Distributed Discrete-time Optimization with Coupling Constraints Based on Dual Proximal Gradient Method in Multi-agent Networks

Jianzheng Wang, Guoqiang Hu, Senior Member, IEEE

Abstract—In this paper, we aim to solve a distributed optimization problem with coupling constraints based on proximal gradient method in a multi-agent network, where the cost function of the agents is composed of smooth and possibly non-smooth parts. To solve this problem, we resort to the dual problem by deriving the Fenchel conjugate, resulting in a consensus based constrained optimization problem. Then, we propose a fully distributed dual proximal gradient algorithm, where the agents make decisions only with local parameters and the information of immediate neighbours. Moreover, provided that the non-smooth parts in the primal cost functions are with some simple structures, we only need to update dual variables by some simple operations and the overall computational complexity can be reduced. Analytical convergence rate of the proposed algorithm is derived and the efficacy is numerically verified by a social welfare optimization problem in the electricity market.

Index Terms—Multi-agent network; proximal gradient method; distributed optimization; Fenchel conjugate; dual problem.

I. INTRODUCTION

A. Background and Motivation

Decentralized optimization has become an active topic in recent years for solving various engineering problems, such as detection and localization in sensor networks [11], machine learning and regression problems [2], and economic dispatch in power systems [3], etc. As a typical optimization architecture, each agent maintains an individual cost function and the global optimal solution can be attained with multiple rounds of communications and decision-makings. In this paper, we focus on a class of composite optimization problems, where the cost functions are composed of smooth (differentiable) and possibly non-smooth (non-differentiable) parts, which are often discussed in various fields, such as resource allocation problems [4], Lasso regressions [5], and support vector machines [6], etc. To solve these problems, widely discussed techniques include alternating direction method of multipliers [7], primal-dual subgradient methods [8], and proximal gradient methods [9], etc.

Most existing works on decentralized optimizations assume that the agents are fully connected to ensure the correctness of the optimization results, which limits their usage in large-scale distributed networks [10][11]. To overcome this issue, a valid alternative is applying graph theory in modelling the communication links, leading to the distributed setup where the agents only communicate with their immediate neighbours [12]. However, with the increasing demand on the computational efficiency in various fields, more explorations on the algorithm development for distributed optimization problems (DOPs) are needed [13]. Noticing that proximal gradient methods are usually numerically more stable than the subgradient based counterparts in composite optimizations [14], in this work, we aim to develop an efficient distributed optimization algorithm based on proximal gradient method.

B. Literature Review

Fruitful distributed algorithms for solving DOPs can be found in the existing works. To adapt to large-scale distributed networks, consensus based DOPs without coupling constraint were studied in [15][19], where the agents make decisions with local variables and certain agreement on the optimal solution is achieved only through local communications.

Alternatively, we focus on optimizing a class of composite DOPs subject to coupling affine constraints via dual proximal gradient method. In this work, dual proximal gradient method corresponds to the proximal gradient method applied to the dual as also discussed in [4][20][21], where, however, no coupling constraint is considered. To the best knowledge of the authors, this work incorporates dual proximal gradient method in distributed setups with general coupling affine constraints for the first time, which enriches the existing algorithms for constrained DOPs.

To develop a fully distributed algorithm for the problem of interest, we propose a distributed dual proximal gradient (DDPG) algorithm. To highlight the new features and advantages of this work, the comparisons with some state-of-the-art works with similar problem setups are listed as follows.

- One distinct feature of the proposed DDPG algorithm is that, by resorting to the dual problem, we can only update the dual variables by some simple operations, e.g., basic proximal mappings and simple iterations, provided that the proximal mapping of the non-smooth parts in the primal cost functions can be explicitly derived, which is more efficient than the existing distributed algorithms with possibly costly computations of the primal variables.
or other auxiliary variables, e.g., provided in [22, 24, 31].

• In [22, 24, 31], some common fixed or varying global step-sizes are required. By contrast, the proposed fully distributed DDPG algorithm allows for heterogeneous step-sizes determined by local information, e.g., private objective functions and local parameters in the global constraints, which provides more flexibilities for the initialization process and is more adaptive to large-scale distributed networks.

• The consensus based distributed optimization algorithms studied in [23, 26, 27, 31] require the updating of some weighted running averages of variables or gradients, which increase the computational complexity and require more memory capacity for the auxiliary variables.

• An explicit convergence rate is derived for the proposed DDPG algorithm, which is not provided in [24, 26, 28, 30]. In addition, the algorithms in [22, 24, 28, 30, 31] assume some compact local constraints to ensure the convergence of the algorithms. By contrast, this work focuses on dual sequences without boundedness requirement on the primal variables.

The contributions of this work are summarized as follows.

• We consider a class of composite DOPs with local convex and coupling affine constraints. A DDPG algorithm is proposed by deriving the dual problem based on Fenchel conjugate, where the optimal solution can be attained when the agents execute updates only with the dual information of immediate neighbours and locally determined step-sizes, leading to a fully distributed computation environment.

• Different from the existing research works with similar problem setups, the proposed DDPG algorithm only requires the update of dual variables by some simple operations if the non-smooth parts of the objective functions are simple-structured, which can reduce the overall computational complexity. In addition, the proposed DDPG algorithm requires some widely used assumptions on the primal problems and explicit convergence rate is provided.

C. Paper Structure and Notations

The remainder of this paper is organized as follows. Section II provides some fundamental definitions and mathematical properties employed by this work. Section III formulates the optimization problem of interest and introduces the assumptions. In Section IV, the dual problem is formulated and the DDPG is proposed therein. The convergence analysis is conducted in Section V. The efficacy of the proposed DDPG algorithm is demonstrated in Section VI with a social welfare optimization problem in the electricity market. Section VII concludes this paper.

1For the DOPs with smooth cost functions, some existing works on dual algorithms, e.g., [32], can avoid the update of primal variables. However, directly extending their results to non-smooth cases can be costly in the sense that the computation of the gradient of the formulated dual function requires an additional nontrivial optimization process. Therefore, the contribution to computational efficiency of this work is established for possibly non-smooth cost functions.

2The proximal mapping can be equivalently written as prox_{\alpha \psi} as in some other works.

