Distributed Optimization for Generalized Phase Retrieval Over Networks

Ziping Zhao*[1], Songtao Lu†[2], Mingyi Hong†[3], and Daniel P. Palomar*[4]
*Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong
†Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, United States
Email: {[1]ziping.zhao, [4]palomar}@ust.hk; {[2]lus, [3]mhong}@umn.edu

***Abstract*—In this paper, we will solve the generalized phase retrieval (PR) problem over a network, where each agent only has a subset of the measurements. The problem is formulated as minimizing the squared loss between the measurements and linear sensing intensity. To solve the problem in a distributed setting, an algorithm named distributed Wirtinger flow (DWF) is proposed. Theoretical analyses show that the proposed DWF algorithm converges to the (approximate) KKT points of the original problem globally in a sublinear rate. The performance of the DWF algorithm is numerically compared with the state-of-the-art method. Simulation results show that DWF is able to recover a high-quality solution for the original PR problem.**

***Index Terms*—Quadratic systems, nonconvex optimization, decentralized optimization, distributed learning, statistical learning over networks.**

I. Introduction

The *phase retrieval (PR)* problem has received intensive investigations in recent years [1], [2]. It arises in various fields of science and engineering, such as optical imaging, X-ray crystallography, radar signal processing, communications, astronomy, and so on, when the underlying problem is to reconstruct a desired signal given only the magnitudes of its linear measurements [3]. Mathematically, the PR problem is aimed at finding the solutions to a quadratic system of equations. Suppose the interested solution or desired signal is denoted by $\mathbf{x} \in \mathbb{C}^K$, the PR problem is given as follows:

$$\begin{array}{ll} \mathsf{find} & \mathbf{x} \\ \mathsf{subject\ to} & y_t = \left|\langle \mathbf{a}_t, \mathbf{x}\rangle\right|^2, \; t = 1, 2, \ldots, T, \end{array} \tag{1}$$

where $y_t \in \mathbb{R}$ is the measurement, $\mathbf{a}_t \in \mathbb{C}^K$ is the sampling vector which is known beforehand, and $\langle \mathbf{a}, \mathbf{b}\rangle \triangleq \mathbf{a}^H\mathbf{b}$. This problem is nonconvex and NP-hard in general [4].

In practice, the measurements y_t's are usually corrupted with noise. To recover signal $\mathbf{x}$, one commonly used formulation is based on the least squares error minimization between the observation y_t and the square modulus of the transformation $\langle \mathbf{a}_t, \mathbf{x}\rangle$. Then, the optimization problem can be formulated as follows

$$\underset{\mathbf{x}}{\mathsf{minimize}} \quad \frac{1}{2T}\sum_{t=1}^{T}\left(y_t - \left|\langle \mathbf{a}_t, \mathbf{x}\rangle\right|^2\right)^2 \quad , \tag{2}$$

This work was supported in part by the University of Minnesota and the Hong Kong University of Science and Technology via bilateral collaboration funds, and in part by the Sponsorship Scheme for Targeted Strategic Partnership FP602 of the Hong Kong University of Science and Technology. The work of Z. Zhao was supported by the Hong Kong PhD Fellowship Scheme.

which is also the Gaussian maximum likelihood estimation. It is easy to see that problem (2) is a nonconvex problem.

A number of nonconvex and convex relaxation based methods have been proposed to solve problem (2). In [4], the PhaseLift algorithm was proposed to recover the signal $\mathbf{x}$ based on semidefinite programming (SDP) through "matrix lifting" technique. Although SDP relaxations can offer tractable solutions, solving SDP will be computationally expensive as the dimension of the signal increases. In [5], the Wirtinger flow (WF) algorithm was developed by directly applying gradient descent for updating variable $\mathbf{x}$. As a variant of WF, the truncated WF algorithm was further given in [6]. In [7], the author proposed to use the Kaczmarz method to reduce the computational complexity per iteration. A majorization-minimization based method, termed as PRIME [8], was also proposed to gain a better convergence property by updating the variable via solving an eigenvalue decomposition problem per iteration.

In problem (1), the set of quadratic equations can be also equivalently rewritten as $\sqrt{y_t} = |\langle \mathbf{a}_t, \mathbf{x}\rangle|$. Then the optimization can be accordingly carried out by minimizing the difference between the amplitude measurement $\sqrt{y_t}$ and $|\langle \mathbf{a}_t, \mathbf{x}\rangle|$. In this setup, many algorithms have also been developed, such as the error reduction algorithms in [9] and [10], the AltMinPhase algorithm via alternating minimization in [11], the PhaseCut algorithm based on SDP in [12], and the amplitude flow algorithm based on gradient descent in [13].

These is another line of works which have provided geometric analysis for problem (2). By assuming the sampling vector $\mathbf{a}_t$ follows a random distribution or is deterministic (say, from FFT matrix), the global optimal solutions can be provably located under some mild conditions for this problem. The resulting algorithms can achieve the optimal (statistical) solutions with guarantees. For example, a global geometric structure of the PR problem was analyzed in [14]. Based on some prior assumption on the sensing vectors, all the Karush-Kuhn-Tucker (KKT) points of problem (2) are within a compact set $\mathcal{X}$ (with a high probability), which leads to a trust region based method that initializes the iterates within a ℓ_2-norm ball, i.e., $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|^2 \leq \tau\}$, where τ is a constant depending on the measurements (see [14, Theorem 3.10]).

