Robust Primal-Dual Proximal Algorithm for Cooperative Localization in WSNs

Mei Zhang, Xiaojing Shen, Zhiguo Wang
Department of Mathematics
Sichuan University
Chengdu, China
Email: 963205859@qq.com, xiao23332@163.com, wangzhiguo@scu.edu.cn

Pramod K. Varshney
Department of Electrical Engineering and Computer Science
Syracuse University
Syracuse, USA
Email: varshney@syr.edu

***Abstract*—This paper addresses the localization challenge in cooperative multi-agent wireless sensor networks, specifically focusing on range-based localization. To enhance robustness against outliers in range measurements, we employ the Huber function, leading to the formulation of a robust yet nonconvex optimization problem with coupled agent variables. Confronted with this nonconvex optimization challenge, particularly in large-scale networks, we reformulate the problem using Lagrange duality and conjugate theory. This restructuring yields subproblems characterized by smooth strong convexity for dual variables and a simplified form for primal variables, thereby facilitating an efficient solution. Building upon this reformulation, we introduce a novel distributed primal-dual algorithm that employs coordinate descent and proximal minimization techniques within an iterative framework. This approach furnishes closed-form solutions for both primal and dual variables. Theoretically, our method ensures not only the convergence of the sequence of objective function values but also, by leveraging the Kurdyka-Łojasiewicz property, we establish the guaranteed global convergence of the location estimates sequence to a critical point of the original objective function. Notably, our proposed approach exhibits lower computational complexity, communication cost, and storage space compared to existing methods. Numerical experiments underscore the superiority of the proposed method in terms of robustness and localization accuracy when compared to the other methods in the literature.**

***Index Terms*—Primal-dual algorithm, Huber loss, non-convex optimization, wireless sensor network localization**

I. Introduction

Accurate localization of nodes within a wireless sensor network (WSN) is critical for numerous applications, such as environmental monitoring, disaster management to healthcare, and military operations [1]. This paper focuses on a cooperative sensor network, comprising N nodes that need to be localized and m anchors with known locations. The network graph structure is assumed known, where each sensor node is uniquely identified by an index in the set $\mathcal{V} = \{1, 2, \ldots, N\}$, and anchor indices belong to $\mathcal{A} = \{N+1, N+2, \ldots, N+m\}$. The true location of node $i \in \mathcal{V}$ is denoted as $\mathbf{p}_i \in \mathbb{R}^n$, and the location of anchor $j \in \mathcal{A}$ is denoted as $\mathbf{a}_j \in \mathbb{R}^n$. The set of neighboring node pairs in the entire graph is represented as $\mathcal{E}$, with a cardinality of $M = |\mathcal{E}|$. For each pair of nodes $(i, j) \in \mathcal{E}$, the available noisy distance $d_{i,j}$ represents the true distance $\|\mathbf{p}_i - \mathbf{p}_j\|$ disturbed by measured noise and outliers. Following the convention in [2], assuming symmetry such that $d_{i,j} = d_{j,i}$. To estimate the positions $\mathbf{p} = [\mathbf{p}_1^T, \ldots, \mathbf{p}_N^T]^T \in \mathbb{R}^{mN}$ of all nodes, a commonly used formulation is provided by the following least squares (LS) optimization problem:

$$\min_{\mathbf{p}\in\mathbb{R}^{nN}} \sum_{(i,j)\in\mathcal{E}} \left(d_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\|\right)^2 \tag{1a}$$

$$\text{subject to} \quad \mathbf{p}_j = \mathbf{a}_j,\ j \in \mathcal{A}. \tag{1b}$$

Various methods have been proposed to tackle problem (1), as outlined in [3]–[5]. Unfortunately, LS based localization is sensitive to the outliers in distance measurements and can lead to a poor performance in the presence of outliers. Practical scenarios may introduce outliers due to non-line-of-sight (NLOS) transmission errors, malicious attacks, and other factors [6]. This necessity prompts the investigation of robust variants of localization algorithm [7]. A popular choice is to use the l_1 norm. However, l_1 norm exhibits non-smoothness due to a singularity at its origin [8]. To mitigate this issue, Huber loss function [9] offers a beneficial interpolation between l_1 and l_2 norm based minimizations. Thus, instead of minimizing the square of ranging errors, one seeks to minimize its Huber loss, i.e.,

$$\min_{\mathbf{p}\in\mathbb{R}^{nN}} \ \mathcal{J}_\varepsilon(\mathbf{p}) \tag{2a}$$

$$\text{subject to} \quad \mathbf{p}_j = \mathbf{a}_j,\ j \in \mathcal{A}, \tag{2b}$$

where $\mathcal{J}_\varepsilon(\mathbf{p}) = \sum_{(i,j)\in\mathcal{E}} j_\varepsilon\left(d_{ij} - \|\mathbf{p}_i - \mathbf{p}_j\|\right)$, and $j_\varepsilon(\cdot)$ is defined, for a given $\varepsilon > 0$, by

$$j_\varepsilon(x) = \begin{cases} \frac{1}{2}x^2, & |x| \le \varepsilon, \\ \varepsilon|x| - \frac{1}{2}\varepsilon^2, & |x| > \varepsilon. \end{cases} \tag{3}$$

However, due to the nonconvex and nonseparable nature of problem (2), directly minimizing it to obtain a satisfactory solution is challenging. Therefore, many methods resort to employing convex relaxation techniques. In [8], the authors approximated the Huber function using a convex upper bound. Leveraging majorization-minimization (MM) [10] and the distributed weighted Multi-Dimensional Scaling (dwMDS) algorithm [11], the authors derived a centralized approach that relies on *a priori* information about node positions. Another ap-

TABLE I
COMPARISONS OF DIFFERENT ALGORITHMS

Algorithm	Consider Problem	Locate Nodes	Number of Tuning Parameters	Convergence: CvxRelax problem	Convergence: Original Global[†]	Sensor i: Computational Complexity	Sensor i: Communication Cost	Sensor i: Storage Space
SDP [4]	(1)	$N \geq 1$	0	✔	✗	$\mathcal{O}(n^3)$	nN_i	$nN + M - \lvert\mathcal{E}_{\mathrm{a}}\rvert^{\circledast}$
SF [5]	(1)	$N \geq 1$	0	✔	✗	$\mathcal{O}(nN_i)$	nN_i	$(1+2n)^{\divideontimes}$
HuberMM [12]	(2)	$N = 1$	$(m+1)^{*}$	✔	✗	$\mathcal{O}(2nm)$	nm	$m+1+n$
GD-Huber [13]	(2)	$N \geq 1$	2	✔	✗	$\mathcal{O}(2nN_i)$	nN_i	$2+n$
2-StageHuber [15]	(2)	$N \geq 1$	4	✗	✗	$\mathcal{O}(2nN_i)$	nN_i	$4+n$
STRONG [18]	(2)	$N \geq 1$	0	✔	✗	$\mathcal{O}(2nN_i)$	nN_i	$1+2n$
Algorithm 1 (ours)	(2)	$N \geq 1$	$2^{\star}$	✔	✔	$\mathcal{O}(nN_i)$	nN_i	$2+n$