N and \( \mathbb{N} \) denote the non-negative and positive integer spaces, respectively. Let notation \( |A| \) be the size of set \( A \). \( \mathbb{R}_+^n \) denotes the \( n \)-dimensional Euclidean space only with non-negative real elements. Operator \( \cdot^\top \) represents the transpose of a matrix. \( A_1 \times A_2 \) denotes the Cartesian product of sets \( A_1 \) and \( A_2 \). \( \text{relint} A \) represents the relative interior of set \( A \). \( \| \cdot \| \) and \( \| \cdot \|_1 \) refer to the \( l_2 \) - and \( l_1 \)-norms, respectively. Define

\[
\| u \|_X^2 = u^\top X u \quad \text{with} \quad X \text{ a square matrix}. 
\]

\( \otimes \) is Kronecker product. \( I_n \) is an \( n \)-dimensional identity matrix and \( O_{n \times m} \) is an \((n \times m)\)-dimensional zero matrix. \( 1_n \) and \( 0_n \) denote the \( n \)-dimensional column vectors with all elements 1 and 0, respectively. Define

\[
D_A^n[u_n] = \begin{bmatrix} u_1 I_m & O & \cdots & O \\ O & u_{|A|} I_m \end{bmatrix} \in \mathbb{R}^{m|A| \times m|A|}. \tag{1}
\]

II. Preliminaries

Some frequently used definitions and relevant properties of graph theory, proximal mapping, and Fenchel conjugate are provided in this section.

A. Graph Theory

Define an undirected graph \( G = \{V, E\} \) for a multi-agent network, where \( V = \{1, 2, \ldots, N\} \) is the set of vertices and \( E \subseteq \{(i, j) | i, j \in V \text{ and } i \neq j\} \) is the set of edges with \((i, j) \in E \) unordered. \( G \) is connected if any two distinct vertices are linked by at least one path. \( V_i = \{j | (i, j) \in E\} \) is the neighbour set of agent \( i \). Let \( L \in \mathbb{R}^{N \times N} \) be the Laplacian matrix of \( G \). Then, the \((i, j)\)th element of \( L \), defined by \( d_{ij} \), follows \( d_{ij} = -1 \) if \((i, j) \in E \), \( d_{ij} = 0 \) if \((i, j) \notin E \) \& \( i \neq j \), and \( d_{ii} = |V_i| \) [33].

B. Proximal Mapping

A proximal mapping of a proper, convex, and closed function \( \psi : \mathbb{R}^n \to (-\infty, +\infty] \) is defined by

\[
\text{prox}_{\alpha \psi}[v] = \arg\min_u (\psi(u) + \frac{1}{2\alpha} \| u - v \|^2), \quad \alpha > 0, \quad v \in \mathbb{R}^n. \tag{2}
\]

A generalized version of proximal mapping can be defined as

\[
\text{prox}_{\sqrt{\alpha} \psi}[v] = \arg\min_u (\psi(u) + \frac{1}{2} \| u - v \|^2_{X^{-1}}), \tag{3}
\]

with \( X \in \mathbb{R}^{n \times n} \) a positive definite matrix [20].

C. Fenchel Conjugate

\( \psi : \mathbb{R}^n \to (-\infty, +\infty] \) is a proper function. Then, the Fenchel conjugate of \( \psi \) is defined by

\[
\psi^\circ(v) = \sup_u \{ v^\top u - \psi(u) \}, \text{ which is convex [34 Sec. 3.3].}
\]

Lemma 1. (Extended Moreau Decomposition [35 Thm. 6.45])

\( \psi : \mathbb{R}^n \to (-\infty, +\infty] \) is a proper, convex, and closed function. \( \psi^\circ \) is its Fenchel conjugate. Then,

\[
v = \alpha \text{prox}_{\frac{1}{\alpha} \psi} \left[ \frac{v}{\alpha} \right] + \text{prox}_{\psi^\circ} [v], \tag{3}
\]

\( v \in \mathbb{R}^n, \alpha > 0, \)
Lemma 2. \textcolor{red}{[29]} Lemma V.7] Let $\psi : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ be a proper, closed, $\sigma$-strongly convex function and $\psi^\circ$ be its Fenchel conjugate, $\sigma > 0$. Then,\[
abla^2 \psi^\circ(v) \text{ is } \frac{1}{\sigma} \text{-Lipschitz continuous.}
\]

III. PROBLEM FORMULATION

The problem formulation and relevant assumptions are provided as follows.

Let $F(x) = \sum_{i \in V} F_i(x_i)$ be the global cost function of a multi-agent network $G = \{V, E\}$, $x_i \in \mathbb{R}^M$, $x = [x_1^\top, \ldots, x_N^\top] \in \mathbb{R}^{NM}$. Agent $i$ maintains the private cost function $F_i(x_i) = f_i(x_i) + g_i(x_i)$. Let $X_i \subseteq \mathbb{R}^M$ be the feasible region of $x_i$. Then, the feasible region of $x$ can be defined by $X = X_1 \times X_2 \times \ldots \times X_N \subseteq \mathbb{R}^{NM}$. Then, an affine-constrained optimization problem of $G$ can be given by

\[
\begin{align*}
(P1) \quad & \min_{x \in X} \sum_{i \in V} F_i(x_i) \\
& \text{subject to } Ax = b, \\
& A \in \mathbb{R}^{B \times NM}, \ b \in \mathbb{R}^B, \ \text{which is equivalent to} \\
(P2) \quad & \min_x \sum_{i \in V} (F_i(x_i) + \mathbb{I}_{X_i}(x_i)) \\
& \text{subject to } Ax = b, \\
& \mathbb{I}_{X_i}(x_i) = \begin{cases} 0, & \text{if } x_i \in X_i, \\ +\infty, & \text{otherwise.} \end{cases}
\end{align*}
\]

Remark 1. Note that for an inequality constraint $Ax \leq b$, one can formulate an equality constraint $Ax + y = b$ with $y \in \mathbb{R}_+^B$ being a slack variable. Then, the inequality-constrained problem can be equivalently written as

\[
(P1+) \quad \min_{x \in X, y \in \mathbb{R}_+^B} \sum_{i \in V} F_i(x_i) \\
& \text{subject to } Ax + y = b.
\]

To realize decentralized computations, $y$ can be decomposed and assigned to the agents. Hence, the structure of Problem $(P1+)$ complies with that of Problem $(P1)$.

Assumption 1. $G$ is connected and undirected.

Assumption 2. $f_i : \mathbb{R}^M \rightarrow (-\infty, +\infty]$ and $g_i : \mathbb{R}^M \rightarrow (-\infty, +\infty]$ are both proper, convex, and closed extended real-valued functions. In addition, $f_i$ is differentiable and $\sigma_i$-strongly convex, $\sigma_i > 0$, $i \in V$.

Similar assumptions in Assumption 2 can be referred to in \textcolor{red}{[4, 20, 37, 41]}.\]

Assumption 3. $X_i$ is non-empty, convex and closed, $i \in V$; there exists an $\bar{x} \in \text{relint}X$ such that $A\bar{x} = b$.