It is worth noting that all the aforementioned algorithms for the PR problem in (2) are based on a set of centralized data. In practice, however, the centralized data may not be

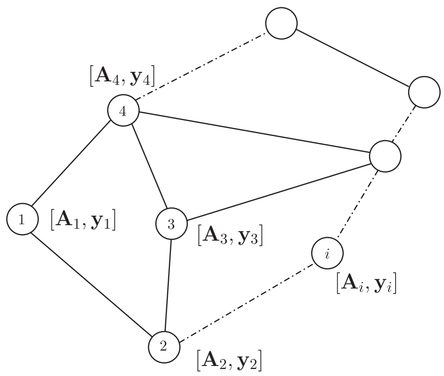

Fig. 1. Schematic representation of solving the phase retrieval problem over a distributed network. (Each agent i has only a subset of the data, i.e. $[\mathbf{A}_i, \mathbf{y}_i]$.)

obtainable due to the following reasons: i) the data is collected and stored in distributed sensing nodes; ii) some data privacy issue so that it might not be possible to collect the data into one center processor; iii) the measurement data is of a large scale (especially for image data), where dealing with all the measurements jointly can bring considerable computational overhead to the center processor. In fact, many works have discussed about the distributed data processing and optimization over networks [15], [16]. In particular, distributed algorithms have been developed by classical problems such as solving linear systems of equations [17], principal component analysis [18], regression analysis [19], dictionary learning [20], etc.

A natural and interesting question to ask is: *Can we solve the generalized phase retrieval problem over a distributed network?* This question is straightforward and motivated by real applications. For example, in optical imaging, the measurements are usually of high dimension in nature or acquired by distributed sensors [2]. Another example is the MIMO beampattern synthesis problem in array signal processing when the matching pattern is of large-dimension [21], [22].

In this paper, a distributed nonconvex optimization algorithm for the phase retrieval problem (2) named *distributed Wirtinger flow (DWF)* is proposed. The DWF algorithm is based on the general *perturbed proximal primal-dual method* proposed in [23]. Convergence to (approximate) Karush-Kuhn-Tucker (KKT) points is established by theoretical analyses. Numerical results show that the DWF algorithm is able to recover a high-quality solution for the original problem.

II. Problem Formulation

In this paper, we will focus on a distributed decentralized algorithm for solving problem (2). Consider a connected network with N agents defined by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as shown in Fig. 1, with $|\mathcal{V}| = N$ vertices and $|\mathcal{E}| = M$ edges. The standard graph (signed) incidence matrix is denoted as $\mathbf{B}_- \in \mathbb{R}^{M\times N}$ and defined as follows: if $e \in \mathcal{E}$ and it connects vertex i and j with $i > j$, then $[\mathbf{B}_-]_{e,v} = 1$ if $v = i$, and $[\mathbf{B}_-]_{e,v} = -1$ if $v = j$, otherwise $[\mathbf{B}_-]_{e,v} = 0$. The signless incidence matrix is given as $\mathbf{B}_+ \triangleq |\mathbf{B}_-|$ where $|\cdot|$ means taking the elementwise absolute value of $\mathbf{B}_-$.

For agent $i \in \mathcal{V}$, it has the measurements $\mathbf{y}_i$ denoted by

$$\mathbf{y}_i \triangleq [y_{t_i}; y_{t_i+1}; \dots; y_{t_i+T_i}] \in \mathbb{R}^{T_i}, \tag{3}$$

where T_i denotes the sample length and y_{t_i} denotes the first measurement in $\mathbf{y}_i$. Accordingly, the sensing vectors for agent $i \in \mathcal{V}$ are denoted by

$$\mathbf{A}_i \triangleq [\mathbf{a}_{t_i}, \mathbf{a}_{t_i+1}, \dots, \mathbf{a}_{t_i+T_i}] \in \mathbb{C}^{K\times T_i}, \tag{4}$$

with $\sum_{i=1}^N T_i = T$. The neighborhood set of agent i is defined as $\mathcal{N}_i \triangleq \{i \in \mathcal{V} | j \in \mathcal{V}, (i,j) \in \mathcal{E}\}$. Each agent can only communicate with its neighbors and it is responsible for optimizing the local objective $f_i(\mathbf{x}_i) = \frac{1}{2}\left\|\mathbf{y}_i - |\langle \mathbf{A}_i, \mathbf{x}_i\rangle|^2\right\|^2$. Let $\mathbf{x}_i$ denote the signal recovered by the ith agent. The PR problem in (2) can be rewritten as follows:

$$\begin{aligned} \underset{\mathbf{x}_i \in \mathcal{X}}{\text{minimize}} \quad & \frac{1}{T}\sum_{i=1}^N f_i(\mathbf{x}_i) = \frac{1}{2T}\sum_{i=1}^N \left\|\mathbf{y}_i - |\langle \mathbf{A}_i, \mathbf{x}_i\rangle|^2\right\|^2 \\ \text{subject to} \quad & \mathbf{x}_i = \mathbf{x}_j, \ \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{5}$$

where $\mathcal{X}$ defined in [14] denotes a set that includes all the KKT points of problem (2).

