Distributed 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 affine coupling constraints 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 distributed dual proximal gradient algorithm, where the agents make decisions based on the information of immediate neighbors. 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, by which the overall computational complexity can be reduced. An ergodic convergence rate of the proposed algorithm is derived and the feasibility is numerically verified by solving a social welfare optimization problem in the electricity market.

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

I. INTRODUCTION

DECENTRALIZED optimization has become an active topic in recent years for solving various engineering problems, such as detection and localization in sensor networks [1], machine learning problems [2], and economic dispatch in power systems [3]. As a typical optimization procedure, each agent usually maintains an individual decision variable and the global optimal solution can be obtained with multiple rounds of communication and decision making. 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 resource allocation problems [4], least absolute shrinkage and selection operator (LASSO) regressions [5], and support vector machines [6]. To solve these problems, widely discussed techniques include alternating direction method of multipliers, primal-dual subgradient method, and proximal gradient method, etc.

The majority of 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 [7]. To overcome this issue, a valid alternative is applying graph theory in modeling the communication links, leading to the distributed setup where the agents only communicate with their immediate neighbors [8]. However, with the increasing demand in the computational efficiency of various fields, more explorations on the algorithm development for distributed optimization problems (DOPs) are required [9]. Observing that proximal gradient method is usually numerically more stable than the subgradient-based counterpart in composite optimization problems [10], 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 [11][12], where the agents make decisions with local variables and certain agreement on the optimal solution is achieved through local communication. Alternatively, we focus on optimizing a class of composite DOPs subject to affine coupling constraints, where the global objective functions are the sum of the local cost functions. To solve the problems of interest, some primal-dual subgradient methods were studied in the past few years, where the average consensus technique was employed for the iteration of local variables [13][14]. Alternatively, by formulating some local subproblems in the iteration steps, dual decomposition methods were actively investigated in the recent works, where the agents exchange the dual information with their neighbors based on the result of subproblems [15][21]. When it comes to composite optimizations, dual proximal gradient method was exploited recently by applying the proximal gradient method to the dual problems as in [22][23][24], where, however, no general affine coupling constraints were considered in the distributed computations.

Different from the aforementioned works, we aim to incorporate the dual proximal gradient method in a distributed setup with general affine coupling constraint. Then, a distributed dual proximal gradient (DDPG) algorithm is proposed, by which the computational complexity of the local computations can be reduced if the non-smooth parts of the cost functions are with some simple structures. To highlight of features of the DDPG algorithm, the comparisons with some state-of-the-art works with similar problem setups are summarized as follows.
• One key feature of the proposed DDPG algorithm is that one only needs to update the dual variables by some simple operations provided that the proximal mapping of the non-smooth parts in the primal cost functions can be explicitly derived, which is different from [16, 26, 27]. This settlement can be efficient in the sense that no costly inner-loop optimizations of the primal or other auxiliary variables are required. In addition, compared with the algorithms without inner-loop optimizations, no explicit convergence rates were provided in [13, 14]. By contrast, an asymptotic convergence is ensured by the DDPG algorithm with an ergodic convergence rate $O(1/T)$ for the dual function value.

• In terms of the mathematical assumptions, the algorithms in [13, 14, 17, 20, 25, 26] require some compact constraints on the primal variables and [13, 28] assume some bounded subgradients to ensure the convergence of their algorithms. By contrast, this work focuses on dual sequences without the boundedness requirement on the primal variables or subgradients. Also, the objective functions in [13, 14] are assumed to be continuous, while the DDPG algorithm allows the non-smooth parts of the objective functions to be lower semi-continuous.

The contributions of this work are summarized as follows.

• We consider a class of composite DOPs with affine coupling constraints. A DDPG algorithm is proposed by deriving the dual problem based on Fenchel conjugate, where the optimal solution can be obtained when the agents update only with the dual information of immediate neighbors, leading to a distributed computation environment.

• The proposed DDPG algorithm only requires the update of dual variables by some simple operations if the non-smooth parts of the cost functions are simple-structured, which can reduce the overall computational complexity. Then, some cost functions with more complicated structures are discussed by tolerating some inner-loop optimizations in each iteration. An ergodic convergence rate of the DDPG algorithm is derived.

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 DDPG algorithm is proposed based on the dual problem. The convergence analysis is conducted in Section V. The feasibility of the proposed algorithm is verified by a numerical simulation in Section VI. Section VII concludes this paper.

II. Preliminaries

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

A. Notations

$\mathbb{N}$ and $\mathbb{N}_+$ denote the non-negative and positive integer spaces, respectively. Let $|A|$ be the size of set $A$. Operator $(\cdot)^{\top}$ represents the transpose of a matrix. $A_1 \times A_2$ denotes the Cartesian product of sets $A_1$ and $A_2$. relint $A$ represents the relative interior of set $A$. Let $\|u\|_{L_2} = u^{\top}Xu$ with $X$ being a square matrix. $X \succ 0$ ($\succeq 0$) means that the square matrix $X$ is positive definite (semi-definite). $\tau(X)$ and $\tau_2(X)$ represent the largest and smallest eigenvalues of $X$, respectively. $\otimes$ denotes Kronecker product. $I_n$ is an $n$-dimensional identity matrix and $O_{n \times m}$ indicates an $(n \times m)$-dimensional zero matrix. $1_n$ and $0_n$ denote the $n$-dimensional column vectors with all elements being 1 and 0, respectively.