Remark 2. By Assumption 3, $\mathbb{I}_{X_i}$ is proper, convex, and closed \textcolor{red}{[36]}, which complies with the assumption on $g_i$. Therefore, it is also feasible to omit the discussion on $\mathbb{I}_{X_i}$ in Problem (P2) as in \textcolor{red}{[4, 20, 27]}. In this work, we highlight the existence of $X_i$ for more detailed discussions.

IV. DISTRIBUTED DUAL PROXIMAL GRADIENT ALGORITHM DEVELOPMENT

In this section, we propose a fully distributed DDPG algorithm for solving the problem of interest.

A. Dual Problem

The dual problem of Problem (P2) is formulated in this subsection. By decoupling the objective function of Problem (P2), we have

\[
\begin{align*}
(P3) \quad & \min_{x, z} \sum_{i \in V} (f_i(x_i) + (g_i + \mathbb{I}_{X_i})(z_i)) \\
& \text{subject to } Ax = b, \ x_i = z_i, \ \forall i \in V,
\end{align*}
\]

where $z = [z_1^\top, \ldots, z_N^\top] \in \mathbb{R}^{NM}$ with $z_i \in \mathbb{R}^M$ a slack vector. The Lagrangian function of Problem (P3) can be given by

\[
L(x, z, \theta, \mu) = \sum_{i \in V} (f_i(x_i) + (g_i + \mathbb{I}_{X_i})(z_i)) \\
+ \mu_i^\top (x_i - z_i) + \theta^\top (Ax - b) \\
= \sum_{i \in V} (f_i(x_i) + x_i^\top (A_i^\top \theta + \mu_i)) \\
+ (g_i + \mathbb{I}_{X_i})(z_i) - z_i^\top \mu_i - b^\top \theta, \quad (5)
\]

where $\mu_i \in \mathbb{R}^M$ and $\theta \in \mathbb{R}^B$ are the Lagrangian multiplier vectors associated with constraints $x_i = z_i$ and $Ax = b$, respectively. $\mu = [\mu_1^\top, \ldots, \mu_N^\top]^\top \in \mathbb{R}^{NM}$. $A_i \in \mathbb{R}^{B \times M}$ is the $i$th column sub-block of $A$ with $A = [A_1, \ldots, A_N]$. Then, the dual function can be obtained by minimizing $L(x, z, \theta, \mu)$ with $(x, z)$, which is

\[
D(\theta, \mu) = \min_{x, z} \sum_{i \in V} (f_i(x_i) + x_i^\top (A_i^\top \theta + \mu_i)) \\
+ (g_i + \mathbb{I}_{X_i})(z_i) - z_i^\top \mu_i - b^\top \theta \\
= \min_{x, z} \sum_{i \in V} (f_i(x_i) - x_i^\top H_i \lambda_i) \\
+ (g_i + \mathbb{I}_{X_i})(z_i) - z_i^\top F \lambda_i - \kappa_i E \lambda_i) \\
= \sum_{i \in V} (-f_i^\top (H_i \lambda_i) - \kappa_i E \lambda_i) \\
- (g_i + \mathbb{I}_{X_i})^\circ (F \lambda_i)), \quad (6)
\]

where $H_i = -[A_i^\top, -\mathbb{I}_M] \in \mathbb{R}^{M \times (M + B)}, \lambda_i = [\theta^\top, \mu_i^\top]^\top \in \mathbb{R}^{M + B}, \ F = [O_M \times B, \mathbb{I}_M] \in \mathbb{R}^{M \times (M + B)}, \ E = [b^\top, 0_{M}] \in \mathbb{R}^{1 \times (M + B)}, \ \sum_{i \in V} \kappa_i = 1$. The forth equality in (6) employs the definition of Fenchel conjugate and $(g_i + \mathbb{I}_{X_i})^\circ$ denotes the Fenchel conjugate of $g_i + \mathbb{I}_{X_i}$. Hence, the dual problem of Problem (P3) can be formulated as

\[
(P4) \quad \min_{\lambda} \quad P(\lambda) + R(\lambda),
\]

where $\lambda = [\lambda_1^\top, \ldots, \lambda_N^\top]^\top \in \mathbb{R}^{NB + NM}, \ P(\lambda) = \sum_{i \in V} (f_i^\top (H_i \lambda_i) + \kappa_i E \lambda_i)$ and $R(\lambda) = \sum_{i \in V} (g_i + \mathbb{I}_{X_i})^\circ (F \lambda_i)$.\]
B. Distributed Dual Proximal Gradient Algorithm

In this subsection, we aim to solve Problem (P4) in a distributed manner based on proximal gradient method.

In Problem (P4), the variables of $f^i_*(H,\lambda_i)$ are coupled in terms of the common component $\theta$ in $\lambda_i$, but those of $(g_l + \mathbb{I}_X)_*(F \lambda_i)$ are decoupled since $F \lambda_i = \mu_i$. In the following, with a slight abuse of notation, we redefine $\lambda_i = [\theta_i^\top, \mu_i^\top]^\top \in \mathbb{R}^{M^*_i}$, where $\theta_i$ is the local estimate of the common $\theta$. Then, Problem (P4) can be equivalently rewritten as

$$
\begin{align*}
\text{(P5)} & \quad \min_{\lambda} \ P(\lambda) + R(\lambda) \\
& \text{subject to} \quad K \lambda_j = K \lambda_l, \quad \forall (j, l) \in E, \quad (7)
\end{align*}
$$

where $K = [I_B, O_{B \times M}]$. Constraint (7) ensures the partial consistency among $\lambda_i$ in terms of component $\theta_i$, i.e., $\theta_i = K \lambda_i$. Let $\lambda^* = [(\lambda^*_1)^\top, ..., (\lambda^*_M)^\top]^\top$ be the optimal solution to Problem (P5) with $\lambda^*_i = [\theta^*_i]^\top, [\mu^*_i]^\top]^\top$.

In the following, we assume that the range of $\theta^*_i$ is estimable, i.e., $\theta^*_i \in S_i$, with $S_i \subset \mathbb{R}^D$ being the estimated non-empty, convex and compact zone. For the convenience of the following discussion, we define $\Gamma = \max_{\lambda \in C} \sup_{\epsilon \in S_i} \| \lambda \|_1$. Note that considering constraint $\theta_i \in S_i$ is equivalent to accommodating an indicator function $I_{S_i}(\theta_i)$ into the non-smooth part [36]. Then, Problem (P5) can be modified into

$$
\begin{align*}
\text{(P6)} & \quad \min_{\lambda} \ \Phi(\lambda) \\
& \text{subject to} \quad K \lambda_j = K \lambda_l, \quad \forall (j, l) \in E, \quad (8)
\end{align*}
$$

where $\Phi(\lambda) = P(\lambda) + Q(\lambda), P(\lambda) = \sum_{i \in V} p_i(\lambda_i), Q(\lambda) = \sum_{i \in V} g_i(\lambda_i) + \kappa_i E \lambda_i; q_i(\lambda_i) = (g_i + \mathbb{I}_X)_*(\lambda_i) + \mu_i(\theta_i) = (g_i + \mathbb{I}_X)_*(F \lambda_i)$ + $I_{S_i}(K \lambda_i)$. Note that (8) can be represented by a compact equation $ML = 0$, where $M = L \otimes K \in \mathbb{R}^{BN \times (NB + NM)}$. It can be checked that $ML = L \theta$, where $L = L \otimes I_B \in \mathbb{R}^{NB \times NB}$ is an augmented Laplacian matrix of $G$ and $\theta = [\theta_1^\top, ..., \theta_N^\top]^\top \in \mathbb{R}^{NB}$.

