j}^t, \lambda_{i,j}^t, d_{i,j}, j \in N_i$ and $c, \rho, N_i$. Compared with the storage requirement by node $i$ of Algorithm 1, Algorithm 2 demands only $4nN_i + N_i + 3$ storage units, which is significantly reduced compared to that required by Algorithm 1. We remark that the update (49) and (55)-(58) of Algorithm 2 should be interpreted as an improved version that optimizes the storage requirement of Algorithm 1. Although Algorithm 2 offers superior storage efficiency, the presentation of Algorithm 1 still has value for several reasons. Firstly, Algorithm 1 serves as a foundation for understanding the principles and techniques behind Algorithm 2. By presenting Algorithm 1 first, we can demonstrate the motivation and rationale for the modifications made in Algorithm 2. Additionally, the convergence analysis for Algorithm 1 is more direct since Algorithm 2 streamlines the steps of Algorithm 1.

Remark 4: (Complexity Comparison) For each iteration of Algorithm 1, the computation cost is derived by (35) that involves the inversion of diagonal matrix $W_i \in \mathbb{R}^{(2nN_i+1) \times n(2nN_i+1)}$, which is independent of the iteration index $t$, and thus needs to be computed only once. Besides, equation (34) requires calculating the Euclidean norm of $u_i \in \mathbb{R}^{nN_i \times 1}$. Hence, the computational complexity of the proposed SP-ADMM algorithm is $O(nN_i)$, which is lower than SDP relaxation method $O(n^3)$ and ADMM-H, but equal to SF and AM-FD. The communication cost per iteration per node is directly proportional to the number of scalar variables that sensors need to transmit to their neighboring nodes [9]. Thus, the AM-FD, SDP, and SF algorithms have a communication cost of $nN_i$, while the proposed SP-ADMM requires sending two variables to its neighbor, resulting in a communication cost per iteration of $2nN_i$. It should be noted that the proposed SP-ADMM has lower communication cost than ADMM-H since the latter also needs to transmit a penalty parameter to its neighboring nodes at each iteration. The detailed comparisons between the proposed SP-ADMM algorithm with the other methods have been shown in Table I.

IV. CONVERGENCE ANALYSIS

In this section, we present a proof for the convergence of the proposed Algorithm 1, which consists of three main steps. First, we provide an upper bound [33] for the difference $\mathcal{L}(z_{i,j}^{t+1}, u_{i,j}^{t+1}, \lambda_{i,j}^{t+1}) - \mathcal{L}(z_i, u_i, \lambda_i).$ Then, we construct
a potential function $\zeta^t$ based on the AL function $L(z^t, u^t, \lambda^t)$, which monotonically decreases with each iteration and has a lower bound. Employing the Kurdyka–Łojasiewicz property [34], [35], [36], [37], [38], we establish the global convergence of the sequence generated by Algorithm 1. Next, we introduce the optimal gap function $F(z, u, \lambda)$ to show that the proposed Algorithm 1 converges to both a KKT point of problem (15) and a critical point of the original problem (2). Lastly, we define the $\varepsilon$-solution and prove that the algorithm sequence converges with a sublinear rate.

First, we bound the update step of the AL function $L(z, u, \lambda)$ at each iteration.

**Lemma 1:** Suppose $cB_i^T B_i$ takes the form of (25), and let $\{ (z^t_i, u^t_i, \lambda^t_i) \}$ be a sequence generated by Algorithm 1. Then, for all $t \geq 1$, we have

$$
L(z^{t+1}, \lambda^{t+1}, u^{t+1}) - L(z^t, \lambda^t, u^t) \\
\leq \sum_{i \in N} \left[ -\frac{c}{2} \| z^{t+1}_i - z^t_i \|^2_{B_i^T B_i} - \frac{\rho}{2} \| u^{t+1}_i - u^t_i \|^2 + \frac{3(N_{\text{max}} + 1)}{c} \| Q_i (z^{t+1}_i - z^t_i) - D_i (u^{t+1}_i - u^t_i) \|^2 \\
+ \frac{3c}{2} \| z^{t+1}_i - z^t_i - (z^t_i - z^{t-1}_i) \|^2_{B_i^T B_i} \\
+ \frac{3(1 + c) (1 + N_{\text{max}})}{2} \| (z^{t+1}_i - z^t_i - (z^t_i - z^{t-1}_i)) \|^2_{B_i^T B_i} \right],
$$

where $N_{\text{max}} := \max\{ N_i, i \in N \}$.

**Proof:** See the [Section 2] of the supplement [32].

Obviously, the right-hand-side (rhs) of (59) is a sum of three positive terms, no matter how large $c$ and $\rho$ are, there is no guarantee that $L(z, u, \lambda)$ decreases at each iteration step. Thus, it cannot be used as a potential function. In search for an appropriate potential function, we add the constraint violation and the proximal term to the AL function, then a novel potential function is designed as follows

$$
\zeta^t = \sum_{i \in N} \left[ -\frac{c}{2} \| z^{t+1}_i - z^t_i \|^2_{B_i^T B_i} - \frac{\rho}{2} \| u^{t+1}_i - u^t_i \|^2 + \frac{\kappa_1 - 1}{2} \| z^{t+1}_i - z^t_i \|^2_Q_i Q_i \\
- \frac{c(\kappa_1 - 1)}{2} \| z^{t+1}_i - z^t_i \|^2_{A_i^T A_i} - \frac{\rho}{4} \| u^{t+1}_i - u^t_i \|^2 \\
- \frac{c(\kappa_1 - 6(N_{\text{max}} + 1))}{2} \| z^{t+1}_i - z^t_i - (z^t_i - z^{t-1}_i) \|^2_{A_i^T A_i} \\
- \frac{c}{2} \| Q_i (z^{t+1}_i - z^t_i) - D_i (u^{t+1}_i - u^t_i) \|^2 \right] \\
- \frac{ck_1 - 6 (1 + c) (N_{\text{max}} + 1)}{2} \| (z^{t+1}_i - z^t_i - (z^t_i - z^{t-1}_i)) \|^2_{B_i^T B_i} \\
- \left( \frac{ck_2 \cdot \bar{\varepsilon}_\text{min} (c + 1)}{2N_{\text{sum}} n (c + 1)^2} \right) \| z^{t+1}_i - z^t_i \|^2 \\
- \left( \frac{\rho}{4} - d^2_{\text{max}} (\kappa_1 + \kappa_2) \right) \| u^{t+1}_i - u^t_i \|^2 \right],
$$

(61)

where $N_{\text{sum}} := \sum_{i \in N} N_i$ is the total number of neighboring nodes, $d_{\text{max}} := \max\{ d_{ij}, i \in N_j, j \in N_i \}$ is the maximum measurement distance and $\bar{\varepsilon}_\text{min} := \min\{ (c + 1)^2 N_i^2 + c^2 N_i^2, N_i, i \in N \}$.