B. Graph Theory

Define an undirected graph $G = \{V, E\}$ for a multi-agent network, where $V = \{1, 2, ..., N\}$ is the set of vertices and $E = \{e_1, e_2, ..., e_{|E|}\} \subseteq \{(i,j)|i,j \in V \text{ and } i \neq j\}$ (no self-loop) is the set of edges with $(i,j) \in E$ unordered. Given the index of vertices, the index of edges is determined as follows. For any two distinct edges $e_k = (k_1, k_2) \in E$ and $e_\ell = (v_1, v_2) \in E$, if $\min\{k_1, k_2\} > \min\{v_1, v_2\}$, then $\ell > k$, and vice versa. For the case where $\min\{k_1, k_2\} = \min\{v_1, v_2\}$, if $\max\{k_1, k_2\} > \max\{v_1, v_2\}$, then $\ell > k$, and vice versa. $G$ is connected if any two distinct vertices are linked by at least one path. $V_i = \{j|(i,j) \in E\}$ represents the neighbor set of agent $i$. Let $Q$ be the incidence matrix of $G$. The $(j,k)$th element of $Q$ is defined by

$$Q_{jk} = \begin{cases} 1 & \text{if } e_k = (j,l) \in E \text{ and } j < l \\ -1 & \text{if } e_k = (j,l) \in E \text{ and } j > l \\ 0 & \text{otherwise} \end{cases}.$$ 

Then, it can be checked that $V_i = S_i \cup S_i^\complement$, $i \in V$.

C. Proximal Mapping

A proximal mapping of a proper, convex, and closed function $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is defined by $\text{prox}_\alpha^\psi(v) = \arg \min_u (\psi(u) + \frac{1}{2\alpha}\|u - v\|^2)$, $\alpha > 0$, $v \in \mathbb{R}^n$.

D. Fenchel Conjugate

$$\psi : \mathbb{R}^n \to (-\infty, +\infty]$$ is a proper function. Then the Fenchel conjugate of $\psi$ is defined by $\psi^\psi(v) = \sup_u \{v^\top u - \psi(u)\}$, which is convex [30, Sec. 3.3].

Lemma 1. (Extended Moreau Decomposition [37, Thm. 6.45]) $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is a proper, convex, and closed function. $\psi^\psi$ is its Fenchel conjugate. Then, for certain $v \in \mathbb{R}^n$ and $\alpha > 0$, we have $v = \alpha \text{prox}_{\psi^\psi}^\psi$.

Lemma 2. [22] Lemma V7 $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is a proper, closed, $\sigma$-strongly convex function and $\psi^\psi$ is its Fenchel conjugate, $\sigma > 0$. Then, $\arg \max \{v^\top u - \psi(u)\} = \nabla_v \psi^\psi(v)$ and $\nabla_\psi \psi^\psi(v)$ is Lipschitz continuous with Lipschitz constant $\frac{1}{\sigma}$. 

\footnote{For the DOPs with smooth cost functions, some existing works on dual algorithms, e.g., [21], also 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 functions requires an additional nontrivial optimization process. Therefore, the contribution to the computational efficiency of this work is established for possibly non-smooth cost functions.}
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 $G = \{V, E\}$, $x_i \in \mathbb{R}^M$, $x = [x_1^T, ..., x_N^T]^T \in \mathbb{R}^{NM}$. Agent $i$ maintains a 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 $X = X_1 \times X_2 \times ... \times X_N \subseteq \mathbb{R}^{NM}$. An affine-constrained optimization problem of $V$ can be given by

(P1) $\min_{x \in X} \sum_{i \in V} F_i(x_i) \quad \text{s.t.} \quad Ax = b,$

which is equivalent to

(P2) $\min_{x} \sum_{i \in V} (F_i(x_i) + \|x_i\|_{X_i}(z_i)) \quad \text{s.t.} \quad Ax = b,$

with $A \in \mathbb{R}^{B \times NM}$, $b \in \mathbb{R}^B$, $\|x_i\|_{X_i}(z_i) = \begin{cases} 0 & \text{if } x_i \in X_i \\ +\infty & \text{otherwise} \end{cases}.$

Assumption 1. $G$ is connected and undirected.

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

The assumptions in Assumption 2 are often discussed in composite optimization problems [4][22][32][56].

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

By Assumption 3, $\|X_i\|_{X_i}$ is proper, convex, and closed [37], which complies with the assumption on $g_i$.