Then, the Lagrangian function of Problem (P6) can be given by

$$
L(\lambda, \xi) = P(\lambda) + Q(\lambda) + \xi^\top M \lambda, \quad (9)
$$

where $\xi = [\xi_1^\top, ..., \xi_M^\top]^\top \in \mathbb{R}^{NB}$ is the collection of Lagrangian multipliers. Let $C$ be the set of the saddle points of $L(\lambda, \xi)$. Then, any saddle point $(\lambda^*, \xi^*) \in C$ satisfies [42]

$$
\begin{align*}
\lambda^* = \lambda^*, \quad \xi^* = \xi^*, \quad (10)
\end{align*}
$$

All $\lambda \in \mathbb{R}^{NB + NM}, \xi \in \mathbb{R}^{NB}$. We aim to seek a saddle point $(\lambda^*, \xi^*)$, which can be characterized by Karush-Kuhn-Tucker (KKT) conditions [43]

$$
\begin{align*}
0 & \in \nabla_{\lambda} P(\lambda^*) + \partial_{\lambda} Q(\lambda^*) + M^\top \xi^*, \quad (11) \\
M^* \lambda^* & = 0. \quad (12)
\end{align*}
$$

Based on the previous discussion, the DDPG algorithm for solving Problem (P6) is designed as

$$
\begin{align*}
\lambda(t + 1) & = \text{pro}x_{Q}^{M^*_i + B}[\lambda(t) - D_{\lambda}^{M + B}[\xi(t)] \\
& \cdot (\nabla_{\lambda} P(\lambda(t)) + M^\top \xi(t))], \quad (13) \\
\xi(t + 1) & = \xi(t) + D_{\lambda}^{M^*_i}[\gamma_i] M \lambda(t + 1), \quad (14)
\end{align*}
$$

which means

$$
\begin{align*}
\lambda_i(t + 1) & = \text{pro}x_{Q}^{M^*_i}[\lambda_i(t) - c_i(\nabla_{\lambda_i} p_i(\lambda_i(t)) + M_i^\top \xi_i(t))] \\
& = \text{pro}x_{Q}^{M^*_i}[\lambda_i(t) - c_i(\nabla_{\lambda_i} p_i(\lambda_i(t)) \\
& + \sum_{j \in V_j} K_i^\top (\xi_i(t) - \xi_j(t))], \quad (15)
\end{align*}
$$

due to the separability of $P$ and $Q$, $\forall i \in V$, $t \in \mathbb{N}$. $M_i \in \mathbb{R}^{NB \times (B + M)}$ and $M_i^* \in \mathbb{R}^{B \times (NB + NM)}$ are the $i$th column and $i$th row sub-blocks of $M$, respectively, i.e., $M = [M_1, ..., M_i, ..., M_N] = [(M_1^*)^\top, ..., (M_i^*)^\top, ..., (M_N^*)^\top]^\top$. $c_i, \gamma_i > 0$ are step-sizes.

**Remark 3.** The estimated $S_i$ enables the range of $\theta_i$ to be bounded, which, as we will see later, facilitates the convergence analysis of DDPG algorithm. Similar settlement can be referred to in [24]. In practice, the estimation of $S_i$ relies on the experience in the specific problems. For example, in some social welfare optimization problems in the electricity market, the optimal dual variables can be the settled energy prices [44], whose range can be easy to estimate with historical prices.

**Remark 4.** From [15] and [16], it can be seen that each agent only needs the information of its neighbours and updates with locally determined step-size ($K$ contains the dimension information of primal and dual variables without other shared global information), which results in a fully distributed computation fashion of the DDPG algorithm.

The detailed computation procedure of DDPG algorithm is stated in Algorithm 1.

**Algorithm 1 Distributed Dual Proximal Gradient Algorithm**

1: Initialize $\lambda(0), \xi(0)$. Determine step-sizes $c_i, \gamma_i > 0$, $\forall i \in V$.
2: for $t = 0, 1, 2, \ldots$ do
3:   for $i = 1, 2, \ldots, N$ do (in parallel)
4:     Update $\lambda_i$ by (15).
5:   for $i$ do (in parallel)
6:     Update $\xi_i$ by (16).
7:   end for
8: end for

C. Computational Complexity of DDPG Algorithm

To apply (15), one needs to compute (i) $\nabla p_i$, and (ii) the proximal mapping of $q_i$, $i \in V$. For (i), $\nabla p_i$ can be efficiently obtained given that $f_i$ is simple-structured and, consequently, $\nabla f_i^p$ can be analytically derived, e.g., $f_i$ is a quadratic function [36, Sec. 3.3.1]. For (ii), some feasible methods for different cases are introduced as follows.
1) Case 1: If the proximal mapping of \( g_i + I_{X_i} \) can be easily obtained by \( q_i \), we have \( \text{prox}^{g_i}_{q_i} = \text{prox}^{g_i}_{\mu} \times \text{prox}^{(g_i+I_{X_i})o}_{\mu} \) [35, Thm. 6.6], where \( \text{prox}^{g_i}_{q_i} \) is a Euclidean projection onto \( S_i \) [9, Sec. 1.2] and \( \text{prox}^{(g_i+I_{X_i})o}_{\mu} \) can be obtained by calculating \( \frac{1}{c_i} \text{prox}^{g_i}_{q_i} \) with Lemma [1]. Then, [15] can be modified into

\[
\begin{align*}
\theta_i(t) &= \theta_i(t) - c_i(\nabla_{\theta_i} p_i(\lambda_i(t)) + \sum_{j \in V_i} \langle \xi_j(t) - \xi_j(t) \rangle), \\
\varrho_i(t+1) &= \text{prox}^{\varrho_i}_{\mu_i}(\varrho_i(t)) = \Pi_{S_i}[\varrho_i(t)], \\
\mu_i(t+1) &= \text{prox}^{\varrho_i}_{\mu_i}(\mu_i(t)) = \mu_i(t) - c_i \text{prox}^{\varrho_i}_{\mu_i}(\varrho_i(t)),
\end{align*}
\]