* $(m+1)$ refers to the noise standard deviation of m anchors and the parameter of the Huber function.
* In numerical simulations, the penalty parameter $\tau = 0$, so only one parameter ε needs to be trained.
$^{\divideontimes}$ This refers to the storage space of parallel method in [5].
$^{\circledast}$ $\lvert\mathcal{E}_{\mathrm{a}}\rvert$ 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 [21].
† Here, "Original Global" signifies the global convergence of the sequence to a critical point of the original objective function $\mathcal{J}_{\varepsilon}(\mathbf{p})$. A critical point is defined as a point where the derivative of the function equals zero.

proach requiring prior information is presented in [12], where the authors, focusing on the case of $N = 1$, developed an algorithm with a tightly upper bound for the Huber loss within the MM framework. For cases where $N \geq 1$ and no prior information is available, optimization methods include the distributed stochastic gradient descent (SGD) algorithm based on a convex relaxation of the Huber loss function proposed in [13]. While this algorithm exhibits fast convergence, rendering it suitable for real-time applications, it is susceptible to harsh NLOS effects [14]. Another approach is introduced in [15], that presents a two-stage robust distributed gradient descent (GD) algorithm. The first stage employed is a convex underestimator[1] of the Huber function, while the second stage directly applies GD methods to tackle problem (2). However, this approach introduces more parameters requiring adjustment and lacks theoretical proof of convergence properties. Moreover, utilizing GD methods to find the minimum value of the Huber function can easily get trapped in local optima [17]. In [18], the authors developed the same relaxation technique as in [15] and devised both synchronous and asynchronous algorithms. Recently, in [19], the authors also adopted this relaxation to investigate problem (2) under hybrid received signal strength (RSS) and time-of-arrival (TOA) observations. While most of the aforementioned works have provided convergence analysis for the convexified problem, global convergence[2] guarantees for the original nonconvex problem (2) remain unaddressed.

Based on the above motivating discussion, we propose a robust distributed method for tackling problem (2) that not only guarantees global convergence but also maintains low computational complexity and communication cost.

Our contributions can be succinctly outlined as follows:

- By applying Lagrange duality and conjugate theory, we introduce a novel reformulation of problem (2). This reformulation exhibits smoothness and strong convexity for dual variables while maintaining a simplified structure for primal variables, enabling efficient resolution.
- Exploiting the favorable optimization structure of our reformulation, we devise a distributed primal-dual algorithm that combines coordinate descent and proximal minimization techniques. Noteworthy is the algorithm's efficiency in large-scale networks, necessitating only simple computations and imposing minimal communication requirements at each node, as illustrated in Table I.
- We not only demonstrate the convergence of the sequence of objective function values but, more significantly, leverage the Kurdyka-Łojasiewicz (KŁ) property (refer to [22]) to establish that the sequence generated by our proposed method globally converges to a critical point of the original nonconvex problem (2). This global convergence assurance adds a solid theoretical foundation to the effectiveness of our algorithm.

Notation: We denote vectors using lowercase bold letters. The set of real numbers is represented as $\mathbb{R}$. The set of real n-vectors is denoted as $\mathbb{R}^n$ and the set of positive real n-vectors is expressed as $\mathbb{R}^n_+$. The Euclidean norm of a real vector $\mathbf{p}$ is represented by $\|\mathbf{p}\|$. The column vector obtained by concatenating $\mathbf{u}_{i,j}$ for all $(i,j) \in \mathcal{E}$ is denoted as $\mathrm{vec}(\mathbf{u}_{i,j}, (i,j) \in \mathcal{E})$. The unit closed ball of $\mathbb{R}^n$ centered at the origin is denoted as $\mathcal{B} := \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v}\| \leq 1\}$. The Cartesian product of M copies of the unit ball $\mathcal{B}$ is denoted as $\mathcal{B}^M := \mathcal{B} \times \mathcal{B} \times \cdots \times \mathcal{B}$. Furthermore, we use $\delta_{\mathcal{B}}(\cdot)$ to represent the indicator function of $\mathcal{B}$, i.e.,

$$\delta_{\mathcal{B}}(\mathbf{u}) := \begin{cases} 0, & \mathbf{u} \in \mathcal{B}, \\ +\infty, & \text{otherwise.} \end{cases}$$

[1]Function g is an underestimator of function f if $g(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathrm{dom}\, f$ [16].

[2]Here, "global convergent sequences" denotes that, for any arbitrary starting point, the algorithm generates a sequence that converges to a solution [20].

Organization: In Section II, we present the reformulation of problem (2). The proposed algorithm and the corresponding theoretical results are presented in Section III and IV, respectively. The performance of the proposed algorithm is demonstrated in Section V, and the conclusion is provided in Section VI.

II. Problem Reformulation

To address the difficulties associated with directly minimizing the loss function in the original problem (2), we adopt a dual approach to attain a simplified and tractable structure. By introducing an auxiliary variable, we initially reformulate problem (2) as follows:

$$\min_{\mathbf{z},\mathbf{p}} \sum_{(i,j)\in\mathcal{E}} j_\varepsilon\left(z_{i,j}\right) \tag{4a}$$

$$\text{subject to} \quad d_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\| - z_{i,j} = 0, (i,j) \in \mathcal{E}, \tag{4b}$$

$$\mathbf{p} \in \mathcal{X}, \tag{4c}$$

where $\mathbf{z} = \text{vec}(z_{i,j}, (i,j) \in \mathcal{E})$, $\mathcal{X} = \{\mathbf{p} \mid \mathbf{p}_j = \mathbf{a}_j, \forall\, j \in \mathcal{A}\}$. The Lagrangian function corresponding to (4) is defined as

$$\mathcal{L}\left(\mathbf{p},\mathbf{z},\boldsymbol{\lambda}\right) = \sum_{(i,j)\in\mathcal{E}} \left[j_\varepsilon\left(z_{i,j}\right) + \lambda_{i,j}\left(d_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\| - z_{i,j}\right)\right].$$

where $\boldsymbol{\lambda} = \text{vec}(\lambda_{i,j}, (i,j) \in \mathcal{E})$. Then the dual function is given by

$$\begin{aligned}\inf_{\mathbf{p}\in\mathcal{X},\mathbf{z}} \mathcal{L}\left(\mathbf{p},\mathbf{z},\boldsymbol{\lambda}\right) = &\inf_{\mathbf{p}\in\mathcal{X}} \Big\{ \sum_{(i,j)\in\mathcal{E}} \lambda_{i,j}\left(d_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\|\right) \Big\} \\ &+ \inf_{\mathbf{z}} \Big\{ \sum_{(i,j)\in\mathcal{E}} j_\varepsilon\left(z_{i,j}\right) - \boldsymbol{\lambda}^T\mathbf{z} \Big\}. \end{aligned} \tag{5}$$

Using the definition of conjugate [16], the second term on the right-hand side of (5) can be expressed as

$$\begin{aligned}\inf_{\mathbf{z}} \sum_{(i,j)\in\mathcal{E}} j_\varepsilon\left(z_{i,j}\right) - \boldsymbol{\lambda}^T\mathbf{z} &= -\sup_{\mathbf{z}} \boldsymbol{\lambda}^T\mathbf{z} - \sum_{(i,j)\in\mathcal{E}} j_\varepsilon\left(z_{i,j}\right) \\ &= -\sum_{(i,j)\in\mathcal{E}} j_\varepsilon^*\left(\lambda_{i,j}\right), \end{aligned} \tag{6}$$

where $j_\varepsilon^*\left(\lambda_{i,j}\right)$ represents the conjugate of the Huber function $j_\varepsilon\left(x\right)$, and this can be determined using the following result.