**Proof:** See the [Section 3] of the supplement [32].

From the above analysis, it is clear that as long as $\kappa_1, \kappa_2$ and $\rho$ are sufficiently large, the rhs of (61) is less than zero. As such, the potential function $\zeta^t$ decreases at each iteration of SPADMM. Below, we derive the precise bounds for $\kappa_1, \kappa_2$ and $\rho$. First, following the first five rows on the rhs of (61), a sufficient condition for $\kappa_1$ is given below (for any given $c > 0$)

$$
\kappa_1 \geq 6 (N_{\text{max}} + 1) \left(1 + \frac{1}{c}\right).
$$

(62)

Second, for any given $c$ and $\kappa_1$, according to the sixth row on the rhs of (61), we need

$$
\kappa_2 \geq \frac{N_{\text{sum}} n (c + 1)^2 (N_{\text{max}} + 1) \kappa_1}{\bar{\varepsilon}_\text{min}}.
$$

(63)

Finally, given $c, \kappa_1, \kappa_2$, and based on the last row on the rhs of (61), parameter $\rho$ requires to satisfy

$$
\rho \geq 4d_{\text{max}}^2 (\kappa_1 + \kappa_2).
$$

(64)

We conclude that if (62)–(64) are satisfied, then the potential function $\zeta^t$ will decrease at every iteration.

Now, we are ready to establish the main result. To this end, let us define the function $F(z^t, u^t, \lambda^t)$ as the optimal gap of problem (15) given by

$$
F(z^t, u^t, \lambda^t) = \sum_{i \in N} \left[ \| z^t_i - \text{proj}_{X_i} (z^t_i - (\nabla_{z_i} F_i (z^t_i, u^t_i) + A_i^T \lambda_i)) \|^2 \\
+ \| A_i z_i \|^2 + \| u^t_i - u^t_i \|^2 \right].
$$

(65)

Based on the function $F(z^t, u^t, \lambda^t)$ defined above, Lemma 3 establishes its relationship with a KKT point of problem (15) and a critical point of problem (6).

**Lemma 3:** When $F(z^t, u^t, \lambda^t) = 0$, then $(z^t, u^t, \lambda^t)$ is a KKT solution of problem (15) satisfying:

$$
z^t \in \arg \min_{z \in Z} \sum_{z \in N} F_i (z, u^t_i) + (\lambda^t_i, A_i z_i),
$$

(66a)

$$
0 \in \nabla_{u_i} F_i (z^t_i, u^t_i) + \partial \delta_{B_i \kappa_i} (u^t_i),
$$

(66b)

$$
0 = A_i z_i.
$$

(66c)

Moreover, $(p^t, u^t)$ is a critical point of problem (6), and $p^t$ is a critical point of the original problem (2).

**Proof:** See the [Section 4] of the supplement [32].

**Lemma 3** suggests that we can demonstrate the sublinear convergence of Algorithm 1 to a critical point of the original
Definition 1: Given problem (15), we define \((z, u, \lambda)\) as an \(\epsilon\)-solution if \(\|A_z z_t \| < \epsilon\) and there exists a vector \(v_1 \in \nabla z F_i (z_i, u_i) + A_z \lambda_t + \partial \delta_{X \times Z} (z_t)\) and a vector \(v_2 \in \nabla z F_i (z_i, u_i) + \partial \delta_{\mathcal{G}_N} (u_i)\) such that \(\|v_1\| \leq \epsilon, \|v_2\| \leq \epsilon\).

Lemma 4: Let parameters \(c\) and \(\rho\) satisfy (62)–(64), then for any \(t > 1\), there exists a constant \(M > 0\) such that we can find an \(s \in \{1, 2, ..., t - 1\}\) satisfying \((z_{i+1}, u_{i+1}, \lambda_{i+1})\) being a \(M/\sqrt{t-1}\)-solution. This implies that an \(\epsilon\)-solution can be obtained within \(M^2/\epsilon^2\) iterations.

Proof: See the [Section 5] of the supplement [32].

Theorem 1: For each nodes \(i \in \mathcal{N}\), suppose the sequence \(\{(z_t^i, u_t^i, \lambda_t^i)\}\) is generated by Algorithm 1 and the conditions (62)–(64) are satisfied by properly setting the parameters. Then we have

- **Lower Bound:**
  \[ \exists \zeta > -\infty \text{ s.t. } c^t \geq \zeta, \forall t > 0. \]

- **Eventual Consensus:**
  \[ \lim_{t \to \infty} z_{i+1}^t - z_i \to 0, \quad \lim_{t \to \infty} \lambda_{i+1}^t - \lambda_i \to 0, \quad \lim_{t \to \infty} u_{i+1}^t - u_i \to 0, \quad \lim_{t \to \infty} A_z z_t \to 0. \]

- **Global Convergence:** The sequence \(\{y^t = (z_t^i, u_t^i, \lambda_t^i)\}\) has a finite length, i.e., \(\sum_{t=1}^{\infty} \|y_{i+1}^t - y_i\| < \infty\), and it converges to a KKT point of problem (15).