where \( q_i(t) = (g_i + I_{X_i})^o \), \( q_i^o(t) = (g_i + I_{X_i})^{oo} = g_i + I_{X_i} \), due to the convexity and lower semi-continuity of \( g_i + I_{X_i} \), and \( (g_i + I_{X_i})^{oo} \) is the biconjugate of \( g_i + I_{X_i} \) [36, Sec. 3.3.2], \( S_i[t] \) is an Euclidean projection onto \( S_i \). Essentially, [17] - [20] are obtained by decomposing \( \lambda_i(t+1) \) and using the above mentioned properties. With the above arrangement, the calculation of the proximal mapping of \( (g_i + I_{X_i})^o \) can be avoided as shown in [20], leading to the reduction of the computational complexity if the proximal mapping of \( g_i + I_{X_i} \) is easier to obtain. For instance, in a Lasso problem with penalty \( g_i(x_i) = \|x_i\|_1 \), and \( X_i = \mathbb{R}^M \), the proximal mapping of \( l_1 \)-norm is a soft thresholding operator with analytical solution [35, Sec. 6.3].

2) Case 2: Take the advantage of the structure of \( g_i \) in some specific problems. For example, consider a regularization problem, where the penalty is an Euclidean e-norm: \( g_i(x_i) = \|x_i\|_e \), \( X_i = \mathbb{R}^M \). Then, we can have

\[
\begin{align*}
q_i(\lambda_i) &= g_i^o(\mu_i) + I_{S_i}[\theta_i], \\
&= I_{Y_i}[\mu_i] + I_{S_i}[\theta_i], \\
&= \begin{cases} 
0, & \text{if } \mu_i \in \mathcal{V}_i & \& \theta_i \in S_i, \\
\infty, & \text{otherwise},
\end{cases} \\
&= I_{Y_i}[\lambda_i],
\end{align*}
\]

where \( \mathcal{V}_i = \{ v \in \mathbb{R}^M \mid \|v\|_e \leq 1 \} \) (convex zone) with \( \cdot \), \( \cdot \) - the dual norm of \( \cdot \), \( Y_i = S_i \times \mathcal{V}_i \) (convex zone). The second equality holds by computing the conjugate of a norm [36, Sec. 3.3.1]. Then, the proximal mapping of \( q_i \) is an Euclidean projection onto \( Y_i \) [9, Sec. 1.2].

3) Case 3: If \( g_i \) is with certain complicated structure, as a general method, we can construct a strongly convex nonsmooth \( g_i \) (e.g., shift a strongly convex component of the smooth part to \( g_i \)). Then, rewrite [15] by the definition of proximal mapping, which gives

\[
\begin{align*}
\lambda_i(t+1) &= \arg\min_{\lambda_i} (q_i(\lambda_i) + \frac{1}{2c_i} \|\lambda_i - \lambda_i(t)\|_1^2) \\
&= c_i(\nabla_{\lambda_i} p_i(\lambda_i(t)) + \sum_{j \in V_i} \xi_j(t) - \xi_j(t)),
\end{align*}
\]

To solve [22], one can utilize a subgradient descent method by computing

\[
\nabla_{\lambda_i} p_i(\lambda_i) = \nabla_{\lambda_i} (g_i + I_{X_i})^o(\mathcal{F} \lambda_i) + \nabla_{\lambda_i} I_{S_i}(\mathcal{K} \lambda_i).
\]

Remark 5. (Extension of assumption on \( f_i \)) In the case that the structure of \( f_i \) is complicated (can be non-smooth but still strongly convex), [15] can also be implemented by computing

\[
\begin{align*}
\nabla_{\lambda_i} p_i(\lambda_i) &= \nabla_{\lambda_i} f_i^o(H_i \lambda_i) + \kappa_i E^T \\
&= H_i^T \nabla_{H_i \lambda_i} f_i^o(H_i \lambda_i) + \kappa_i E^T,
\end{align*}
\]

where \( \nabla_{H_i \lambda_i} f_i^o(H_i \lambda_i) = \arg\max_u (\langle H_i \lambda_i \rangle^T u - (g_i + I_{X_i})^o(u)) \) by Lemma [2]. However, similar to Case 3, [24] requires a higher computational complexity since an inner-loop optimization process to compute the subgradient of \( g_i \), which can be completed only with local information.

V. CONVERGENCE ANALYSIS

The convergence analysis of the proposed DDPG algorithm is conducted in this section.

Lemma 3. With Assumption [2] that the Lipschitz constant of \( \nabla_{\lambda} P(\lambda) \) is given by \( h = \sqrt{\sum_{i \in V} h_i^2} \), where \( h_i = \frac{\|H_i\|}{\sigma_i} \).

See the proof in Appendix A.

Lemma 4. Suppose that Assumptions [12] hold. Based on Algorithm [7] for any \( (\lambda^*, \xi^*) \in C \) and \( t \in \mathbb{N} \), we have

\[
\Phi(\lambda(t+1)) - \Phi(\lambda^*) - (\xi^*)^T M(\lambda(t+1)) \leq 0, \forall t, \Psi_t,
\]

where

\[
\begin{align*}
\Psi_t &= \|\lambda^* - \lambda(t)\|_D^2 + \|\lambda^* - \lambda(t)\|_D^2 + \|\xi^* - \xi(t)\|_D^2 - \|\xi^* - \xi(t)\|_D^2 \\
&- \|\lambda^* - \lambda(t+1)\|_D^2 - \|\xi^* - \xi(t+1)\|_D^2 \\
&- \|\xi^* - \xi(t+1)\|_D^2.
\end{align*}
\]
where $\omega$ and $\phi$ are parameters, whose values are set in Table I.

In this section, we demonstrate of the performance of the DDPG algorithm by solving a social welfare optimization problem in an electricity market.