Define the block-signed incidence matrix as $\overline{\mathbf{B}}_- \triangleq \mathbf{B}_- \otimes \mathbf{I}_K \in \mathbb{R}^{MK\times NK}$, and the block signless incidence matrix as $\overline{\mathbf{B}}_+ \triangleq |\overline{\mathbf{B}}_-|$. Due to the network connectivity, we use condition $\overline{\mathbf{B}}_-\overline{\mathbf{x}} = \mathbf{0}$ to denote the network-wide consensus. Let $\bar{\mathbf{x}} \triangleq [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N] \in \mathbb{C}^{NK}$. A compact problem formulation for problem (5) is given as follows:

$$\begin{aligned} \underset{\overline{\mathbf{x}}|\mathbf{x}_i \in \mathcal{X}}{\text{minimize}} \quad & \frac{1}{2T}\left\|\mathbf{y} - |\langle \overline{\mathbf{A}}, \overline{\mathbf{x}}\rangle|^2\right\|^2 \\ \text{subject to} \quad & \overline{\mathbf{B}}_-\overline{\mathbf{x}} = \mathbf{0}, \end{aligned} \tag{6}$$

where the measurement vector $\mathbf{y} \triangleq [\mathbf{y}_1; \mathbf{y}_2; \dots; \mathbf{y}_N] \in \mathbb{R}^T$, and the sensing matrix $\overline{\mathbf{A}} \triangleq \text{blkdiag}[\mathbf{A}] \in \mathbb{C}^{NK\times T}$ with $\mathbf{A} \triangleq [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N] \in \mathbb{C}^{K\times T}$.

III. DWF: Distributed Wirtinger Flow Algorithm

In this section, we will propose a primal-dual based algorithm to solve problem (6). The proposed algorithm is based on the general method for solving nonconvex distributed optimization problems which was proposed in [23]. However, this optimization problem (6) is with respect to complex variable. In this paper, we give a detailed derivation of the distributed algorithm for the complex case. Considering the distributed algorithm introduced in this paper is based on the Wirtinger derivatives [24], we name the proposed algorithm distributed Wirtinger flow (DWF).

The proposed algorithm builds upon the classical augmented Lagrangian method [25]. To this end, we define $f(\overline{\mathbf{x}}) \triangleq \sum_{i=1}^N f_i(\mathbf{x}_i)$ and the *perturbed augmented Lagrangian* for problem (6) as

$$\begin{aligned} & L_{\gamma,\rho}(\overline{\mathbf{x}}, \overline{\boldsymbol{\omega}}) \\ \triangleq & f(\overline{\mathbf{x}}) + \text{Re}\left(\langle (1-\rho\gamma)\overline{\boldsymbol{\omega}}, \overline{\mathbf{B}}_-\overline{\mathbf{x}}\rangle\right) + \frac{\rho}{2}\langle \overline{\mathbf{B}}_-\overline{\mathbf{x}}, \overline{\mathbf{B}}_-\overline{\mathbf{x}}\rangle, \end{aligned} \tag{7}$$

where $\overline{\boldsymbol{\omega}} \triangleq [\boldsymbol{\omega}_1; \boldsymbol{\omega}_2; \dots; \boldsymbol{\omega}_M] \in \mathbb{C}^{MK}$ is the dual variable vector with $\boldsymbol{\omega}_e \in \mathbb{R}^M$ being the variable for the consensus constraint on link $e=(i,j)$, i.e, $\mathbf{x}_i=\mathbf{x}_j$, ρ denotes the penalty parameter, and γ is the perturbation parameter on the dual variable satisfying $1-\rho\gamma>0$. Introducing the perturbation parameter is to control the size of dual update and, hence, guarantee the algorithm convergence. After the initialization of the algorithm, the updating rules of the primal and dual variables are introduced in the following.

A. The Primal Variable $\overline{\mathbf{x}}$ Update

The primal variable $\overline{\mathbf{x}}$ is updated by minimizing the perturbed augmented Lagrangian plus a proximal term which is given as follows:

$$\overline{\mathbf{x}}^{r+1} = \underset{\mathbf{x}_i\in\mathcal{X}}{\arg\min}\left\{L_{\gamma,\rho}\left(\overline{\mathbf{x}},\overline{\boldsymbol{\omega}}^r\right)+\underbrace{\frac{\rho}{2}\left\langle\overline{\mathbf{B}}_+\left(\overline{\mathbf{x}}-\overline{\mathbf{x}}^r\right),\overline{\mathbf{B}}_+\left(\overline{\mathbf{x}}-\overline{\mathbf{x}}^r\right)\right\rangle}_{\text{proximal term}}\right\}, \tag{8}$$