Lemma 1: The conjugate of Huber function $j_\varepsilon\left(x\right) : \mathbb{R} \to \mathbb{R}$ is

$$j_\varepsilon^*\left(y\right) = \begin{cases} \frac{1}{2}y^2, & |y| \le \varepsilon, \\ +\infty, & |y| > \varepsilon. \end{cases} \tag{7}$$

Proof: See Appendix A. ∎

Therefore, by substituting (7) and (6) into (5), the Lagrange dual problem corresponding to problem (4) is

$$\max_{\|\boldsymbol{\lambda}\|_\infty \le \varepsilon} \min_{\mathbf{p}\in\mathcal{X}} \overbrace{\sum_{(i,j)\in\mathcal{E}} \lambda_{i,j}\left(d_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\|\right) - \frac{1}{2}\|\boldsymbol{\lambda}\|^2}^{F(\boldsymbol{\lambda},\mathbf{p})}. \tag{8}$$

Notably, $F\left(\boldsymbol{\lambda},\mathbf{p}\right)$ is quadratic strongly concave for fixed $\mathbf{p}$. Moreover, for a fixed $\boldsymbol{\lambda}$, $F\left(\boldsymbol{\lambda},\mathbf{p}\right)$ takes the simple form of a nonsmooth Euclidean norm with coefficient $\boldsymbol{\lambda}$. This structure can be efficiently addressed by introducing an auxiliary variable and a proximal term, as illustrated in the following section.

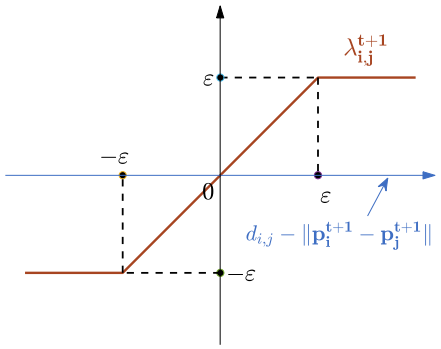


Fig. 1. Illustration depicting the iterative update of the dual variables $\lambda_{i,j}^{t+1}$ as defined in Equation (11). Each update process is constrained within the interval $[-\varepsilon, \varepsilon]$.

III. A Distributed Primal-Dual Algorithm

To solve the optimization problem stated in (8), we adopt an alternate iteration approach where variables are updated individually. At iteration t, the update steps for the variables $(\mathbf{p}^t, \boldsymbol{\lambda}^t)$ are as follows:

$$\mathbf{p}^{t+1} = \arg\min_{\mathbf{p}\in\mathcal{X}} F\left(\boldsymbol{\lambda}^t, \mathbf{p}\right), \tag{9}$$

$$\boldsymbol{\lambda}^{t+1} = \arg\max_{\|\boldsymbol{\lambda}\|_\infty \le \varepsilon} F\left(\boldsymbol{\lambda}, \mathbf{p}^{t+1}\right). \tag{10}$$

For each $(i,j) \in \mathcal{E}$ in problem (10), we have

$$\begin{aligned}\lambda_{i,j}^{t+1} =& \arg\max_{|\lambda_{i,j}|\le\varepsilon} -\frac{1}{2}\left(\lambda_{i,j} - \left(d_{i,j} - \|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}\|\right)\right)^2 \\ =& \text{proj}_{|\lambda_{i,j}|\le\varepsilon}\left(d_{i,j} - \|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}\|\right), \end{aligned}$$

where $\text{proj}_{|\lambda_{i,j}|\le\varepsilon}$ represents the projection onto the interval $[-\varepsilon, \varepsilon]$ and the analytic solution for this problem is

$$\lambda_{i,j}^{t+1} = \max\left(-\varepsilon, \min\left(\varepsilon, d_{i,j} - \|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}\|\right)\right). \tag{11}$$

The updated step (11) for $\lambda_{i,j}^{t+1}$ implies that its value can take on positive, negative, or zero value during each iteration. Both its sign and magnitude depend on the estimate of $\mathbf{p}^{t+1}$, as visualized in Fig. 1. This introduces uncertainty regarding the convexity of problem (9). Therefore, the main challenges in solving problem (9) involve eliminating the uncertainty associated with $\lambda_{i,j}^t$ and reformulating (9) into a computationally tractable optimization problem. To address this issue, we introduce a proximal term, leading to a modified problem:

$$\begin{aligned}\arg\min_{\mathbf{p}\in\mathcal{X}} \Big\{ F\left(\boldsymbol{\lambda}^t, \mathbf{p}\right) + \sum_{(i,j)\in\mathcal{E}} \frac{\tilde{\lambda}_{i,j}^t}{2}\left(\|\mathbf{p}_i - \mathbf{p}_j\| - \|\mathbf{p}_i^t - \mathbf{p}_j^t\|\right)^2 \\ + \frac{\tau}{2}\|\mathbf{p} - \mathbf{p}^t\|^2 \Big\} \end{aligned} \tag{12}$$

where $\tau > 0$ is a penalty coefficient, and the parameter $0 < \tilde{\lambda}_{i,j}^t \leq 1$ is defined as

$$\tilde{\lambda}_{i,j}^t := \frac{\varepsilon}{\max\{\varepsilon, |d_{i,j} - \|\mathbf{p}_i^t - \mathbf{p}_j^t\||\}}. \tag{13}$$

According to the $\lambda_{i,j}^t$ update rule in (11), we have

$$\lambda_{i,j}^t = \tilde{\lambda}_{i,j}^t (d_{i,j} - \|\mathbf{p}_i^t - \mathbf{p}_j^t\|). \tag{14}$$

Then, by substituting (14) into (12) and combining the result with the form of F in (8), we can further rewrite (12) as:

$$\begin{aligned}\arg\min_{\mathbf{p}\in\mathcal{X}} \Big\{ \sum_{(i,j)\in\mathcal{E}} \frac{\tilde{\lambda}_{i,j}^t}{2} \Big(\|\mathbf{p}_i - \mathbf{p}_j\|^2 - 2d_{i,j} \underbrace{\|\mathbf{p}_i - \mathbf{p}_j\|}_{\text{nonsmooth}} + d_{i,j}^2 \\ + (1 - \tilde{\lambda}_{i,j}^t)(d_{i,j} - \|\mathbf{p}_i^t - \mathbf{p}_j^t\|)^2 \Big) + \frac{\tau}{2}\|\mathbf{p} - \mathbf{p}^t\|^2 \Big\}. \end{aligned} \tag{15}$$