### A. Simulation Setup

Define $V_{i, UC}$ and $V_{i, user}$ as the sets of utility companies (UCs) and users, respectively. Define $x = [x_{i, UC}^{user}, x_{i, user}^{user}]^T$, where $x_{i, UC}$ is the energy generation quantity of UC $i$ and $x_{i, user}$ is the demand of user $j$. $\phi_i(x_{i, UC})$ and $\omega_j(x_{j, user})$ are the cost function of UC $i$ and utility function of user $j$, respectively, $i \in V_{i, UC}$, $j \in V_{i, user}$. Then, the social welfare optimization problem of the market can be formulated as

\begin{equation}
\begin{aligned}
& \text{min} \quad \sum_{i \in V_{i, UC}} \phi_i(x_{i, UC}) - \sum_{j \in V_{i, user}} \omega_j(x_{j, user}) \\
& \text{subject to} \quad \sum_{i \in V_{i, UC}} x_{i, UC} = \sum_{j \in V_{i, user}} x_{j, user}, \\
& \quad x_{i, UC} \in [0, \frac{50}{\gamma_{max}}], \quad \forall i \in V_{i, UC}, \\
& \quad x_{j, user} \in [0, \frac{50}{\gamma_{max}}], \quad \forall j \in V_{i, user}, \\
\end{aligned}
\end{equation}

where

\begin{equation}
\begin{aligned}
\phi_i(x_{i, UC}) &= \delta_i(x_{i, UC})^2 + \theta_i x_{i, UC} + \beta_i, \\
\omega_j(x_{j, user}) &= \left\{ \begin{array}{ll}
\tau_j x_{j, user} - \pi_j (x_{j, user})^2, & x_{j, user} \leq \frac{\pi_j}{\tau_j}, \\
\tau_j x_{j, user}^2, & x_{j, user} > \frac{\pi_j}{\tau_j},
\end{array} \right.
\end{aligned}
\end{equation}

with $\delta_i, \theta_i, \beta_i, \tau_j, \pi_j$ being parameters, whose values are set in Table I. $x_{i, UC}$ is the supply-demand balance constraint. $x_{i, UC}^T = [0, x_{i, UC}^{user}]$ and $x_{i, user}^T = [0, x_{i, user}^{user}]$ with $x_{i, UC}^{user}, x_{i, user}^{user} > 0$. Define $A = [1, -1]$. Then, (28) is equivalent to $Ax = 0$. 

### VI. NUMERICAL RESULT

In this section, we demonstrate of the performance of the DDPG algorithm by solving a social welfare optimization problem in an electricity market.

#### B. Simulation Result

To demonstrate the performance of Algorithm 1, we consider the communication typology shown in Fig. 1. The simulation result is shown in Figs. 2 to 4. Fig. 2 shows the dynamics of dual variables $\theta$ and $\mu$. It can be seen that all the elements in $\theta$ converge to $\theta^* = -8.1$ while $\mu$ converges to $\mu^* = [-0.61, 2.34, 0, 0, 0]^T$. One can check that the optimal solution at the saddle point of $L$ is $x^* = \arg \min_{x} L(x, z, \theta^*, \mu^*) = [0, 150, 0, 50, 0, 50]^T$ which means the lower bound and upper bound of $x_{1, UC}$ and $x_{2, UC}$ are activated, respectively, while other variables reach interior optimal solutions. Fig. 3 depicts the dynamics of $\xi$. Fig. 4 shows the simulation result is shown in Figs. 2 to 4.
shows that the value of dual function $\Phi(\lambda)$ (as defined in Problem (P6)) is decreased to around 756.53.

VII. Conclusion

In this work, we considered solving a composite DOP with both local convex and coupling affine constraints. A fully distributed DDPG algorithm was proposed for solving the this problem by resorting to the dual problem. As a distinct feature compared with the existing research works with similar problem setups, we showed that if the non-smooth parts of the objective functions are with some simple structures, one only needs to update dual variables by some simple operations, leading to the reduction of overall computational complexity.

APPENDIX

A. Proof of Lemma 3

By Lemma 2, $\nabla f_i^\phi$ is $\sigma_i$-Lipschitz continuous, which means
$$\| \nabla_v f_i^\phi(H_v, v) - \nabla_u f_i^\phi(H_u, u) \|
= \| H_v^T \nabla_{H_v} f_i^\phi(H_v, v) - H_u^T \nabla_{H_u} f_i^\phi(H_u, u) \|
\leq \| H_v^T \| \| \nabla_{H_v} f_i^\phi(H_v, v) - \nabla_{H_u} f_i^\phi(H_u, u) \|
\leq \| H_v^T \| \| H_v - H_u \|
\leq \frac{\| H_v^T \|^2}{\sigma_i} \| v - u \| = h_i \| v - u \|, \quad (34)$$

$\forall v, u \in \mathbb{R}^{M+B}$, which means $\nabla_{\lambda_i} f_i^\phi(H_i, \lambda_i)$ is $h_i$-Lipschitz continuous and, therefore, $\nabla_{\lambda_i} \lambda_i(\lambda_i) = \nabla_{\lambda_i} f_i^\phi(H_i, \lambda_i) + \kappa_i E^\top$ is also $h_i$-Lipschitz continuous.

On the other hand, due to the separability of $P(\lambda)$, $\nabla_{\lambda} P(\lambda)$ can be decoupled with respect to each $\lambda_i$, i.e.,
$$\nabla_{\lambda} P(\lambda) = \begin{bmatrix} \nabla_{\lambda_1} p_1(\lambda_1) \\ \vdots \\ \nabla_{\lambda_N} p_N(\lambda_N) \end{bmatrix}. \quad (35)$$

By using the Euclidean $l_2$-norm, the Lipschitz constant of $\nabla_{\lambda} P(\lambda)$ can be obtained as $h$.

B. Proof of Lemma 4

By the first-order optimality condition of (13) in terms of (2), we have
$$0 \in \partial_\lambda Q(\lambda(t+1)) + D_{\lambda}^{M+B} \frac{1}{\gamma_1} (\lambda(t+1) - \lambda(t)), \quad -\nabla_{\lambda} Q(\lambda(t+1)) - M^\top \xi(t)$$
$$= \partial_\lambda Q(\lambda(t+1)) - D_{\lambda}^{M+B} \frac{1}{\gamma_1} (\lambda(t) - \lambda(t+1))$$
$$+ \nabla_{\lambda} P(\lambda(t)) + M^\top \xi(t+1)$$
$$- M^\top \frac{1}{\gamma_1} \lambda(t+1), \quad (36)$$

where $D_{\lambda}^{M+B} \frac{1}{\gamma_1} = (D_{\lambda}^{M+B} [\gamma_1])^{-1}$. From the convexity of $Q(\lambda)$, we have
$$Q(\lambda) - Q(\lambda(t+1)) \geq (\lambda - \lambda(t+1))^\top D_{\lambda}^{M+B} \frac{1}{\gamma_1} (\lambda(t) - \lambda(t+1))$$

Fig. 2. Dynamics of $\hat{\theta}$ and $\mu$.

Fig. 3. Dynamics of $\xi$.

Fig. 4. Dynamics of $\Phi(\lambda)$. 