where the proximal term is added to ensure the strongly convexity of the primal problem and also make the problem becomes variable separable over the distributed nodes. Substituting $L_{\gamma,\rho}\left(\overline{\mathbf{x}},\overline{\boldsymbol{\omega}}\right)$ into (8), we have

$$\begin{aligned}\overline{\mathbf{x}}^{r+1} =& \underset{\mathbf{x}_i\in\mathcal{X}}{\arg\min}\Big\{f\left(\overline{\mathbf{x}}\right)+\mathrm{Re}\left(\left\langle(1-\rho\gamma)\,\overline{\boldsymbol{\omega}}^r,\overline{\mathbf{B}}_-\overline{\mathbf{x}}\right\rangle\right)\\ &+\frac{\rho}{2}\left\langle\overline{\mathbf{B}}_-\overline{\mathbf{x}},\overline{\mathbf{B}}_-\overline{\mathbf{x}}\right\rangle+\frac{\rho}{2}\left\langle\overline{\mathbf{B}}_+\left(\overline{\mathbf{x}}-\overline{\mathbf{x}}^r\right),\overline{\mathbf{B}}_+\left(\overline{\mathbf{x}}-\overline{\mathbf{x}}^r\right)\right\rangle\Big\}\\ =&\underset{\mathbf{x}_i\in\mathcal{X}}{\arg\min}\Big\{f\left(\overline{\mathbf{x}}\right)+\mathrm{Re}\left(\left\langle(1-\rho\gamma)\,\overline{\mathbf{B}}_-^T\overline{\boldsymbol{\omega}}^r,\overline{\mathbf{x}}\right\rangle\right)\\ &+\frac{\rho}{2}\left\langle\mathbf{L}_-\overline{\mathbf{x}},\overline{\mathbf{x}}\right\rangle+\frac{\rho}{2}\left\langle\mathbf{L}_+\overline{\mathbf{x}},\overline{\mathbf{x}}\right\rangle-\rho\mathrm{Re}\left(\left\langle\mathbf{L}_+\overline{\mathbf{x}}^r,\overline{\mathbf{x}}\right\rangle\right)\Big\},\end{aligned} \tag{9}$$

where $\mathbf{L}_-\triangleq\overline{\mathbf{B}}_-^T\overline{\mathbf{B}}_-\in\mathbb{R}^{NK\times NK}$ is the the block-signed Laplacian matrix, $\mathbf{L}_+\triangleq\overline{\mathbf{B}}_+^T\overline{\mathbf{B}}_+\in\mathbb{R}^{NK\times NK}$ is the block-signless Laplacian matrix. Let $\mathbf{D}\triangleq\mathrm{diag}\left(\mathbf{d}\right)\otimes\mathbf{I}_K\in\mathbb{R}^{NK\times NK}$ be the block-degree matrix, where $\mathbf{d}\triangleq[d_1,d_2,\ldots,d_N]^T$ represents the degree vector. Since $\mathbf{D}=\frac{1}{2}\left(\mathbf{L}_-+\mathbf{L}_+\right)$, we have

$$\begin{aligned}\overline{\mathbf{x}}^{r+1} =& \underset{\mathbf{x}_i\in\mathcal{X}}{\arg\min}\Big\{f\left(\overline{\mathbf{x}}\right)+\mathrm{Re}\left(\left\langle(1-\rho\gamma)\,\overline{\mathbf{B}}_-^T\overline{\boldsymbol{\omega}}^r,\overline{\mathbf{x}}\right\rangle\right)\\ &+\rho\left\langle\mathbf{D}\overline{\mathbf{x}},\overline{\mathbf{x}}\right\rangle-\rho\mathrm{Re}\left(\left\langle\mathbf{L}_+\overline{\mathbf{x}}^r,\overline{\mathbf{x}}\right\rangle\right)\Big\}.\end{aligned} \tag{10}$$

However the nonconvexity of the objective $f\left(\overline{\mathbf{x}}\right)$ imposes on much difficulty of solving this primal problem exactly. Here, we consider a linear approximation for $f\left(\overline{\mathbf{x}}\right)$ to enable an easy update of $\overline{\mathbf{x}}$. The approximate surrogate function $\tilde{f}\left(\overline{\mathbf{x}};\overline{\mathbf{x}}^r\right)$ at iterates $\overline{\mathbf{x}}^r$ for $f\left(\overline{\mathbf{x}}\right)$ is given as

$$\begin{aligned}&\tilde{f}\left(\overline{\mathbf{x}};\overline{\mathbf{x}}^r\right)\triangleq f\left(\overline{\mathbf{x}}^r\right)+\mathrm{Re}\left(\left\langle 2\nabla f\left(\overline{\mathbf{x}}^{*r}\right),\overline{\mathbf{x}}\right\rangle\right)\\ &=\mathrm{Re}\left(\left\langle\frac{2}{T}\overline{\mathbf{A}}\mathrm{diag}\left(\left|\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle\right|^2-\mathbf{y}\right)\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle,\overline{\mathbf{x}}\right\rangle\right)+const.,\end{aligned} \tag{11}$$