- **Convergence to Stationary Points:** \(F(z^t, u^t, \lambda^t) \to 0\) and the iteration sequence \(\{(z_t^i, u_t^i, \lambda_t^i)\}\) converges to a KKT stationary point of problem (15).

- **Sublinear Convergence Rate:** For any given \(\epsilon > 0\), suppose that \(F(z^t, u^t, \lambda^t)\) in (65) is less than \(\epsilon\) for the first time in the \(T\)-th iteration step, i.e.,
  \[ T := \arg \min_{t} F(z^t, u^t, \lambda^t) \leq \epsilon. \]

Then there exists a positive constant \(\epsilon_2 > 0\) (the specific form is defined in the [S.119]) of the supplementary material [32]) such that \(\epsilon_1 \leq \epsilon_2\), which means the convergence rate of function \(F(z^t, u^t, \lambda^t)\) is \(O(1/T)\).

Proof: See the [Section 6] of the supplement [32].

To our knowledge, Theorem 1 is the first result that shows the \(O(1/T)\) convergence rate of the distributed ADMM algorithm for the nonconvex and nonsmooth localization problem. Since Algorithm 2 is a simplified version of Algorithm 1 in Section III, the convergence result also holds for Algorithm 2. In the next section, numerical results will corroborate that the SP-ADMM algorithms can demonstrate more favorable convergence behavior than the benchmark methods.

In the next section, numerical results will corroborate that the SP-ADMM algorithms can demonstrate more favorable convergence behavior than the benchmark methods.

3Given the linear constraint and nonsmooth term in the nonconvex optimization problem (15), we refer to the points that satisfy the KKT condition as the stationary points (or KKT points) of the problem (15), such as [27], [39], [40], [41], [42].

V. NUMERICAL RESULTS

We consider the two-dimensional sensor network localization problem and evaluate the performance of the proposed Algorithm 1 against several other state-of-the-art methods, including SDP [10], SF [19], AM-FD [21] and ADMM-H [20]. We remark that the proposed Algorithm 2 only simplifies the storage space of Algorithm 1. Therefore, there is no difference in positioning accuracy between Algorithm 1 and Algorithm 2. Note also that the first two methods employ convex relaxation, the third one solves the original nonconvex problem directly, and the last one is a hybrid convex/nonconvex solver. Following [21], we will also examine the performance of SP-ADMM-NAG50 and AM-FD-NAG50, where SP-ADMM-NAG50 means that we run 50 steps of the NAG method [43] to obtain a good initial point, after that, the proposed SP-ADMM is used until convergence. For fairness, AM-FD-NAG50 uses the same initialization.

The criteria under which we compare the above algorithms are the practical running time counted as the maximal computation time among all parallel computing components and the averaged root mean squared error (RMSE) in a particular iteration, as described in [20], [24].

\[ \text{RMSE}(t) = \sqrt{\frac{\sum_{i \in \mathcal{N} \setminus \mathcal{A}} \|p_i^t - p_i\|^2}{N - m}}. \]

where \(\mathcal{N} / \mathcal{A}\) represents the set of agents with unknown positions, which has \(N - m\) elements, \(p_i\) is the true position of node \(i\), and \(p_i^t\) is the estimated position of node \(i\) in the \(t\)-th iteration of the algorithm. In addition, to check the convergence of the proposed algorithm as stated in Theorem 1, let

\[ S(t) = \sum_{i \in \mathcal{N}} \|\nabla z_i F_i (z_i^t, u_i^t) + A_z^T \lambda_i^t\|^2, \]

\[ U(t) = \sum_{i \in \mathcal{N}} \|u_i^t - u_{i-1}^t\|^2, \]

\[ P(t) = \sum_{i \in \mathcal{N}} \|A_z z_i\|^2, \]

where \(S(t)\) and \(U(t)\) are defined as the stationarity gap while \(P(t)\) is defined as the feasibility gap for problem (15). The MATLAB code that implements the proposed SP-ADMM is available at: https://github.com/zm-stu/SP-ADMM.

A. Benchmark Network

In the experiment, various methods are applied to the benchmark network data collected by the Stanford Computational Imaging Lab [44]. Table II lists the details of the two

| Network Parameters | Method Parameters |
|--------------------|-------------------|
| \(N\) | \(m\) | \(C_{\text{range}}\) | \(\sigma_{\text{std}}\) | \(D_{\text{avg}}\) | \(\epsilon\) | \(\zeta\) | \(\tau\) | \(\theta\) | \(\delta\) | \(\lambda_{\text{max}}\) | \(u^0\) |
|------|------|-------|--------|--------|--------|-------|-------|-------|-------|-------|--------|-------|
| 500  | 10   | 0.3   | 0.02   | 14.15  | 0.002  | 0.25  | 0.008 | 0.98  | 1.01  | 0.003  | 0.002  | 0.98  | 1.01  | 0.003  |
| 1000 | 20   | 0.1   | 0.007  | 11.09  | 0.003  | 0.07  | 0.002 | 0.98  | 1.01  | 0.003  | 0.002  | 0.98  | 1.01  | 0.003  |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
networks and the specific parameters used for both the ADMM-H method and AM-FD method, which refer to the settings in [20] and [21]. $C_{\text{range}}$ and $D_{\text{avg}} := \frac{1}{N} \sum_{i \in N} N_i$ in Table II represent the communication range and the average number of neighbors, respectively. For the proposed algorithm, parameters are set to $\rho = c = 0.11$, $\mathbf{u}_0 = 0$ for $N = 500$ networks and $\rho = c = 0.0197$, $\mathbf{u}_0 = 0$ for $N = 1000$ networks. The initial point $\mathbf{z}^0$ for both networks is selected from a uniform distribution, \textcolor{red}{\text{Unif}} $(-1, 1)^{(n(4|E|+N)}$. Following [15], we consider two different kinds of measurement noise: one is an additive white Gaussian noise (AWGN) with standard deviation $\sigma_i^2 = \sigma_{\text{add}} \| p_i - p_j \|^2$ (second row). The purpose of including the range-dependent Gaussian noise experiment is to evaluate the robustness of our proposed algorithm.