Motivated by recent works [3], [23], we apply the Cauchy-Schwartz inequality to obtain

$$\begin{aligned} -\|\mathbf{p}_i - \mathbf{p}_j\| &= -\max_{\mathbf{u}_{i,j}\in\mathcal{B}} \ \mathbf{u}_{i,j}^T(\mathbf{p}_i - \mathbf{p}_j) \\ &= \min_{\mathbf{u}_{i,j}\in\mathcal{B}} \ -\mathbf{u}_{i,j}^T(\mathbf{p}_i - \mathbf{p}_j), \end{aligned} \tag{16}$$

where $\mathbf{u}_{i,j} \in \mathbb{R}^n$ represents an auxiliary variable. By substituting (16) into (15), we obtain the following smooth reformulation of problem (15):

$$\arg\min_{\mathbf{p}\in\mathcal{X}, \mathbf{u}\in\mathcal{B}^M} G(\mathbf{p}, \mathbf{u}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) + \frac{\tau}{2}\|\mathbf{p} - \mathbf{p}^t\|^2, \tag{17}$$

where

$$\begin{aligned} G(\mathbf{p}, \mathbf{u}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) = \sum_{(i,j)\in\mathcal{E}} \frac{\tilde{\lambda}_{i,j}^t}{2} \Big(\|\mathbf{p}_i - \mathbf{p}_j\|^2 - 2d_{i,j}\mathbf{u}_{i,j}^T(\mathbf{p}_i - \mathbf{p}_j) \\ + d_{i,j}^2 + (1 - \tilde{\lambda}_{i,j}^t)(d_{i,j} - \|\mathbf{p}_i^t - \mathbf{p}_j^t\|)^2 \Big). \end{aligned}$$

Note that the variables of the function $G(\tilde{\boldsymbol{\lambda}}^t, \mathbf{u}, \mathbf{p}, \mathbf{p}^t)$ are nonseparable across neighboring nodes. To propose a distributed approach, we fix the neighboring nodes $\mathbf{p}_j, \mathbf{u}_{i,j}, j \in \mathcal{N}_i$ during the $\mathbf{p}_i$ update process. The specific optimization problem is

$$\arg\min_{\mathbf{p}_i\in\mathcal{X}} G_i^t(\xi), \tag{18}$$

where

$$G_i^t(\xi) := G(\mathbf{p}_1^{t+1}, \ldots, \mathbf{p}_{i-1}^{t+1}, \xi, \mathbf{p}_{i+1}^t, \ldots, \mathbf{p}_N^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t).$$

By using the first-order optimality condition, the sequential update step for $\mathbf{p}_i$ step is

$$\begin{aligned} \mathbf{p}_i^{t+1} = \frac{1}{\tau + \sum_{j\in\mathcal{N}_i} \tilde{\lambda}_{i,j}^t} \Big(\sum_{\substack{j\in\mathcal{N}_i \bigcap \mathcal{V} \\ j<i}} \tilde{\lambda}_{i,j}^t \mathbf{p}_j^{t+1} + \sum_{\substack{j\in\mathcal{N}_i \bigcap \mathcal{V} \\ j>i}} \tilde{\lambda}_{i,j}^t \mathbf{p}_j^t \\ + \tau\mathbf{p}_i^t + \sum_{j\in\mathcal{N}_i} d_{i,j}\tilde{\lambda}_{i,j}^t \mathbf{u}_{i,j}^t + \sum_{l\in\mathcal{N}_i \bigcap \mathcal{A}} \tilde{\lambda}_{i,l}^t \mathbf{a}_l \Big). \end{aligned} \tag{19}$$

The complete procedure is described in Algorithm 1.

In summary, to solve the nonconvex and nonseparable problem (2), we propose a novel distributed primal-dual optimization algorithm based on Lagrangian relaxation techniques. A key innovation involves constructing a tight upper bound G for the original objective function $\mathcal{J}_\varepsilon$. Moreover, function G exhibits convexity when the remaining variables are fixed. The following lemma shows that the constructed function G indeed possesses the aforementioned properties.

Lemma 2: Let $\mathcal{J}_\varepsilon$ be defined in (2), F as specified in (8), and G as defined in (17). Then, for any $\mathbf{p}, \mathbf{p}' \in \mathbb{R}^{nN}$, $\mathbf{v} \in \mathcal{B}^M$, $\boldsymbol{\mu} \in \mathbb{R}_+^{nN}$, and $\mathbf{u}, \boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}$ defined in (20), (11), and (13), respectively, the following hold:

1) $\mathcal{J}_\varepsilon(\mathbf{p}) = F(\boldsymbol{\lambda}, \mathbf{p}) = G(\mathbf{p}, \mathbf{u}, \tilde{\boldsymbol{\lambda}}, \mathbf{p})$,
2) $\mathcal{J}_\varepsilon(\mathbf{p}) = F(\boldsymbol{\lambda}, \mathbf{p}) \leq G(\mathbf{p}, \mathbf{v}, \boldsymbol{\mu}, \mathbf{p}')$.

Combining problem (12) with the forms of $\mathcal{J}_\varepsilon$, F, G, and incorporating the update steps for $\mathbf{u}$, , $\boldsymbol{\lambda}$, and $\tilde{\boldsymbol{\lambda}}$ from (20), (11), and (13), respectively, Lemma 2 can be readily verified.

Remark 1 (Computation, Communication, and Storage): Note that the computation cost of Algorithm 1 is primarily incurred in steps (13) and (20) , where the Euclidean norm $\|\mathbf{p}_i - \mathbf{p}_j\|$, $(i,j) \in \mathcal{E}$ is calculated. As a result, given $|\mathcal{E}| = M$ and $\mathbf{p}_i \in \mathbb{R}^n$, the total computation cost is characterized by $\mathcal{O}(nM)$. Regarding communication cost, during each iteration t, the algorithm necessitates that each node i transmits to its neighbors $j, (i,j) \in \mathcal{E}$ a vector with the position estimate $\mathbf{p}_i^t \in \mathbb{R}^n$. The proximal coefficient $\tilde{\lambda}_{i,j}$ and auxiliary variables $\mathbf{u}_{i,j}$ are computed internally at each node. We evaluate the storage space required at node i for different algorithms, assuming the storage unit occupied by any real number is one. Detailed comparisons between the proposed Algorithm 1 and other methods are presented in Table I.

Remark 2 (Comparison with: AM-FD): We establish a close connection between Algorithm 1 and the AM-FD method [23]. Specifically, by setting $\tau = 0$ (no proximal term) and choosing ε to be sufficiently large (LS objective function), we obtain $\tilde{\boldsymbol{\lambda}}^t = 1$ for all $t \geq 1$. As a result, expressions (19) and (20) have an identical form to the AM-FD algorithm in [[23], Eqn. (12)-(13)]. Consequently, Algorithm 1 can be viewed as a robust variant of AM-FD, offering additional capabilities to address localization problems with outliers. It is noteworthy that, in contrast to the constant coefficient of AM-FD, the robustness of Algorithm 1 stems from a more dynamic choice of coefficients $\tilde{\lambda}_{i,j}$ and is closely related to the weighted LS formulation [8], [11]. Additionally, using the matrix notations related to network structure defined in [23], problem (17) can also be reformulated in a compact form. This enables the development of a centralized method, where the step (19) can be transformed into a centralized recursive process.