where $\nabla f\left(\overline{\mathbf{x}}^{*r}\right)$ is the Wirtinger derivative with respect to $\overline{\mathbf{x}}^*$ (conjugate of $\overline{\mathbf{x}}$) at point $\overline{\mathbf{x}}^{*r}$. By leveraging $\tilde{f}\left(\overline{\mathbf{x}};\overline{\mathbf{x}}^r\right)$, the primal variable update rule becomes a variable decoupled convex quadratic programming problem, i.e.,

$$\begin{aligned}&\overline{\mathbf{x}}^{r+1}\\ =&\underset{\mathbf{x}_i\in\mathcal{X}}{\arg\min}\Big\{\mathrm{Re}\Big(\Big\langle\frac{2}{T}\overline{\mathbf{A}}\mathrm{diag}\left(\left|\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle\right|^2-\mathbf{y}\right)\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle\\ &+(1-\rho\gamma)\,\overline{\mathbf{B}}_-^T\overline{\boldsymbol{\omega}}^r-\rho\mathbf{L}_+\overline{\mathbf{x}}^r,\mathbf{x}\Big\rangle\Big)+\rho\left\langle\mathbf{D}\overline{\mathbf{x}},\overline{\mathbf{x}}\right\rangle\Big\}\\ =&\mathrm{Proj}_{\mathcal{X}}\Big\{\frac{1}{2\rho}\mathbf{D}^{-1}\Big[-\frac{2}{T}\overline{\mathbf{A}}\mathrm{diag}\left(\left|\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle\right|^2-\mathbf{y}\right)\left\langle\overline{\mathbf{A}},\overline{\mathbf{x}}^r\right\rangle\\ &-(1-\rho\gamma)\,\overline{\mathbf{B}}_-^T\overline{\boldsymbol{\omega}}^r+\rho\mathbf{L}_+\overline{\mathbf{x}}^r\Big]\Big\},\end{aligned} \tag{12}$$

where $\mathrm{Proj}_{\mathcal{X}}\left\{\cdot\right\}$ is the projection operator.

More specifically, for every agent $i\in\mathcal{V}$, it suffices to update the variable at each agent according to the following rule

$$\begin{aligned}&\mathbf{x}_i^{r+1}\\ =&\mathrm{Proj}_{\mathcal{X}}\Big\{\frac{1}{2\rho}d_i^{-1}\Big[-\frac{2}{T}\mathbf{A}_i\mathrm{diag}\left(\left|\left\langle\mathbf{A}_i,\mathbf{x}_i^r\right\rangle\right|^2-\mathbf{y}_i\right)\left\langle\mathbf{A}_i,\mathbf{x}_i^r\right\rangle\\ &-(1-\rho\gamma)\Big(\sum_{e\in\mathcal{U}_i}\boldsymbol{\omega}_e^r-\sum_{e\in\mathcal{H}_i}\boldsymbol{\omega}_e^r\Big)+\rho\Big(d_i\mathbf{x}_i^r+\sum_{j\in\mathcal{N}_i}\mathbf{x}_j^r\Big)\Big]\Big\},\end{aligned} \tag{13}$$

where the set $\mathcal{U}_i\triangleq\{e\mid e=(i,j)\in\mathcal{E},i\geq j\}$ and $\mathcal{H}_i\triangleq\{e\mid e=(i,j)\in\mathcal{E},j\geq i\}$. Next, we will introduce the dual variable update.

B. The Dual Variable $\overline{\boldsymbol{\omega}}$ Update

Compared with the traditional dual update [25], the proposed dual variable $\overline{\boldsymbol{\omega}}$ is updated by taking a perturbed dual update rule, which is given by

$$\overline{\boldsymbol{\omega}}^{r+1}=(1-\rho\gamma)\,\overline{\boldsymbol{\omega}}^r+\rho\overline{\mathbf{B}}_-\overline{\mathbf{x}}^{r+1}. \tag{14}$$

Specifically, for the edge $e=(i,j)$ with $i>j$, the dual variable $\boldsymbol{\omega}_e$ is updated as follows:

$$\boldsymbol{\omega}_e^{r+1}=(1-\rho\gamma)\,\boldsymbol{\omega}_e^r+\rho\left(\mathbf{x}_i^{r+1}-\mathbf{x}_j^{r+1}\right) \tag{15}$$

where $\rho\gamma<1$ is the perturbed term.

C. Summary of Distributed Wirtinger Flow (DWF) Algorithm

The DWF algorithm can be summarized in Algorithm 1.

Algorithm 1 Distributed Wirtinger Flow (DWF) Algorithm

Require: $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$, $\mathbf{A}_i$, and $\mathbf{y}_i$ with $i\in\mathcal{V}$.