Convergence performance: Let us first examine the convergence behaviors of the proposed algorithm, SDP [10], SF [19], AM-FD [21], and ADMM-H [20]. Fig. 2(a) and 2(c) display the RMSE versus the iteration number when we use AWGN and range-dependent Gaussian noise, respectively. As shown in Fig. 2(a) that ADMM-H based on a two-stage approach performs the best and its RMSE approaches the Cramer–Rao Lower Bound (CRLB) [45], but it is less advantageous in terms of the running time (see Table III). We note that the AM-FD-NAG50 and SP-ADMM-NAG50 show lower RMSE than both SF and SDP, illustrating that non-relaxed problems generally result in better location estimation than that of the relaxed problems. When we focus on the first-order methods, Fig. 2(a) and 2(c) demonstrates that the SP-ADMM-NAG50 algorithm achieves higher accuracy with fewer iteration steps than other methods for both AWGN and range-dependent Gaussian noise scenarios. Furthermore, we see that AM-FD-NAG50 outperforms AM-FD, which reflects the advantage of using the NAG method to provide an initial value that is beneficial for the nonconvex localization problem (2).

In Fig. 3, we plot the RMSE versus the iteration number for a large network with $(N = 1000, m = 20)$, where $\sigma_i^2 = \sigma_{\text{add}} \| p_i - p_j \|^2$ (second row). The purpose of including the range-dependent Gaussian noise experiment is to evaluate the robustness of our proposed algorithm.

Fig. 2. Performance with benchmark network ($N = 500$, $m = 10$). Measurement noise: AWGN (first row) and range-dependent Gaussian noise $\sigma^2_{i,j} = \sigma_{\text{add}} \| p_i - p_j \|^2$ (second row). RMSE value (left column); feasibility gap and stationarity gap (right column).

Fig. 3. Performance with benchmark network ($N = 1000$, $m = 20$). Measurement noise: AWGN (first row) and range-dependent Gaussian noise $\sigma^2_{i,j} = \sigma_{\text{add}} \| p_i - p_j \|^2$ (second row). RMSE value (left column); feasibility gap and stationarity gap (right column).

**TABLE III**

| Algorithm | RMSE (AWGN) | Run time (seconds) |
|-----------|-------------|--------------------|
| SF        | 9.85e02     | 0.05401            |
| AM-FD     | 4.99e-02    | -                  |
| AM-FD-NAG50 | 2.75e-02    | -                  |
| SDP       | 2.97e-02    | -                  |
| ADMM-H    | 1.49e-02    | 0.1518             |
| SP-ADMM-NAG50 | 2.03e-02 | 0.01091          |
| SP-ADMM   | 2.98e-02    | 0.01784            |

| Benchmark Network ($N = 500$, step = 1000, CRLB $\approx 0.014$) | COMPARISONS OF RUNNING TIME |
|---------------------------------------------------------------|-----------------------------|
| Algorithm | RMSE (AWGN) | Run time (seconds) |
|-----------|-------------|--------------------|
| SF        | 9.85e02     | 0.05401            |
| AM-FD     | 4.99e-02    | -                  |
| AM-FD-NAG50 | 2.75e-02    | -                  |
| SDP       | 2.97e-02    | -                  |
| ADMM-H    | 1.49e-02    | 0.1518             |
| SP-ADMM-NAG50 | 2.03e-02 | 0.01091          |
| SP-ADMM   | 2.98e-02    | 0.01784            |

* These symbols "-" in the table mean that the methods cannot be implemented in parallel.
Gap: To further examine the convergence of the SP-ADMM algorithm, we plot in Fig. 2(b) (Fig. 2d) and Fig. 3(b) (Fig. 3d) the curves of the optimal gap of problem (15) versus the iteration number. Observe explicitly that the performance gap of our proposed SP-ADMM algorithm reduces as the iteration number increases. This result verifies the efficacy of the Theorem 1, which states that the iterative sequence \( \{ (z^t, u^t, \lambda^t) \} \) generated by Algorithm 1 converges to a KKT stationary point of problem (15) and the convergence rate is \( O(1/T) \).

Running time: Table III shows the running time of the methods in Figs. 2 and 3. It can be seen that the proposed SP-ADMM algorithm is most efficient in computation time than other methods. Note that while SF has a competitive running time compared to ours, its RMSE is bigger than ours. In addition, although the SDP solver here uses SeDuMi [46], it still consumed a lot of computational time, especially when the network size \( N \) is large. SDP terminates when the norm of the constraint gap reaches the order of \( 10^{-7} \), while the other methods perform 1000 iterations. In contrast, AM-FD and AM-FD-NAG50 require modestly increased computation time as the network size grows. Nevertheless, compared to other parallel methods, AM-FD and AM-FD-NAG50 are not the most sensible choices for large-scale networks and networks with a large average number of neighboring nodes, causing extra delays by the sequential structure under distributed networks. As a result, the structure of parallel implementation would be more efficient. Lastly, we note that the application of the ADMM-H method to large-scale networks is impractical due to the unaffordable sequential running time. Moreover, it can be seen from Table II that the ADMM-H has more parameters to adjust for different networks.

Communication cost: To evaluate the efficiency of various communication methods, we performed a comparison of the RMSE of different methods under benchmark networks as communication costs increased, as depicted in Fig. 4. Our results reveal that although the proposed SP-ADMM incurs a higher communication cost than the SF and AM-FD algorithms during the same iteration (i.e., 1000 steps), it achieves higher accuracy at the same communication cost once convergence has been reached. Additionally, Fig. 4 also indicates that the AM-FD and SF methods were incapable of meeting higher accuracy requirements, even with sufficient communication resources.

B. Synthesized Network

In this subsection, we evaluate the performance of the proposed SP-ADMM algorithm in comparison to ADMM-H [20], SDP [10], SF [19], and AM-FD [21] under varying factors that could potentially impact localization accuracy.