IV. Convergence Analysis

In this section, we provide a convergence analysis of Algorithm 1. For simplicity, we define $\mathbf{v}^t := (\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t)$ for all $t \geq 1$. The results required to prove Theorem 1 are presented below:

Algorithm 1 Distributed Primal-Dual Algorithm

Initialization: $\mathbf{p}^0 \in \mathbb{R}^{nN}$, $\mathbf{u}^0 \in \mathcal{B}^M$, $\tilde{\boldsymbol{\lambda}}^0 \in \mathbb{R}^{nM}$.
For $t = 1, 2, \ldots$ **do**
  For $i = 1, 2, \ldots, N$ **do**
    Update $\mathbf{p}_i^{t+1}$ via (19)
  end
  For all $(i, j) \in \mathcal{E}$ **do**
    $\mathbf{p}_j^{t+1} = \mathbf{a}_j, \ \forall\ j \in \mathcal{A}$
    Update $\tilde{\lambda}_{i,j}^{t+1}$ via (13)

$$\mathbf{u}_{i,j}^{t+1} = \begin{cases} \frac{\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}}{\|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}\|}, & \mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1} \neq \mathbf{0}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{20}$$

    $\tilde{\lambda}_{j,i}^{t+1} = -\tilde{\lambda}_{i,j}^{t+1}, \ \mathbf{u}_{j,i}^{t+1} = -\mathbf{u}_{i,j}^{t+1}$
  end
end

Lemma 3 (Sufficient decrease): Let $\{\mathbf{v}^t\}_{t\geq 1}$ be a sequence generated by Algorithm 1. Then, the sequence $\{G(\mathbf{v}^t)\}_{t\geq 0}$ satisfies

$$G\left(\mathbf{v}^{t+1}\right) - G\left(\mathbf{v}^t\right) \leq -\left(\frac{\tau}{2} - 2N_{\max}\right)\|\mathbf{p}^{t+1} - \mathbf{p}^t\|^2,$$

where $N_{\max} = \max\{N_i, i \in \mathcal{V}\}$, and N_i denotes the number of adjacent nodes to node i.

Proof: See Appendix B. ∎

Lemma 4 (Subgradient bound): The sequence $\{\mathbf{v}^t\}_{t\geq 1}$ generated by Algorithm 1 is bounded, and there exists a scalar $\rho > 0$ and a vector $\mathbf{w}^t \in \partial G(\mathbf{v}^t)$ for all $t \in \mathbb{N}$, such that

$$\|\mathbf{w}^t\| \leq \rho\|\mathbf{p}^{t+1} - \mathbf{p}^t\|^2. \tag{21}$$

The proof of Lemma 4 employs a similar argument as in [[23], Lemma 5-6], with omitted details due to space constraints.

Lemma 5 (Relationship between G and $\mathcal{J}_\varepsilon$): Let $\{\mathbf{v}^t\}_{t\geq 1}$ be a sequence generated by Algorithm 1. Then the gradient of $\mathcal{J}_\varepsilon(\mathbf{p})$ at $\mathbf{p}_i, i \in \mathcal{V}$ is

$$\nabla_{\mathbf{p}_i}\mathcal{J}_\varepsilon(\mathbf{p}) = \sum_{j\in\mathcal{N}_i} \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|} \cdot \begin{cases} \varepsilon\,\mathrm{sgn}(e_{i,j}), & |e_{i,j}| > \varepsilon, \\ e_{i,j}, & |e_{i,j}| \leq \varepsilon. \end{cases}$$

where $e_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\| - d_{i,j}$, and

$$\mathbf{0} \in \partial G(\mathbf{v}^t) \Longleftrightarrow \mathbf{0} \in \nabla\mathcal{J}_\varepsilon(\mathbf{p}^t).$$

Lemma 5 can be verified by examining the expressions for G and $\mathcal{J}_\varepsilon$, along with the update steps of $(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t)$.

Note that the function G is semi-algebraic (a polynomial function) and possesses the KŁ property (see [24], Theorem 6.1). Armed with the above Lemmas and invoking the uniform KŁ property [22], we have the following global convergence result for Algorithm 1.

Theorem 1 (Global convergence for the whole sequence): Let $\tau \geq 4N_{\max}$. Then the sequence $\{\mathbf{p}^t\}_{t\geq 1}$ generated by Algorithm 1 converges to a critical point of the problem (2), and it has finite length, i.e., $\sum_{t=1}^{\infty}\|\mathbf{p}^{t+1} - \mathbf{p}^t\| < +\infty$.

Exploiting KŁ property, Theorem 1 establishes the global convergence of Algorithm 1 for the nonconvex localization problem (2). This result is attributed to the innovative formulation of the function G as defined in (17). The imposed condition on the penalty parameter τ serves the purpose of ensuring a sufficient decrease in the function value of G. Notably, in our experimental investigations, we observed an interesting aspect: the penalty parameter τ can be flexibly chosen, even when deviating from the condition in Theorem 1 (set $\tau = 0$ in experiments). This intriguing finding prompts us to identify this gap between theory and practice for further research.

V. Numerical Results

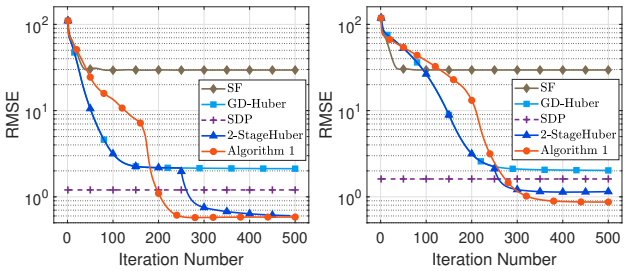
In this section, we present an empirical evaluation of Algorithm 1. To gauge its efficacy, we conduct a comparison with several established methods, including the semidefinite program (SDP) relaxation method outlined in [4], the SF method introduced in [5], the GD-Huber method described in [13], and the 2-StageHuber method elucidated in [15]. It is worth noting that the SDP and SF methods are specifically designed to address the convex relaxation version of problem (1), whereas the GD-Huber and 2-StageHuber methods target the convex relaxation version of problem (2).

The experimental setup considers a network with $m = 8$ and $N = 100$, as referenced in [9], [25]. The network is distributed across a $100 \times 100m^2$ area. The anchor positions are given by $\mathbf{a}_1 = [0, 0]^T$, $\mathbf{a}_2 = [50, 0]^T$, $\mathbf{a}_3 = [100, 0]^T$, $\mathbf{a}_4 = [0, 50]^T$, $\mathbf{a}_5 = [0, 100]^T$, $\mathbf{a}_6 = [50, 100]^T$, $\mathbf{a}_7 = [100, 100]^T$, $\mathbf{a}_8 = [100, 50]^T$. The communication range is set to $30m$, and measurement errors were generated with Gaussian distribution $\mathcal{N}\left(0, \sigma_{i,j}^2\right)$, where $\sigma_{i,j} = 1$. Consistent with [15], outliers are modeled as exponential random variables with parameter $\gamma = 10m$. The percentage of outlier links is denoted as P_{NLOS}. The initial sensor positions for all algorithms are selected from a uniform distribution $\text{Unif}(-100, 100)^{nN}$. The SDP solver employed here utilizes SeDuMi. Simulation parameters for Algorithm 1 are set as $\varepsilon = 5$ and $\tau = 0$. For the GD-Huber method, the parameters are set as $K = 5, \gamma = 0.04$ (notations in Section 4.1 of [13]). For the 2-StageHuber method, the parameters are set as $\mu_1 = 0.04, K_1 = 5, \mu_2 = 0.01, K_2 = 0.1\sigma_{i,j}$ (notations in Section IV of [15]). The chosen performance metric is the Root Mean Square Error (RMSE), as defined in [3] :

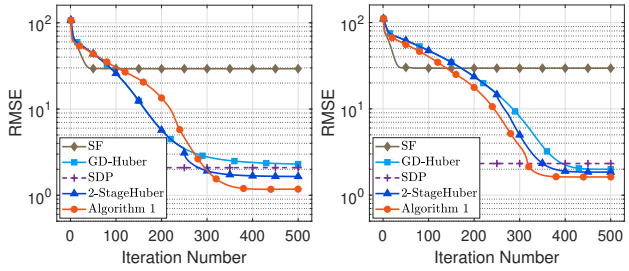
$$\text{RMSE}(t) := \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{p}_i^t - \mathbf{p}_i\|^2},$$

where $\mathbf{p}_i$ is the true position of node i and $\mathbf{p}_i^t$ is the estimated position of node i in the t-th iteration of the algorithm.