1: Set $r=0$, and initialize $\mathbf{x}_i^r$ for $i\in\mathcal{V}$, and $\boldsymbol{\omega}_e^r$ for $e\in\mathcal{E}$.
2: **repeat**
3: For $i\in\mathcal{V}$, update $\mathbf{x}_i^{r+1}$ according to Eq. (13);
4: For $e\in\mathcal{E}$, update $\boldsymbol{\omega}_e^{r+1}$ according to Eq. (15);
5: $r\leftarrow r+1$;
6: **until** convergence

IV. Convergence Analysis

The objective function in problem (2) is a mapping from $\mathbb{C}^K$ to $\mathbb{R}$, which is not holomorphic and therefore not complex differentiable [24]. We can equivalently rewrite the problem in the field of real-valued numbers where the problem is differential. The convergence result of the DWF algorithm will accordingly be established in this transferred domain.

For problem (2), define $\widetilde{\mathbf{x}} \triangleq [\mathrm{Re}(\mathbf{x}); \mathrm{Im}(\mathbf{x})] \in \mathbb{R}^{2K}$, and

$$\widetilde{\mathbf{A}}_t \triangleq \begin{bmatrix} \widetilde{\mathbf{A}}_{t,1} & \widetilde{\mathbf{A}}_{t,2} \\ \widetilde{\mathbf{A}}_{t,2}^T & \widetilde{\mathbf{A}}_{t,1} \end{bmatrix} \in \mathbb{R}^{2K \times 2K},$$

with $\widetilde{\mathbf{A}}_{t,1} = \mathrm{Re}(\mathbf{a}_t)\mathrm{Re}^T(\mathbf{a}_t) + \mathrm{Im}(\mathbf{a}_t)\mathrm{Im}^T(\mathbf{a}_t)$ and $\widetilde{\mathbf{A}}_{t,2} = \mathrm{Re}(\mathbf{a}_t)\mathrm{Im}^T(\mathbf{a}_t) - \mathrm{Im}(\mathbf{a}_t)\mathrm{Re}^T(\mathbf{a}_t)$, for $t = 1, 2, \ldots, T$. We can rewrite objective function $f(\mathbf{x})$ in terms of $\widetilde{\mathbf{x}}$ as

$$\widetilde{f}(\widetilde{\mathbf{x}}) \triangleq \frac{1}{2T}\sum_{t=1}^{T}\left(y_t - \widetilde{\mathbf{x}}^T \widetilde{\mathbf{A}}_t \widetilde{\mathbf{x}}\right)^2.$$

Then, the convergence proof of the DWF algorithm follows the convergence analysis given in [23]. In order to establish the convergence result, we need to check some mild assumptions.

We first check the Lipschitz continuity of the objective function $\widetilde{f}(\widetilde{\mathbf{x}})$.

Lemma 1. *Function $\widetilde{f}(\widetilde{\mathbf{x}})$ is Lipschitz continuous on $\widetilde{\mathcal{X}}$ with Lipschitz constant given by*

$$L \triangleq \frac{1}{2T}\sum_{t=1}^{T}\left(y_t \sigma_{\max}(\mathbf{A}_t) + 3\tau\sigma_{\max}^2(\mathbf{A}_t)\right),$$

where $\widetilde{\mathcal{X}} \triangleq \left\{\widetilde{\mathbf{x}} \mid \|\widetilde{\mathbf{x}}\|^2 \le \tau\right\}$.

Proof. According to the definition of the Lipschitz continuity, for $\forall \mathbf{u}, \mathbf{v} \in \widetilde{\mathcal{X}}$ we have

$$\begin{aligned}
&\left\|\nabla \widetilde{f}(\mathbf{u}) - \nabla \widetilde{f}(\mathbf{v})\right\| \\
\le& \frac{1}{2T}\left\|\sum_{t=1}^{T}\left((y_t - \mathbf{u}^T\widetilde{\mathbf{A}}_t\mathbf{u})\widetilde{\mathbf{A}}_t\mathbf{u} - (\mathbf{v}_t - \mathbf{v}^T\widetilde{\mathbf{A}}_t\mathbf{v})\widetilde{\mathbf{A}}_t\mathbf{v}\right)\right\| \\
\le& \frac{1}{2T}\sum_{t=1}^{T}\left(y_t\sigma_{\max}(\widetilde{\mathbf{A}}_t) + \|\mathbf{u}\|^2\sigma_{\max}^2(\widetilde{\mathbf{A}}_t)\right)\|\mathbf{u} - \mathbf{v}\| \\
&+ \frac{1}{2T}\sum_{t=1}^{T}\left\|\mathbf{u}^T\widetilde{\mathbf{A}}_t\mathbf{u}\widetilde{\mathbf{A}}_t\mathbf{v} - \mathbf{u}^T\widetilde{\mathbf{A}}_t\mathbf{v}\widetilde{\mathbf{A}}_t\mathbf{v}\right\| \\
&+ \frac{1}{2T}\sum_{t=1}^{T}\left\|\mathbf{u}^T\widetilde{\mathbf{A}}_t\mathbf{v}\widetilde{\mathbf{A}}_t\mathbf{v} - \mathbf{v}^T\widetilde{\mathbf{A}}_t\mathbf{v}\widetilde{\mathbf{A}}_t\mathbf{v}\right\| \\
\le& \frac{1}{2T}\sum_{t=1}^{T}\left(y_t\sigma_{\max}(\widetilde{\mathbf{A}}_t) + 3\tau\sigma_{\max}^2(\widetilde{\mathbf{A}}_t)\right)\|\mathbf{u} - \mathbf{v}\|.
\end{aligned}$$