1) Influence of the Number of Sensors (\( N \)): We initiated our investigation by evaluating the performance of random networks with sizes \( N = 2000, 3000, 5000, \) and 10000. Similar to the benchmark networks, the anchor number was set to 2% of the network size, and the value of \( \sigma_{\text{std}} \) of AWGN was chosen to be 7% of the communication range. The results of our experiments are presented in Fig. 5, where Fig. 5(a)–(d) depicts RMSE versus iteration steps and Fig. 5(e)–(h) illustrates RMSE versus communication loss. Similar to the results obtained for the benchmark networks shown in Fig. 4, our findings demonstrate that the proposed SP-ADMM algorithm incurs a higher communication cost compared to the SF and AM-FD algorithms during the same number of iterations (i.e., 1500 steps). However, once convergence has been achieved, the SP-ADMM algorithm provides higher accuracy at the same communication cost. Moreover, Fig. 5(e)–(h) shows that even with sufficient communication resources, the AM-FD and SF methods are incapable of meeting higher accuracy requirements.

2) Influence of the Number of Anchors (\( m \)): The parameters for the random networks and ADMM-H method are summarized in Table IV. The three networks of the experiment are formed by randomly removing 5 anchors, randomly generating 10 anchors, and randomly generating 20 anchors on the benchmark network (\( N = 500, m = 10 \)). Fig. 6 explores the influence of the number of anchor nodes on positioning accuracy. In all three networks, the parameters of our algorithm were set to \( \rho = c = 0.11, u^0 = 0, \) and \( z^0 \) from a distribution \( \text{Unif} (-1, 1)^n \). As expected, the localization accuracy of most methods improves as the number of anchors increases. This is due to the anchors know their true locations, thus can provide more accurate position estimates of the neighboring nodes. In addition, we can see that except for the ADMM-H method, the SP-ADMM-NAG works better than other methods and gets closer to CRBL when increasing the number of anchors. A closer inspection shows that for the AM-FD (light blue line), which is also a nonconvex relaxation method, the proposed SP-ADMM (green line) performs better even though it converges not so fast at the beginning. AM-FD-NAG50 (dark blue line) and SP-ADMM-NAG50 (red line) also have such performance. The running time is shown in Table V, and we find that the SP-ADMM is most computationally time efficient due to the parallel implementation. The results here are similar to those in Table III and consistent with Table I.

3) Influence of the Average Number of Neighbors (\( D_{\text{avg}} \)): Fig. 7 investigates the influence of the average number of neighbors on the performance of four algorithms: the proposed SP-ADMM algorithm, SF, SDP, and AM-FD. The network (\( N = 108, m = 8 \)) randomly placed in the 2D \([0, 1] \times [0, 1] \) area. We set the parameters \( \rho = c = 0.0265, u^0 = 0, \) and \( z^0 = 0.5 \cdot 1 \). As shown in Fig. 7, the RMSE values of most algorithms decrease as the average number of neighbors increases. The reason may be that each node communicates with more neighbors at each iteration, which leads to the position estimated by different nodes for the same node reaching consistency faster.
Notably, the SP-ADMM algorithm consistently achieves the lowest RMSE in all scenarios. Fig. 8 visualizes the position estimates obtained in Fig. 7(c). It clearly shows that the positions estimated by our approach are very close to the true agents’ locations.

4) Influence of the Measurement Noise Variance ($\sigma_{i,j}$): To test the robustness of the proposed algorithm against different measurement noise profiles, the localization scenarios are further extended to sensor networks with different measurement noise variances $\sigma_{i,j}^2 = \sigma_{\text{add}} \|p_i - p_j\|^2$. The corresponding convergence results we obtained are depicted in Fig. 9. Both the network and our proposed algorithm used the same parameters as in Fig. 7(c). Note that Fig. 7(c) shows the test results for $\sigma_{\text{add}} = 0.02$, while we show some other configurations in Fig. 9. Here, we have similar observations as in Fig. 7. Although the performance of these methods demonstrates certain loss with the increase of the noise variance, the localization error of our proposed SP-ADMM algorithm is consistently lower in all scenarios. Moreover, one can see that the non-relaxed methods are relatively insensitive to measurement noise compared with the other convex relaxation methods. Perhaps the reason that the solution of the convex relaxation method is an approximation of the original problem.

We also conducted an experiment to investigate the influence of range-dependent noise on the positioning effectiveness of various methods in extreme cases. To this end, we randomly generated a set of 100 nodes and 8 anchors within a $[0, 1] \times [0, 1]$ square. The communication distance is set to 0.23, and the network topology is depicted in Fig. 10(a). The distance measurements between sensors are subject to Gaussian-distributed noise with zero mean and variance $\sigma_{i,j}^2 = \sigma_{\text{add}} \|p_i - p_j\|^2$. The initial positions are drawn from a uniform

### TABLE IV

| Network Parameters | Method Parameters | ADMM-H | AM-FD |
|--------------------|------------------|---------|-------|
| $N$ | $m$ | $C_{\text{range}}$ | $\sigma_{\text{add}}$ | $D_{\text{avg}}$ | $\epsilon_c$ | $\zeta_c$ | $d_c$ | $\lambda_{\text{max}}$ | $\tau_c$ | $u^T$ |
| Random (Changing $m$) | | | | | | | | | | |
| 495 5 | 0.3 | 0.02 | 12.88 | 0.002 | 0.25 | 0.008 | 0.98 | 1.01 | $10^3$ | 0 |
| 515 20 | 0.3 | 0.02 | 16.48 | 0.002 | 0.25 | 0.008 | 0.98 | 1.01 | $10^3$ | 0 |
| 525 30 | 0.3 | 0.02 | 18.50 | 0.002 | 0.25 | 0.008 | 0.98 | 1.01 | $10^3$ | 0 |

Fig. 5. Convergence performance of different methods with various sizes. (a–d) RMSE value vs. iteration number; (e–h) RMSE value vs. communication cost.