Fig. 2 presents a comparative analysis of the localization performance across various methods, depicting the RMSE concerning the iteration number at different values of P_{NLOS}. As we can see, the SF method is more affected by the quality of range measurements than other methods, possibly attributed to its focus on LS estimation, which is inherently sensitive

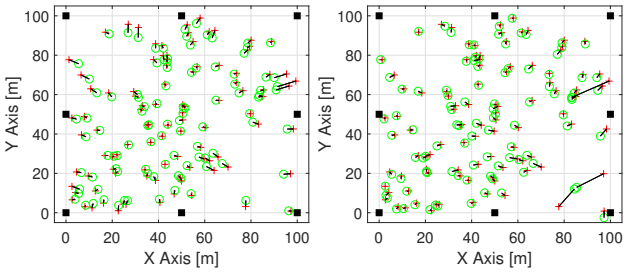


(a) $P_{\text{NLOS}} = 1\%$ (b) $P_{\text{NLOS}} = 5\%$

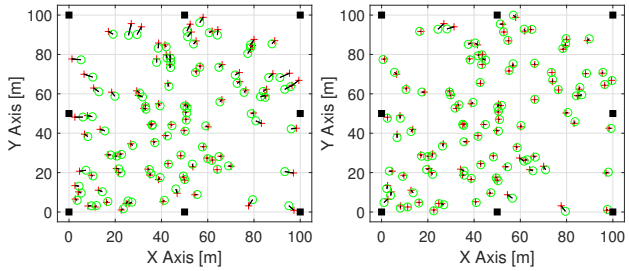


(c) $P_{\text{NLOS}} = 10\%$ (d) $P_{\text{NLOS}} = 20\%$

Fig. 2. Convergence performance versus the number of iterations for different outlier levels.



(a) GD-Huber (b) 2-StageHuber



(c) SDP (d) Algorithm 1

Fig. 3. Localization performance with $P_{\text{NLOS}} = 10\%$ obtained by the GD-Huber, 2-StageHuber, SDP, and proposed Algorithm 1. Here, the anchors are marked by ■, the true agent positions by + and the estimated ones by ○.

to outliers. Conversely, the SDP relaxation method, also an LS relaxation approach, exhibits greater robustness. However, the computational complexity of SDP relaxation method is $\mathcal{O}(n^3)$, resulting in significant computational time, particularly with large network sizes N. Table II presents the running times of the methods illustrated in Fig. 2. Despite the SDP

TABLE II
A COMPARISON OF FIVE METHODS IN TERMS OF RUNNING TIME

Algorithms	Run time (seconds) for different values of P_{NLOS}			
	1%	5%	10%	20%
SF [5]	2.05	1.98	1.47	1.51
SDP [4]	27.37(100)*	19.70(102)*	29.66(100)*	18.80(91)*
GD-Huber [13]	9.69	9.48	9.90	9.52
2-StageHuber [15]	199.55	216.80	252.87	199.14
Algorithm 1	**11.20**	**11.22**	**12.20**	**10.37**

* The $(\cdot)$ in the SDP method are the number of iteration steps that the SDP method actually runs.

method requiring fewer iterations than other methods, it runs approximately twice as long as our proposed method. It is also noteworthy that while the SF method exhibits a competitive running time compared to ours, its RMSE is significantly higher. In contrast, the Huber convex relaxation methods (GD-Huber and 2-StageHuber) display a rapid reduction in RMSE during the initial iterations due to the nonconvex nature of problem (2) with poor initialization. Notably, this reduction becomes less pronounced as the value of P_{NLOS} increases, evidenced by the narrowing gap in Fig. 2(a), (b), and (c), ultimately disappearing in Fig. 2(d). Moreover, the proposed Algorithm 1 consistently obtains higher accuracy compared to other methods once convergence is reached in all scenarios. This empirical observation lends support to the global convergence claimed in Theorem 1 for our proposed Algorithm 1.

Fig. 3 shows the spatial distribution of anchor positions, true sensor locations, and the final position estimates as depicted in Fig. 2(c). This visualization clearly demonstrates that the positions estimated by our Algorithm 1 are very close to the true sensors' locations. Furthermore, Table I confirms that the proposed Algorithm 1 achieves superior performance without incurring additional computational or communication costs in comparison to the SF, GD-Huber, and 2-StageHuber methods.

VI. Conclusions

In this paper, we introduced an efficient distributed primal-dual algorithm designed to address the cooperative localization problem, especially in scenarios with the presence of outliers. By leveraging the KŁ-property, we have established the global convergence of the algorithm to a critical point of the original nonconvex problem (as demonstrated in Theorem 1). Our empirical investigations have validated the potential of the proposed Algorithm 1 in terms of localization accuracy, computational efficiency, communication costs, and storage space (as detailed in Table I). A promising future direction involves determining the convergence rate of our proposed Algorithm 1 by further exploiting the KŁ property. Another possible future work is the design of an adaptive update method for the parameter ε, aimed at enhancing both the positioning accuracy and generalization ability of the algorithm for different networks.

ACKNOWLEDGMENT

This work was supported in part by the NSFC under Grant 62203313, in part by the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC1853, and in part by Sichuan Youth Science and Technology Innovation Team (Grant Nos. 2022JDTD0014, 2021JDJQ0036). Xiaojing Shen is the corresponding author.

APPENDIX A
PROOF OF LEMMA 1

The conjugate of the function $j_\varepsilon(x)$ is defined as

$$j_\varepsilon^*(y) = \sup_{x\in \mathrm{dom} j_\varepsilon} \{yx - j_\varepsilon(x)\}. \tag{22}$$

We analyze two cases.

Case 1: $|x| \le \varepsilon$. By referencing (3), we deduce that

$$\sup_{|x|\le\varepsilon}\{yx - \frac{1}{2}x^2\} = \begin{cases} -\varepsilon y - \frac{1}{2}\varepsilon^2, & y < -\varepsilon, \\ \frac{1}{2}y^2, & |y| \le \varepsilon, \\ \varepsilon y - \frac{1}{2}\varepsilon^2, & y > \varepsilon. \end{cases} \tag{23}$$

Case 2: $|x| > \varepsilon$. By employing (3) once more, we obtain

$$\begin{aligned} \sup_{|x|>\varepsilon}\{yx - \varepsilon|x| + \frac{1}{2}\varepsilon^2\} =& \frac{1}{2}\varepsilon^2 + \sup_{|x|>\varepsilon}\{yx - \varepsilon|x|\} \\ =& \begin{cases} -\varepsilon y - \frac{1}{2}\varepsilon^2, & -\varepsilon \le y \le 0, \\ \varepsilon y - \frac{1}{2}\varepsilon^2, & 0 \le y \le \varepsilon, \\ +\infty, & |y| > \varepsilon. \end{cases} \end{aligned} \tag{24}$$

By combining (22)-(24), we arrive at

$$\begin{aligned} j_\varepsilon^*(y) &= \max\{\sup_{|x|\le\varepsilon}\{yx - \frac{1}{2}x^2\}, \sup_{|x|>\varepsilon}\{yx - \varepsilon|x| + \frac{1}{2}\varepsilon^2\}\} \\ &= \begin{cases} \frac{1}{2}y^2, & \text{if } |y| \le \varepsilon, \\ +\infty, & \text{if } |y| > \varepsilon. \end{cases} \end{aligned}$$

Thus, the proof is complete.