From the convexity and $h_i$-Lipschitz continuous differentiability of $p_i$, we have

$$
\begin{align*}
(\lambda - \lambda(t+1))^T \nabla_\lambda P(\lambda(t)) \\
- (\lambda - \lambda(t+1))^T M^T \xi(t+1) \\
+ (\lambda - \lambda(t+1))^T M^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1).
\end{align*}
$$

(37)

By (37), we have

$$0 = D_l^{[\gamma_l]}[\lambda] \xi(t) - (\xi(t) - \xi(t+1))^T + MA(t+1),
$$

(39)

where $D_l^{[\gamma_l]}[\lambda] = (D_l^{[\gamma_l]})^{-1}$. Therefore, by multiplying the both sides of (39) by $(\xi - \xi(t+1))^T$, we have

$$
(\xi - \xi(t+1))^T D^{[\gamma_l]}(\lambda(t) - \lambda(t+1)) \\
+ (\xi - \xi(t+1))^T M A(t+1) = 0.
$$

(40)

By adding (37) and (38) together from the both sides, we have

$$
\begin{align*}
(\lambda - \lambda(t+1))^T \nabla_\lambda P(\lambda(t)) \\
- (\lambda - \lambda(t+1))^T M^T \xi(t+1) \\
+ (\lambda - \lambda(t+1))^T M^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1)
\end{align*}
$$

$$\leq
\begin{align*}
(\lambda - \lambda(t+1))^T \nabla_\lambda P(\lambda) \\
- (\lambda - \lambda(t+1))^T M^T \xi(t+1) \\
+ (\lambda - \lambda(t+1))^T M^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1)
\end{align*}
$$

$$\leq
\begin{align*}
(\lambda - \lambda(t+1))^T \nabla_\lambda P(\lambda) \\
- (\lambda - \lambda(t+1))^T M^T \xi(t+1) \\
+ (\lambda - \lambda(t+1))^T M^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1)
\end{align*}
$$

$$= (\lambda - \lambda(t))^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1) - (\lambda - \lambda(t))^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1) \\
+ (\lambda - \lambda(t))^T M^T D_{\gamma_l}^{[\gamma_l]}[\lambda]M \lambda(t+1)
$$

$$+ \|\xi - \xi(t+1)\|^2_{D_l^{[\gamma_l]}[\xi(t+1) + MA(t+1)]}.
$$

(41)

Based on (42) and (43), the proof is completed.

C. Proof of Theorem 7

Note that (41) holds for all $\lambda \in R^{N_{B+NM}}$ and $\xi \in R^{NB}$. The proof is conducted by discussing the following two scenarios.

1) Scenario 1: If $M \lambda(T+1) \neq 0$, by letting $\lambda = \lambda^*$ and $\lambda = 2\|\xi^*\|_{\lambda(T+1)}$, in (41), we have

$$
\begin{align*}
\Phi(\lambda(t+1)) - \Phi(\lambda^*) \\
- 2\|\xi_f\|_{M \lambda(T+1)} \\
\leq \|\lambda - \lambda(t+1)\|^2_{D_l^{[\gamma_l]}[\xi(t)+MA(t+1)]} \\
- \|\lambda - \lambda(t+1)\|^2_{D_l^{[\gamma_l]}[\xi(t)+MA(t+1)]} \\
+ \|\xi - \xi(t+1)\|^2_{D_l^{[\gamma_l]}[\xi(t)+MA(t+1)]}.
\end{align*}
$$

(44)

where $c_i \leq \frac{1}{h_i}$ is considered. Summing up (44) over $t = 0, 1, ..., T$ gives

$$
(T+1)(\Phi(\lambda(T+1)) - \Phi(\lambda^*) + 2\|\xi^*\||M \lambda(T+1)|| \\
\leq \sum_{t=0}^{T}(\Phi(\lambda(t+1)) - \Phi(\lambda^*) + 2\|\xi^*\||M \lambda(T+1)|| \\
\leq \|\xi^*\|_{M \lambda(T+1)} - \xi(0)^2_{D_l^{[\gamma_l]}[\lambda(T+1)]} \\
+ \|\lambda - \lambda(0)^2_{D_l^{[\gamma_l]}[\lambda(T+1)]} + \sum_{t=0}^{T}|M \lambda(t+1)|^2_{D_l^{[\gamma_l]}[\lambda(T+1)]}
$$

(45)
By combining the first inequality in (46) and (47), we have
\[
\|\lambda - \lambda(0)\|^2_{D_{\gamma}^{M+\beta} [\frac{1}{\alpha}]} + \|\xi - \lambda(0)\|^2_{D_{\gamma}^{\beta} [\frac{1}{\alpha}]} + (T + 1)\gamma_{\text{max}} N\|L\|^2T^2, \tag{45}
\]
where the first inequality is from the convexity of \(\Phi\) and the third inequality is from Cauchy-Schwarz inequality and \(\|M\lambda\|^2_{D_{\gamma}^\beta [\xi]} = \|L\|^2\|\theta\|^2 \leq \gamma_{\text{max}}\|L\|^2\|\theta\|^2 \leq \gamma_{\text{max}} N\|L\|^2T^2\).
Therefore,
\[
\Phi(\lambda(T+1)) - \Phi(\lambda^*) \leq \frac{1}{T+1}(\|\lambda - \lambda(0)\|^2_{D_{\gamma}^{M+\beta} [\frac{1}{\alpha}]} + \|\xi - \lambda(0)\|^2_{D_{\gamma}^{\beta} [\frac{1}{\alpha}]} + (T + 1)\gamma_{\text{max}} N\|L\|^2T^2 - 2\|\xi\|\|M\lambda(T+1)\|)
\]
\[
\leq \Theta(c_1,\ldots,c_N,\gamma_1,\ldots,\gamma_N) + O(\gamma_{\text{max}}). \tag{46}
\]
By letting \(\lambda = \bar{\lambda}(T+1)\) in (43), we have
\[
\Phi(\bar{\lambda}(T+1)) - \Phi(\lambda^*) \geq -\|\xi\|\|M\bar{\lambda}(T+1)\|. \tag{47}
\]
By combining the first inequality in (46) and (47), we have
\[
\|\xi\|\|M\bar{\lambda}(T+1)\| \leq \Theta(c_1,\ldots,c_N,\gamma_1,\ldots,\gamma_N) + O(\gamma_{\text{max}}). \tag{48}
\]
By (47) and (48), we have
\[
\Phi(\bar{\lambda}(T+1)) - \Phi(\lambda^*) \geq -\Theta(c_1,\ldots,c_N,\gamma_1,\ldots,\gamma_N) - O(\gamma_{\text{max}}). \tag{49}
\]
By combining (46), (49) and (49), (26) and (27) are proved.

2) Scenario 2: If \(M\bar{\lambda}(T+1) = 0\), we let \(\lambda = \lambda^*\) and \(\xi = 2\xi^*\) in (47), which directly gives
\[
\Phi(\lambda(T+1)) - \Phi(\lambda^*) \leq \Theta(c_1,\ldots,c_N,\gamma_1,\ldots,\gamma_N) + O(\gamma_{\text{max}}) \tag{50}
\]
by the same derivation process of (46). Then, considering that \(\|M\bar{\lambda}(T+1)\|^2 = 0\) and \(\Phi(\bar{\lambda}(T+1)) - \Phi(\lambda^*) \geq -\|\xi^*\|\|M\lambda(T+1)\| = 0\), (26) and (27) hold as well.