Fig. 6. Convergence performance under a different number of anchors with AWGN for the $N - m = 490$ nodes network. RMSE value (left column); primal feasibility and stationarity gap (right column).
### Table V: Comparisons of Running Time

| Algorithm              | RMSE  | Run time (seconds)                  | Parallelized | Sequential(Per Step) |
|------------------------|-------|-------------------------------------|--------------|----------------------|
| Random ($N = 495, m = 5, step = 2000, CRLB ≈ 0.016$)  |       |                                     |              |                      |
| SF                     | 4.64e-01 | 0.0402                              | 1.97e02(0.0984) |                      |
| AM-FD                  | 1.86e-01 | -                                   | 1.23e02(0.0613) |                      |
| AM-FD-NAG50            | 9.25e-02 | -                                   | 1.24e02(0.0623) |                      |
| SDP                    | 7.99e-02 | -                                   | 1.59e02(4.9609) |                      |
| ADMM-H                 | 7.47e-02 | 0.1412                              | 6.27e03(3.1338) |                      |
| SP-ADMM-NAG50          | 7.45e-02 | 0.0256                              | 1.16e02(0.0581) |                      |
| SP-ADMM                | 1.38e-01 | 0.0251                              | 1.11e02(0.0554) |                      |
| Random ($N = 510, m = 20, step = 1000, CRLB ≈ 0.013$) |       |                                     |              |                      |
| SF                     | 9.40e-02 | 0.0497                              | 1.39e03(1.3411) |                      |
| AM-FD                  | 4.67e-02 | -                                   | 7.20e01(0.07202) |                      |
| AM-FD-NAG50            | 2.31e-02 | -                                   | 7.66e01(0.0768) |                      |
| SDP                    | 2.65e-02 | -                                   | 3.24e02(10.4481) |                      |
| ADMM-H                 | 1.38e-02 | 1.31                                | 3.56e03(3.5575) |                      |
| SP-ADMM-NAG50          | 1.97e-02 | 0.03232                             | 8.10e01(0.0881) |                      |
| SP-ADMM                | 3.72e-02 | 0.0388                              | 8.29e01(0.0811) |                      |
| Random ($N = 520, m = 30, step = 1000, CRLB ≈ 0.013$) |       |                                     |              |                      |
| SF                     | 7.94e-02 | 0.0411                              | 1.81e03(1.8142) |                      |
| AM-FD                  | 3.67e-02 | -                                   | 8.20e01(0.0821) |                      |
| AM-FD-NAG50            | 1.90e-02 | -                                   | 5.57e02(18.5801)|                      |
| SDP                    | 2.56e-02 | -                                   | 3.51e03(3.5150) |                      |
| ADMM-H                 | 1.33e-02 | 1.41                                | 3.51e03(3.5150) |                      |
| SP-ADMM-NAG50          | 1.35e-02 | 0.0343                              | 8.54e01(0.0854) |                      |
| SP-ADMM                | 1.90e-02 | 0.0351                              | 8.51e01(0.0811) |                      |

b These symbols “-” in the table mean that the methods cannot be implemented in parallel.

---

Fig. 7. Convergence performance under different average number of neighboring nodes ($D_{avg}$). Here, we adopt $\sigma_{add} = 0.02, N = 108, m = 8$. The anchors are marked by ■, the true agent positions by + and the estimated ones by ◦.

---

distribution $Unif(-1, 1)^{200}$, and we calculated the corresponding Root-Mean-Squared-Error (RMSE) value as follows:

$$RMSE(0) = \sqrt{\frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \sum_{i \in (N/A)} (p_0^i - p_i)^T (p_0^i - p_i)},$$

where $N_{MC}$ denotes the count of independent Monte Carlo trials, while $p_{0,k}$ symbolizes the initial position vector of the $i$-th node in the $k$-th Monte Carlo trial. For Fig. 10, we set $w_{i,j} \sim Gauss(0, \sigma_{add}\|p_i - p_j\|^2)$.
performance improves with decreasing values of the parameter too large may significantly slow down the convergence rate in the range of $N_{MC} = 50$. When the RMSE of the algorithm is larger than the initial RMSE, it means that the algorithm has failed to improve the accuracy of the sensor positions and may have even made it worse. From Fig. 10, one can observe that the proposed SP-ADMM failed to provide appropriate position estimates when the variance of the range-noise is bigger than 60. Additionally, Fig. 10 shows that the RMSEs of all methods increase as the noise variance becomes larger. Nevertheless, the proposed SP-ADMM shows the slowest performance degradation, indicating that the proposed method is more robust than AM-FD method and SDP method.

C. Evaluation of the Proposed SP-ADMM Algorithm Versus Penalty Parameters

We simulated the proposed SP-ADMM algorithm under different values of the penalty parameters $c, \rho$, as well as the initial value $u^0$. The different network parameters ($C_{\text{scale}}, C_{\text{range}}, N_{\text{max}} = \{N_i, i \in N\}$) are also considered. Fig. 12 summarizes the RMSE after 1000 iterations. Figs. 11 and 12 not only show the sensitivity of the proposed SP-ADMM algorithm to parameters $\rho, c$, and $u^0$, but also provide practical advice that can assist users, using the proposed SP-ADMM algorithm, to reduce the time in adjusting parameters $\rho, c, u^0$ on different networks.

1) The Selection Strategy of $\rho$ and $c$: Theorem 1 guarantees that the proposed Algorithms will converge if $\rho$ is selected in accordance with (62)–(64) for a given $c$. To corroborate this claim, we present in Fig. 11 the RMSE with iterations for different values of $\rho$, while maintaining $c = 0.1$ under the benchmark network ($N = 500, m = 10$). Our experimental results demonstrate that, in practical implementation, the selection of proximal penalty coefficients impacts the convergence rate and final accuracy of the proposed algorithm. Specifically, for a benchmark network with $N = 500$ and $m = 10$, we observed that choosing a smaller value of $c$ or $\rho$ within an appropriate range resulted in the fastest convergence and highest accuracy. For instance, as depicted in Fig. 11(a) (Fig. 11b), when $c(\rho)$ is in the range of $[1.1 \cdot \rho, 2 \cdot \rho]$ ($[0.7 \cdot c, 1.2 \cdot c]$), the algorithm’s performance improves with decreasing values of the parameter $c$. However, selecting a value of $c$ or $\rho$ that is too small or too large may significantly slow down the convergence rate and compromise the final accuracy, as evidenced by the yellow, brown, red, and purple lines in the two subfigures of Fig. 11. The appropriate range of the parameter values is typically selected empirically based on the specific data used.