APPENDIX B
PROOF OF LEMMA 3

We begin with three preliminary lemmas.

Lemma 6: Let $\{(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t)\}_{t\ge1}$ be a sequence generated by the Algorithm 1. Then we have

$$G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) \le G(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) - \frac{\tau}{2}\|\mathbf{p}^{t+1} - \mathbf{p}^t\|^2. \tag{25}$$

Proof: From (18), we conclude that

$$\begin{aligned} G_1^t(\mathbf{p}_1^{t+1}) + \frac{\tau}{2}\|\mathbf{p}_1^{t+1} - \mathbf{p}_1^t\|^2 \le& G_1^t(\mathbf{p}_1^t), \\ G_2^t(\mathbf{p}_2^{t+1}) + \frac{\tau}{2}\|\mathbf{p}_2^{t+1} - \mathbf{p}_2^t\|^2 \le& G_2^t(\mathbf{p}_2^t), \\ & \cdots \\ G_N^t(\mathbf{p}_N^{t+1}) + \frac{\tau}{2}\|\mathbf{p}_N^{t+1} - \mathbf{p}_N^t\|^2 \le& G_N^t(\mathbf{p}_N^t). \end{aligned} \tag{26}$$

and leveraging the definition of $G_i^t(\mathbf{p}_i^t)$, we obtain

$$G_i^t(\mathbf{p}_i^{t+1}) = G_{i+1}^t(\mathbf{p}_i^t),\ \forall\ 1 \le i \le N-1.$$

in addition, it is evident that

$$\begin{aligned} G_1^t(\mathbf{p}_1^t) =& G(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t), \\ G_N^t(\mathbf{p}_N^{t+1}) =& G(\mathbf{p}^{t+1}, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t). \end{aligned} \tag{27}$$

Therefore, substituting (27) into the cumulative result obtained by summing (26) leads to

$$G(\mathbf{p}^{t+1}, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) + \frac{\tau}{2}\|\mathbf{p}^{t+1} - \mathbf{p}^t\|^2 \le G(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t).$$

Furthermore, from the updating rule of the $\mathbf{u}$ we have that

$$G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) \le G(\mathbf{p}^{t+1}, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t).$$

Summing the last two inequalities establishes (25). ∎

Lemma 7: Consider the sequence $\{(\mathbf{p}^t, \mathbf{u}^t, \tilde{\boldsymbol{\lambda}}^t)\}_{t\ge1}$ generated by the Algorithm 1. Then, we have

$$\begin{aligned} & G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^{t+1}, \mathbf{p}^{t+1}) - G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) \\ =& \sum_{(i,j)\in\mathcal{E}} (|e_{i,j}^{t+1}| - |e_{i,j}^t|)^2, \end{aligned} \tag{28}$$

where $e_{i,j}^{t+1} = \|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1}\| - d_{i,j}, \forall\,(i,j) \in \mathcal{E}$.

Proof: By utilizing the definition of G as presented in (17), we have

$$\begin{aligned} & G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^{t+1}, \mathbf{p}^{t+1}) - G(\mathbf{p}^{t+1}, \mathbf{u}^{t+1}, \tilde{\boldsymbol{\lambda}}^t, \mathbf{p}^t) \\ =& \sum_{(i,j)\in\mathcal{E}} \underbrace{\frac{\tilde{\lambda}_{i,j}^{t+1}(1 - \tilde{\lambda}_{i,j}^{t+1})}{2}((e_{i,j}^{t+1})^2 - (e_{i,j}^t)^2)}_{a} \\ & + \sum_{(i,j)\in\mathcal{E}} \underbrace{\Big(1 - \frac{\tilde{\lambda}_{i,j}^{t+1} + \tilde{\lambda}_{i,j}^t}{2}\Big)(e_{i,j}^{t+1})^2(\tilde{\lambda}_{i,j}^{t+1} - \tilde{\lambda}_{i,j}^t)}_{b}. \end{aligned}$$

Consider the following cases:

Case 1: If $|e_{i,j}^{t+1}| > \varepsilon,\ |e_{i,j}^t| > \varepsilon$. Then by the definition of $\tilde{\lambda}_{i,j}$ in (13), we obtain $\tilde{\lambda}_{i,j}^{t+1} = \frac{\varepsilon}{|e_{i,j}^{t+1}|},\ \tilde{\lambda}_{i,j}^t = \frac{\varepsilon}{|e_{i,j}^t|}$, leading to

$$\begin{aligned} & a + b \\ =& \frac{1}{2}\left(|e_{i,j}^t| - |e_{i,j}^{t+1}|\right)\Big[\Big(\tilde{\lambda}_{i,j}^t - \tilde{\lambda}_{i,j}^{t+1}\Big)|e_{i,j}^{t+1}|\Big(1 - \tilde{\lambda}_{i,j}^t - \tilde{\lambda}_{i,j}^{t+1}\Big) \\ & + \Big(|e_{i,j}^t|\tilde{\lambda}_{i,j}^{t+1} - |e_{i,j}^{t+1}|\tilde{\lambda}_{i,j}^t\Big)\Big(\tilde{\lambda}_{i,j}^{t+1} - 1\Big)\Big]. \end{aligned} \tag{29}$$

since

$$\begin{aligned} \Big(\tilde{\lambda}_{i,j}^t - \tilde{\lambda}_{i,j}^{t+1}\Big)|e_{i,j}^{t+1}| =& -\left(|e_{i,j}^t| - |e_{i,j}^{t+1}|\right)\tilde{\lambda}_{i,j}^t, \\ |e_{i,j}^t|\tilde{\lambda}_{i,j}^{t+1} - |e_{i,j}^{t+1}|\tilde{\lambda}_{i,j}^t =& \left(|e_{i,j}^t| - |e_{i,j}^{t+1}|\right)\Big(\tilde{\lambda}_{i,j}^{t+1} + \tilde{\lambda}_{i,j}^t\Big). \end{aligned}$$