□

It is also easy to check that the feasible set is convex and compact. Define $\widetilde{\mathbf{B}}_- = \mathbf{I}_2 \otimes \overline{\mathbf{B}}_- \in \mathbb{R}^{2M \times 2N}$ and $\widetilde{\mathbf{B}}_+ = \left|\widetilde{\mathbf{B}}_-\right|$. We can have $\widetilde{\mathbf{B}}_-^T\widetilde{\mathbf{B}}_- + \widetilde{\mathbf{B}}_+^T\widetilde{\mathbf{B}}_+ \succ \mathbf{I}$, such that the primal problem is strongly convex. Then by choosing

$$\rho\gamma \in (0,1), \quad c > \frac{1}{\tau} - 1 > 0, \quad \text{and} \quad \rho > (3+2c)L, \quad (16)$$

we are ready to show the convergence result of the DWF algorithm based on the ϵ-stationary solution defined below.

Definition 2. Given $\epsilon > 0$, the tuple $\left(\widetilde{\mathbf{x}}^\star, \widetilde{\boldsymbol{\omega}}^\star\right)$ is an ϵ-stationary solution if the following condition holds

$$\left\|\widetilde{\mathbf{B}}_-\widetilde{\mathbf{x}}^\star\right\|^2 \le \epsilon, \left\langle \nabla f(\mathbf{x}^\star) + \widetilde{\mathbf{B}}_-^T\widetilde{\boldsymbol{\omega}}^\star, \widetilde{\mathbf{x}}^\star - \widetilde{\mathbf{x}}\right\rangle \le 0, \ \forall \widetilde{\mathbf{x}} \in \widetilde{\mathcal{X}}, \quad (17)$$

where $\widetilde{\mathbf{x}}^\star \in \widetilde{\mathcal{X}}$ and $\widetilde{\boldsymbol{\omega}} \triangleq [\mathrm{Re}(\overline{\boldsymbol{\omega}}); \mathrm{Im}(\overline{\boldsymbol{\omega}})] \in \mathbb{R}^{2MK}$.

Note that any point satisfying condition (17) is also termed as an approximate KKT (AKKT) point in the literature.

Then based on the results in [23, Theorem 1 and Corollary 2], we can finally give the following convergence result.

Theorem 3. *For any given $\epsilon > 0$, the sequence $(\widetilde{\mathbf{x}}^r, \widetilde{\boldsymbol{\omega}}^r)$ or, equivalently $(\overline{\mathbf{x}}^r, \overline{\boldsymbol{\omega}}^r)$, generated by the DWF algorithm has the following properties:*

1) $\widetilde{\boldsymbol{\omega}}^{r+1} - \widetilde{\boldsymbol{\omega}}^r \to 0$, $\widetilde{\mathbf{x}}^{r+1} - \widetilde{\mathbf{x}}^r \to 0$ *and sequences* $\{\widetilde{\mathbf{x}}^r\}, \{\widetilde{\boldsymbol{\omega}}^r\}$ *are bounded.*
2) *Every limit point of the sequence $(\widetilde{\mathbf{x}}^r, \widetilde{\boldsymbol{\omega}}^r)$, denoted by $\left(\widetilde{\mathbf{x}}^\star, \widetilde{\boldsymbol{\omega}}^\star\right)$, converges to a $\gamma^2\|\widetilde{\boldsymbol{\omega}}^\star\|^2$-stationary solution globally of the problem (2) in a sublinear rate.*

V. Numerical Simulations

In this part, we present numerical experiments evaluating the performance of the proposed DWF algorithm using synthetic data. We consider a noiseless 1D Gaussian case (i.e., no additive noise in the sensing process, and $\mathbf{a}_t$ is generated from Gaussian distribution) with parameter dimension $K = 50$ and number of samples $T = 300$. A random graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is generated with $N = 10$ agents. We compare our proposed DWF algorithm with the centralized WF algorithm [5]. The two algorithms were initialized at the same starting point.

The convergence result of the two algorithms is shown in Fig. 2. In our proposed DWF algorithm, the stepsize is chosen according to the Lipschitz constant of the original problem, so the DWF algorithm can at most be as good as the WF algorithm. From Fig. 2, it can be observed that the DWF algorithm is slower than WF with respect to the number of iterations but also achieves a competitive rate. The performance of the WF and DWF algorithms are also evaluated in terms of the relative root mean-square error defined as $\mathrm{relative\ error} \triangleq \mathrm{dist}(\mathbf{z}, \mathbf{x}) / \|\mathbf{x}\|$. The relative error performance with respect to iterations is reported in Fig. 3. It can be seen that both the WF and DWF algorithms can finally retrieve very high quality solutions of the original problem. For the proposed DWF algorithm, we also show the result of the consensus error which is defined by $\mathrm{consensus\ error} \triangleq \sum_{i=1}^{N}\left\|\mathbf{x}_i - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\right\|^2$. The consensus error versus iterations is reported in Fig. 4, illustrating that. our proposed algorithm can finally reach the consensus.