REFERENCES

[1] V. Lesser, C. L. Ortiz Jr, and M. Tambe, *Distributed sensor networks: A multiagent perspective*. Springer Science & Business Media, 2003, vol. 9.

[2] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Spreading up distributed machine learning using codes,” *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[3] L. Bai, M. Ye, C. Sun, and G. Hu, “Distributed economic dispatch control via saddle point dynamics and consensus algorithms,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 898–905, 2017.

[4] A. Beck and M. Teboulle, “A fast dual proximal gradient algorithm for convex minimization and applications,” *Operations Research Letters*, vol. 42, no. 1, pp. 1–6, 2014.

[5] C. Hans, “Bayesian lasso regression,” *Biometrika*, vol. 96, no. 4, pp. 835–845, 2009.

[6] S.-Y. Zhao, R. Xiang, Y.-H. Shi, P. Gao, and W.-J. Li, “Scope: Scalable composite optimization for learning on spark,” in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] Y. Wang, W. Yin, and J. Zeng, “Global convergence of admm in nonconvex nonsmooth optimization,” *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[8] Y. Nesterov, “Primal-dual subgradient methods for convex problems,” *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[9] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[10] A. Beck and M. Teboule, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[11] J. Wang and G. Hu, “Composite optimization with coupling constraints via penalized proximal gradient method in partially asynchronous networks,” *arXiv preprint arXiv:2102.12816v5*, 2021.

[12] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[14] D. P. Bertsekas, “Incremental proximal methods for large scale convex optimization,” *Mathematical Programming*, vol. 129, no. 2, p. 163, 2011.

[15] B. Ning, Q.-L. Han, and Z. Zuo, “Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach,” *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 122–132, 2017.

[16] X. Wang, Y. Hong, and H. Ji, “Distributed optimization for a class of nonlinear multiagent systems with disturbance rejection,” *IEEE Transactions on Cybernetics*, vol. 46, no. 7, pp. 1655–1666, 2015.

[17] Y. Pang and G. Hu, “Randomized gradient-free distributed optimization methods for a multiagent system with unknown cost function,” *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 333–340, 2019.

[18] B. Ning, Q.-L. Han, and Z. Zuo, “Distributed optimization of multiagent systems with preserved network connectivity,” *IEEE Transactions on Cybernetics*, vol. 49, no. 11, pp. 3980–3990, 2018.

[19] X. Wang, J. Yan, B. Jin, and W. Li, “Distributed and parallel admm for structured nonconvex optimization problem,” *IEEE Transactions on Cybernetics*, 2019.

[20] I. Notarnicola and G. Notarstefano, “Asynchronous distributed optimization via randomized dual proximal gradient,” *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2095–2106, 2016.

[21] D. Kim and J. A. Fessler, “Fast dual proximal gradient algorithms with rate \(o(1/k^{1.5})\) for convex minimization,”
arXiv preprint arXiv:1609.09441, 2016.

[22] D. Mateos-Núñez and J. Cortés, “Distributed saddle-point subgradient algorithms with laplacian averaging,” IEEE Transactions on Automatic Control, vol. 62, no. 6, pp. 2720–2735, 2016.

[23] I. Neccoara, V. Nedelcu, D. Clipici, and L. Toma, “On fully distributed dual first order methods for convex network optimization,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 2788–2793, 2017.

[24] I. Notarnicola and G. Notarstefano, “Constraint-coupled distributed optimization: a relaxation and duality approach,” IEEE Transactions on Control of Network Systems, vol. 7, no. 1, pp. 483–492, 2019.

[25] X. Li, G. Feng, and L. Xie, “Distributed proximal algorithms for multi-agent optimization with coupled inequality constraints,” IEEE Transactions on Automatic Control, 2020.

[26] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, “Distributed constrained convex optimization and consensus via dual decomposition and proximal minimization,” in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016, pp. 1889–1894.

[27] ———, “Dual decomposition for multi-agent distributed optimization with coupling constraints,” Automatica, vol. 84, pp. 149–158, 2017.

[28] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” IEEE Transactions on Automatic Control, vol. 57, no. 1, pp. 151–164, 2011.

[29] T.-H. Chang, “A proximal dual consensus admm method for multi-agent constrained optimization,” IEEE Transactions on Signal Processing, vol. 64, no. 14, pp. 3719–3734, 2016.

[30] T.-H. Chang, A. Nedić, and A. Scaglione, “Distributed constrained optimization by consensus-based primal-dual perturbation method,” IEEE Transactions on Automatic Control, vol. 59, no. 6, pp. 1524–1538, 2014.

[31] A. Simonetto and H. Jamali-Rad, “Primal recovery from consensus-based dual decomposition for distributed convex optimization,” Journal of Optimization Theory and Applications, vol. 168, no. 1, pp. 172–197, 2016.

[32] I. Neccoara and V. Nedelcu, “On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems,” Automatica, vol. 55, pp. 209–216, 2015.

[33] F. R. Chung and F. C. Graham, Spectral graph theory. American Mathematical Soc., 1997, no. 92.

[34] J. Borwein and A. S. Lewis, Convex analysis and nonlinear optimization: theory and examples. Springer Science & Business Media, 2010.

[35] A. Beck, First-order methods in optimization. SIAM, 2017.

[36] S. Boyd, S. P. Boyd, and L. Vandenberghe, Convex optimization. Cambridge University Press, 2004.

[37] W. Shi, Q. Ling, G. Wu, and W. Yin, “A proximal gradient algorithm for decentralized composite optimization,” IEEE Transactions on Signal Processing, vol. 63, no. 22, pp. 6013–6023, 2015.

[38] J. Wang and G. Hu, “Composite optimization with coupling constraints via dual proximal gradient method with applications to asynchronous networks,” arXiv preprint arXiv:2102.12797v5, 2021.

[39] M. Schmidt, N. Roux, and F. Bach, “Convergence rates of inexact proximal-gradient methods for convex optimization,” Advances in Neural Information Processing Systems, vol. 24, pp. 1458–1466, 2011.

[40] T.-H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus admm,” IEEE Transactions on Signal Processing, vol. 63, no. 2, pp. 482–497, 2014.

[41] M. I. Florea and S. A. Vorobyov, “A generalized accelerated composite gradient method: Uniting nesterov’s fast gradient method and fista,” IEEE Transactions on Signal Processing, vol. 68, pp. 3033–3048, 2020.

[42] R. T. Rockafellar, Convex analysis. Princeton university press, 1970, no. 28.