2) Localization Performance With $u$ and $c$ Selection: Fig. 12(a) and 12(b) validates this statement. $C_{\text{scale}}$ denotes the side length of the square area where the network is deployed, for example, $C_{\text{scale}} = 10$ indicates that the network deployed over $[0, 10] \times [0, 10]$ area. Fig. 12(a) and 12(b) employs the same sensor network as Figure, except for the variance of the measured noise $\sigma^2_{i,j}$, which is set to $\sigma^2_{i,j} = 0.02 \|p_i - p_j\|^2$. We can see that the curves at different $C_{\text{scale}}$ show tiny fluctuations for varying $u$ than in changing $c$. This is because the convergence rate is strongly dependent on the penalty parameter $c$, please see the supplementary material [32] for the specific form of the convergence rate. These results indicate that the performance of the proposed SP-ADMM is not sensitive to initial parameter $u$ but sensitive to penalty parameter $c$. Besides, Fig. 12(b) shows that the appropriate interval value of $c$ for different $C_{\text{scale}}$ is consistent, and cannot be too large or too small. Therefore, parameter $c$ should be adjusted more carefully when the proposed SP-ADMM algorithm is applied to different networks.

3) The Selection Strategy of $c$ Based on the Communication Range: In this part, we focus on how the parameter $c$ should be chosen for networks with different communication ranges. The network composed of $N = 108, m = 8$ nodes located over $[0, 1] \times [0, 1]$ square area and the measurement noise with variance $\sigma^2_{i,j} = 0.02 \|p_i - p_j\|^2$. Fig. 12(c) depicts the evolution of RMSE under networks at different communication ranges. Under the same number of iterative steps, it can be seen from Fig. 12(c) that networks with large $C_{\text{range}}$ can obtain a lower RMSE by choosing a large $c$, networks with smaller $C_{\text{range}}$ can reach lower RMSE by selecting smaller $c$.

These results are consistent with (62)–(64) derived from Lemma 2. In detail, the maximum measurement distance $d_{\text{max}}$, the maximum number of adjacent nodes $N_{\text{max}}$ and the total
number of adjacent nodes $N_{\text{sum}}$ all increase as $C_{\text{range}}$ increases. Therefore, according to (62)–(64), $\rho$ needs to be increased to ensure the convergence of the proposed algorithm. Since we set $\rho = c$ in all experiments, it is reasonable for $c$ to larger. Moreover, we also obtain similar results for the benchmark network, such as $\rho = c = 0.11$ ($C_{\text{range}} = 0.3, N = 500$) is greater than $\rho = c = 0.0197$ ($C_{\text{range}} = 0.1, N = 1000$). These results serve as the rule-of-thumb for selecting the parameter $c$ of the proposed SP-ADMM algorithm when it is applied to different networks.

4) Convergence Rate With $N_{\text{max}}$: Fig. 12(d) displays the RMSE versus the maximum number of adjacent nodes $N_{\text{max}}$ for different values of $c$. The sensor network graph is the same as Fig. 12(c) at $C_{\text{range}} = 0.25$. As seen in Fig. 12(d), a lower RMSE is obtained at a larger $N_{\text{max}}$, which means that for the same penalty $c$, the proposed algorithm converges faster on a network with a larger $N_{\text{max}}$. This is indeed consistent with our convergence rate of Theorem 1. It can be seen from the supplement [32] that the coefficient $e_2$ of the sublinear convergence rate is inversely proportional to $N_{\text{max}}$ when $c, \rho, N_1$ and $e_2$ are fixed.

D. Summary

The proposed algorithm has been evaluated against SDP, SF, AM-FD, and ADMM-H methods in terms of RMSE, running time, and communication cost across a range of network scenarios. These scenarios include different numbers of anchors and sensor nodes, average number of neighboring nodes, and different variance levels of the measurement noise under both benchmark and synthesized network configurations. Furthermore, we have included an empirical investigation on the impact of the penalty parameters $\rho$ and $c$ for our proposed SP-ADMM algorithm.

The results indicated that the proposed algorithm surpasses the SDP, SF, and AM-FD methods in terms of localization accuracy, while exhibiting lower computational complexity and communication cost than the ADMM-H method. Although the communication cost of the proposed SP-ADMM is higher than that of the SF and AM-FD algorithms per iteration, it provides higher accuracy at the same communication cost once convergence has been reached. Overall, the proposed SP-ADMM is a promising solution for cooperative localization problem in wireless sensor networks.

One possible future direction is to explore selective communication strategy among nodes to reduce the communication burden. This involves nodes randomly or sequentially choosing a subset of neighboring nodes to communicate with during each iteration. Additionally, addressing outliers in realistic measurement data and fusing other types of measurements are also important areas for future research.

VI. CONCLUSION

In this article, we have proposed novel distributed SP-ADMM algorithms for wireless sensor network localization in Gaussian measurement noise. Our proposed algorithms directly tackle the challenging nonconvex and nonsmooth problem without resorting to convex relaxation. We have shown that Algorithm 1 has a low computational complexity (in Table I) and proved that the algorithm converges to a critical point of the original problem (in Lemma 3, 4, and Theorem 1) at rate $O(1/T)$. Moreover, we have proposed Algorithm 2, which is an improved version of Algorithm 1 in terms of storage space. Simulation and experimental results not only have shown that the proposed SP-ADMM algorithm is robust and performs well in the different average number of neighboring nodes, measurement noise variance, coverage area, and the number of anchors and non-anchors, but also provided suggestions of the parameters for migrating the proposed algorithm to different networks.

REFERENCES

[1] P. K. Singh, B. K. Bhargava, M. Paprzycki, N. C. Kaushal, and W.-C. Hong, *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario’s*, vol. 1132, Cham, Switzerland: Springer-Verlag, 2020.