Substituting the above equation into (29) and combining with $\tilde{\lambda}_{i,j}^t < 1$, $\tilde{\lambda}_{i,j}^{t+1} < 1$, and $\tilde{\lambda}_{i,j}^t + \tilde{\lambda}_{i,j}^{t+1} - 1 < 1$, we get

$$a + b < \frac{1}{2}\left(|e_{i,j}^t| - |e_{i,j}^{t+1}|\right)^2.$$

Case 2: If $|e_{i,j}^{t+1}| > \varepsilon \geq |e_{i,j}^{t}|$. Then by the definition of $\tilde{\lambda}_{i,j}$ in (13), we obtain $\tilde{\lambda}_{i,j}^{t+1} = \frac{\varepsilon}{|e_{i,j}^{t+1}|} < \tilde{\lambda}_{i,j}^{t} = 1$. Thus $b < 0$ and

$$\begin{aligned} a + b \leq & \, a \\ =& \frac{\varepsilon}{2}(1 - \tilde{\lambda}_{i,j}^{t} + \tilde{\lambda}_{i,j}^{t} - \tilde{\lambda}_{i,j}^{t+1})(|e_{i,j}^{t+1}| - \frac{|e_{i,j}^{t}|^2}{|e_{i,j}^{t+1}|}) \\ =& \frac{\varepsilon}{2} \cdot \frac{|e_{i,j}^{t+1}| - \varepsilon}{|e_{i,j}^{t+1}|} \cdot \frac{(|e_{i,j}^{t+1}| - |e_{i,j}^{t}|)(|e_{i,j}^{t+1}| + |e_{i,j}^{t}|)}{|e_{i,j}^{t+1}|} \\ \leq& (|e_{i,j}^{t+1}| - |e_{i,j}^{t}|)^2, \end{aligned}$$

where the last inequality holds because $\frac{\varepsilon}{|e_{i,j}^{t+1}|} < 1$, $|e_{i,j}^{t+1}| - \varepsilon \leq |e_{i,j}^{t+1}| - |e_{i,j}^{t}|$, and $\frac{|e_{i,j}^{t+1}| + |e_{i,j}^{t}|}{|e_{i,j}^{t+1}|} \leq 2$.

Case 3: If $|e_{i,j}^{t}| > \varepsilon > |e_{i,j}^{t+1}|$. According to (13), we know that $\tilde{\lambda}_{i,j}^{t} = \frac{\varepsilon}{|e_{i,j}^{t}|} < \tilde{\lambda}_{i,j}^{t+1} = 1$, which implies $a < 0$, therefore,

$$\begin{aligned} a + b \leq & \, b \\ =& \frac{|e_{i,j}^{t}| - \varepsilon}{2|e_{i,j}^{t}|} \cdot (e_{i,j}^{t+1})^2 \cdot \frac{|e_{i,j}^{t}| - \varepsilon}{|e_{i,j}^{t}|} \\ \leq& \frac{1}{2}(|e_{i,j}^{t}| - |e_{i,j}^{t+1}|)^2, \end{aligned}$$

where the last inequality holds due to $|e_{i,j}^{t}| - \varepsilon \leq |e_{i,j}^{t}| - |e_{i,j}^{t+1}|$, and $\frac{(e_{i,j}^{t+1})^2}{(e_{i,j}^{t})^2} \leq 1$.

Case 4: If $|e_{i,j}^{t}| \leq \varepsilon,\ |e_{i,j}^{t+1}| \leq \varepsilon$. Then we obtain $\tilde{\lambda}_{i,j}^{t} = \tilde{\lambda}_{i,j}^{t+1} = 1$, it follows that $a + b = 0$.

Combining theses bounds concludes the proof. ∎

Now, we are ready to derive the result of Lemma 3. By applying the triangle inequality and using the definition of $e_{i,j}$ in (28), we have

$$\begin{aligned} ||e_{i,j}^{t}| - |e_{i,j}^{t+1}|| \leq |e_{i,j}^{t} - e_{i,j}^{t+1}| \leq & \|\mathbf{p}_i^{t+1} - \mathbf{p}_j^{t+1} - \left(\mathbf{p}_i^{t} - \mathbf{p}_j^{t}\right)\| \\ \leq & \|\mathbf{p}_i^{t+1} - \mathbf{p}_i^{t}\| + \|\mathbf{p}_j^{t+1} - \mathbf{p}_j^{t}\|. \end{aligned}$$

Summing over $(i,j) \in \mathcal{E}$ and employing the Cauchy-Schwarz inequality, it follows that

$$\sum_{(i,j)\in\mathcal{E}} (|e_{i,j}^{t}| - |e_{i,j}^{t+1}|)^2 \leq 2N_{\max}\|\mathbf{p}^{t+1} - \mathbf{p}^{t}\|^2, \qquad (30)$$

where $N_{\max}$ is defined in Lemma 3. Combining (28) with (25) and (30), then we proved Lemma 3.

REFERENCES

[1] P. K. Singh, B. K. Bhargava, M. Paprzycki, N. C. Kaushal, and W.-C. Hong, eds., *Handbook of wireless sensor networks: issues and challenges in current Scenario's*, vol. 1132. Springer, 2020.

[2] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, "Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1448–1463, 2015.

[3] M. Zhang, Z. Wang, F. Yin, and X. Shen, "Distributed scaled proximal ADMM algorithms for cooperative localization in WSNs," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3312 – 3327, 2023.

[4] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 188–220, 2006.

[5] C. Soares, J. Xavier, and J. Gomes, "Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4532–4543, 2015.

[6] M. Safaei, S. Asadi, M. Driss, W. Boulila, A. Alsaeedi, H. Chizari, R. Abdullah, and M. Safaei, "A systematic literature review on outlier detection in wireless sensor networks," *Symmetry*, vol. 12, no. 3, p. 328, 2020.

[7] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[8] S. Korkmaz and A.-J. van der Veen, "Robust localization in sensor networks with iterative majorization techniques," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2049–2052, 2009.

[9] D. Jin, F. Yin, A. M. Zoubir, and H. C. So, "Exploiting sparsity of ranging biases for NLOS mitigation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3782–3795, 2021.

[10] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[11] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 39–64, 2006.

[12] P. He, X. Jiang, and Q. Shi, "Robust TOA-based source self-positioning with clock imperfection," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2020.

[13] E. M. A. Etiabi, Yaya and E. Sabir, "A distributed and collaborative localization algorithm for internet of things environments," in *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, pp. 114–118, 2020.

[14] E. M. A. Etiabi, Yaya and E. Sabir, "Huber estimator and statistical bootstrap based light-weight localization for IoT systems," in *Ubiquitous Networking: 7th International Symposium, UNet 2021*, pp. 79–92, Springer, 2021.

[15] S. Yousefi, X.-W. Chang, and B. Champagne, "Distributed cooperative localization in wireless sensor networks without NLOS identification," *2014 11th Workshop on Positioning, Navigation and Communication (WPNC)*, pp. 1–6, 2014.

[16] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[17] Z. Wang, D. Li, S. Wu, Y. Huang, Z. Yang, and W. Nai, "Huber loss function based on cockroach swarm algorithm with T-distribution parameters," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 2490–2493, IEEE, 2021.

[18] C. Soares and J. Gomes, "STRONG: Synchronous and asynchronous robust network localization, under non-Gaussian noise," *Signal Processing*, vol. 185, p. 108066, 2021.

[19] W. Xiong, S. Mohanty, C. Schindelhauer, S. J. Rupitsch, and H. C. So, "Convex relaxation approaches to robust RSS-TOA based source localization in NLOS environments," *IEEE Transactions on Vehicular Technology*, vol. 72, pp. 11068 – 11073, 2023.

[20] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[21] L. Ding, A. Yurtsever, V. Cevher, J. A. Tropp, and M. Udell, "An optimal-storage approach to semidefinite programming using approximate complementarity," *SIAM Journal on Optimization*, vol. 31, no. 4, pp. 2695–2725, 2021.

[22] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[23] E. Gur, S. Sabach, and S. Shtern, "Alternating minimization based first-order method for the wireless sensor network localization problem," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6418–6431, 2020.

[24] J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd, "First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2131–2151, 2018.

[25] D. Jin, F. Yin, M. Fauß, M. Muma, and A. M. Zoubir, "Exploiting sparsity for robust sensor network localization in mixed LOS/NLOS environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5915–5915, IEEE, 2020.