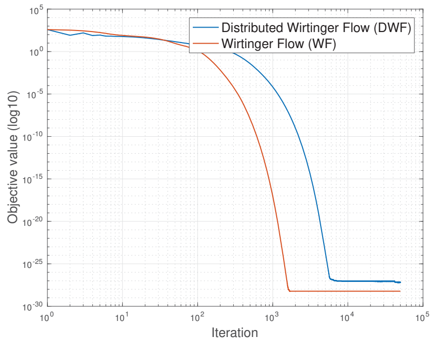

Fig. 2. Convergence result of the objective function value of WF and DWF.

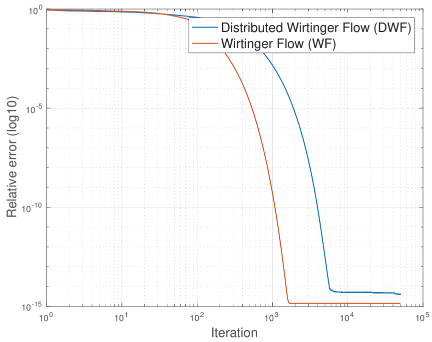

Fig. 3. The relative error of WF and DWF.

VI. Conclusions

In this paper, we have discussed the problem of solving a system of quadratic equations, a.k.a., the generalized phase retrieval problem over a distributed network. The problem has been formulated to minimize the squared loss of the measurements and the linear sensing intensity. A decentralized algorithm, named as distributed Wirtinger flow, is proposed, which has been proved to converge to an approximate KKT point of the original problem globally in a sublinear rate.

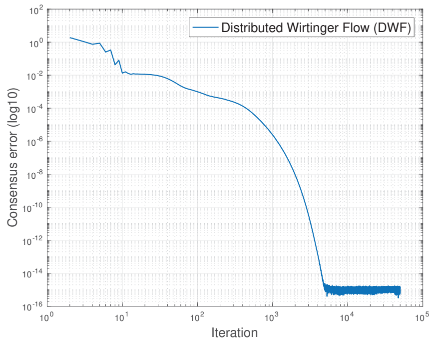

Fig. 4. The consensus error of DWF.

References

[1] R. Balan, P. Casazza, and D. Edidin, "On signal reconstruction without phase," *Applied and Computational Harmonic Analysis*, vol. 20, no. 3, pp. 345–356, 2006.

[2] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[3] J. R. Fienup, "Phase retrieval algorithms: A personal tour," *Applied optics*, vol. 52, no. 1, pp. 45–56, 2013.

[4] E. J. Candes, T. Strohmer, and V. Voroninski, "PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[5] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[6] Y. Chen and E. Candes, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 739–747.

[7] K. Wei, "Solving systems of phaseless equations via Kaczmarz methods: A proof of concept study," *Inverse Problems*, vol. 31, no. 12, p. 125008, 2015.

[8] T. Qiu, P. Babu, and D. P. Palomar, "PRIME: Phase retrieval via majorization-minimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5174–5186, 2016.

[9] R. Gerchberg and W. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane images," *Optik*, vol. 35, pp. 225–246, 1972.

[10] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Optics Letters*, vol. 3, no. 1, pp. 27–29, 1978.

[11] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 2796–2804.

[12] I. Waldspurger, A. dAspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," *Mathematical Programming*, vol. 149, no. 1-2, pp. 47–81, 2015.

[13] G. Wang, G. B. Giannakis, and Y. C. Eldar, "Solving systems of random quadratic equations via truncated amplitude flow," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 773–794, 2018.

[14] J. Sun, Q. Qu, and J. Wright, "A geometric analysis of phase retrieval," *Foundations of Computational Mathematics*, Aug 2017.

[15] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 461–497.

[16] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[17] N. Azizan-Ruhi, F. Lahouti, S. Avestimehr, and B. Hassibi, "Distributed solution of large-scale linear systems via accelerated projection-based consensus," in *Proc. the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[18] Y. Liang, M.-F. F. Balcan, V. Kanchanapally, and D. Woodruff, "Improved distributed principal component analysis," in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 3113–3121.

[19] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "A new class of distributed optimization algorithms: Application to regression of distributed data," *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.

[20] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2528–2541, 2014.

[21] P. Stoica, J. Li, and Y. Xie, "On probing signal design for MIMO radar," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4151–4161, 2007.

[22] Z. Zhao and D. P. Palomar, "MIMO transmit beampattern matching under waveform constraints," in *Proc. Speech and Signal Processing (ICASSP) 2018 IEEE Int. Conf. Acoustics*, Apr. 2018, pp. 3281–3285.

[23] D. Hajinezhad and M. Hong, "Perturbed proximal primal dual algorithm for nonconvex nonsmooth optimization," *submitted to Mathematical Programming*, 2018.

[24] W. Rudin, *Real and complex analysis*. McGraw-Hill Education, 1987.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.