[2] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[3] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, “Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms,” *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1448–1463, Mar. 2015.

[4] Y.-M. Pun and A. M.-C. So, “Local strong convexity of source localization and error bound for target tracking under time-of-arrival measurements,” *IEEE Trans. Signal Process.*, vol. 70, pp. 190–201, 2021.

[5] N. Okello, F. Fletcher, D. Musicki, and B. Ristic, “Comparison of recursive algorithms for emitter localisation using TDOA measurements from a pair of UAV’s,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 1723–1732, Jul. 2011.

[6] S. Xu and K. Doğançay, “Optimal sensor placement for 3-D angle-of-arrival target localization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 3, pp. 1196–1211, Jun. 2017.

[7] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, “Received-signal-strength threshold optimization using Gaussian processes,” *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2164–2177, Apr. 2017.

[8] B. Jin, F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “Bayesian cooperative localization using received signal strength with unknown path loss exponent: Message passing approaches,” *IEEE Trans. Signal Process.*, vol. 68, pp. 1120–1135, 2020.

[9] A. Simonetto and G. Jous, “Distributed maximum likelihood sensor network localization,” *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014.

[10] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, “Semidefinite programming based algorithms for sensor network localization,” *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, 2006.

[11] P. Biswas, T.-C. Lian, K.-C. Toh, Y. Ye, and T.-C. Wang, “Semidefinite programming approaches for sensor network localization with noisy distance measurements,” *IEEE Trans. Automat. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[12] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, “Further relaxations of the semidefinite programming approach to sensor network localization,” *SIAM J. Optim.*, vol. 19, no. 2, pp. 655–673, 2008.

[13] S. Srinagarajan, A. H. Tewfik, and Z.-Q. Luo, “Distributed sensor network localization using SOCP relaxation,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4886–4895, Dec. 2008.

[14] P. Di Lorenzo and G. Scutari, “NEXT: In-network nonconvex optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[15] D. Jin, F. Yin, A. M. Zoubir, and H. C. So, “Exploiting sparsity of ranging biases for NLOS mitigation,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3782–3795, 2021.

[16] R. M. Buehrer, H. Wymeersch, and R. M. Vaughan, “Collaborative sensor network localization: Algorithms and practical issues,” *Proc. IEEE*, vol. 106, no. 6, pp. 1089–1114, Jun. 2018.

[17] M. Hong, D. Hajinezhad, and M.-M. Zhao, “Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks,” in *Proc. Int. 4th Conf. Mach. Learn.*, PMLR, 2017, pp. 1529–1538.
ZHANG et al.: DISTRIBUTED SCALED PROXIMAL ADMM ALGORITHMS FOR COOPERATIVE LOCALIZATION IN WSNs

3327

[38] X. Zhang, J. Ma, Z. Cheng, S. Huang, C. W. de Silva, and T. H. Lee, “Improved hierarchical ADMM for nonconvex cooperative distributed model predictive control,” 2020, arXiv:2011.00463.

[39] Y. E. Nesterov, “A method of solving a convex programming problem with convergence rate O(1/k^2),” in Dokl. Akad. Nauk SSSR, vol. 209, no. 3, pp. 543–547, 1983.

[40] Y. Ye, “Computational optimization laboratory,” Stanford University, Stanford, CA, USA, 2007. [Online]. Available: https://web.stanford.edu/~yye/col.html

[41] N. Parwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O’dea, “Relative location estimation in wireless sensor networks,” IEEE Trans. Signal Process., vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[42] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” Optim. Methods Softw., vol. 11, nos. 1–4, pp. 625–653, 1999.

Mei Zhang received the M.Sc. degree from the School of Mathematics at Sichuan University, Chengdu, China, in 2017. She is presently dedicated to pursuing the Ph.D. degree at the same institution. Her primary research interests lie in the domains of nonconvex optimization and its practical applications, with a particular focus on their relevance in the localization of wireless sensor networks.

Zhiqiu Wang received the Ph.D. degree from Sichuan University, Chengdu, China, in 2018. From 2018 to 2020, he was Postdoctoral Researcher with the Chinese University of Hong Kong, Shenzhen, China. Since 2020, he has been working as a specially appointed Associate Research Fellow with the College of Mathematics, Sichuan University. His current research interests are in the fields of multimodal data fusion, federated learning, semisupervised learning, optimal transport, geometric deep learning, and various optimization problems from target tracking and big data applications.

Feng Yin (Senior Member, IEEE) received the B.Sc. degree from Shanghai Jiao Tong University, China, and the M.Sc. and Ph.D. degrees from TU Darmstadt, Germany. From 2014 to 2016, he was a Marie-Curie Postdoctoral Researcher with Ericsson Research, Linköping, Sweden. Since 2016, he has been an Assistant Professor with The Chinese University of Hong Kong, Shenzhen. His research interests include statistical signal processing, Bayesian learning and optimization, and sensory data fusion.

Xiaojing Shen received the B.Sc. degree from Sichuan University, in 2003, the M.Sc. degree from Jilin University, in 2006, and the Ph.D. degree from Sichuan University, in 2009. From 2009 to 2011, he held a Postdoctoral position and was an Assistant Professor with the School of Computer Science, Sichuan University. He was with Syracuse University as a Postdoctoral Research Associate from 2012 to 2013. Since 2013, he has been a Professor with the School of Mathematics, Sichuan University. He is one of the authors of two books—Networked Multisensor Decision and Estimation Fusion: Based on Advanced Mathematical Methods (CRC Press, 2012) and Nonlinear Estimation and Applications to Industrial Systems Control (G. Rigatos, ed., ch. 3, pp. 61–88, Nova Science, 2012). He also served as a co-operator to several major research institutes. His current research interests include multisensor decision and estimation fusion, multtarget data association and tracking, optimization theory with applications to information fusion, and statistical theory with applications to information fusion. He received the China National Excellent Doctoral Dissertation Award (the highest honor of Ph.D. degree).