IV. Distributed Optimization Algorithm Development

The proposed algorithm for solving the problem of interest is presented in this section.

A. Dual Problem

Decoupling the objective function of Problem (P2) gives

(P3) $\min_{x,z} \sum_{i \in V} (f_i(x_i) + (g_i + \|x_i\|_{X_i})(z_i)) \quad \text{s.t.} \quad Ax = b,$

where $z = [z_1^T, ..., z_N^T]^T \in \mathbb{R}^{NM}$ with $z_i \in \mathbb{R}^M$ being a slack variable. The Lagrangian function of Problem (P3) is

$L(x, z, \eta, \mu) = \sum_{i \in V} (f_i(x_i) + (g_i + \|x_i\|_{X_i})(z_i)) + \mu_i^T (x_i - z_i) + \eta^T (Ax - b) = \sum_{i \in V} (\mu_i^T x_i + x_i^T (A_i^T \eta + \mu_i)) + (g_i + \|x_i\|_{X_i})(z_i) - z_i^T \mu_i - b^T \eta.$

Here, $\mu_i \in \mathbb{R}^M$ and $\eta \in \mathbb{R}^B$ are the Lagrangian multipliers associated with constraints $x_i = z_i$ and $Ax = b$, respectively, $\mu = [\mu_1^T, ..., \mu_N^T]^T \in \mathbb{R}^{NM}$, and $A_i \in \mathbb{R}^{B \times M}$ is the $i$th column sub-block of $A$ with $A = [A_1, ..., A_i, ..., A_N]$.

Then the dual function can be obtained by minimizing $L(x, z, \eta, \mu)$ with $x$ and $z$ [27], which is $D(\eta, \mu) = \min L(x, z, \eta, \mu) = \min \sum_{i \in V} (f_i(x_i) - x_i^T H_i \eta_i + (g_i + \|x_i\|_{X_i})(z_i) - z_i^T F_i \eta_i - \kappa_i \theta_i) = \sum_{i \in V} -f_i^\diamond (H_i \eta_i) - \kappa_i \theta_i = \eta^T (F_i^\diamond \theta_i) - (\kappa_i + \|x_i\|_{X_i}) \theta_i,$

with $H_i = [-A_i^T, -I_M] \in \mathbb{R}^{M \times (M+B)}$, $\theta_i = [\eta_i^T, \mu_i^T]^T \in \mathbb{R}^{M+B}$, $F_i = [O_M \otimes B_i, I_M] \in \mathbb{R}^{M \times (M+B)}$, $E_i = \{b_i, 0_M\} \in \mathbb{R}^{B \times (M+B)}$, $\sum_{i \in V} \kappa_i = 1$, and $(g_i + \|x_i\|_{X_i})^\diamond$ denotes the Fenchel conjugate of $g_i + \|x_i\|_{X_i}$. Hence, the dual problem of Problem (P3) can be formulated as

(P4) $\min_{\theta, \eta} \sum_{i \in V} (f_i^\diamond (H_i \eta_i) + \kappa_i \theta_i + (g_i + \|x_i\|_{X_i})^\diamond (F_i \eta_i)),$

where $\zeta = [\zeta_1^T, ..., \zeta_N^T]^T \in \mathbb{R}^{NB+NM}$.

We aim to solve Problem (P4) in a distributed manner. In Problem (P4), the variables of $f_i^\diamond (H_i \eta_i)$ are coupled in terms of the common component $\eta_i$ in $\zeta_i$, but those of $(g_i + \|x_i\|_{X_i})^\diamond (F_i \eta_i)$ are decoupled since $F_i \eta_i = \mu_i$. In the following, we define $\lambda_i = [\theta_i^T, \mu_i^T]^T$ and $\lambda = [\lambda_1^T, ..., \lambda_N^T]^T$, where $\theta_i \in \mathbb{R}^B$ is the local estimate of $\eta_i$, and $\mu_i \in \mathbb{R}^M$ is the local dual estimate of $\lambda_i$.

Then, Problem (P5) is equivalent to

(P5) $\min_{\lambda} \Phi(\lambda) \quad \text{s.t.} \quad K\lambda_i = K\lambda_j, \quad \forall (i, j) \in E,$

where $\Phi(\lambda) = P(\lambda) + Q(\lambda), P(\lambda) = \sum_{i \in V} p_i(\lambda_i), Q(\lambda) = \sum_{i \in V} q_i(\lambda_i), p_i(\lambda_i) = f_i^\diamond (H_i \eta_i) + \kappa_i E_i \lambda_i, q_i(\lambda_i) = (g_i + \|x_i\|_{X_i})^\diamond (F_i \lambda_i), K = [I_B, O_B \otimes M].$ Constraint 2 ensures the partial consistency among $\lambda_i$ in terms of the component $\theta_i$, i.e., $\theta_i = K \lambda_i$. In addition, 1 can be represented by $K \lambda_i = K \lambda_j$, $\forall (i, j) \in E$. If $S_i \neq \emptyset$, which can be written into a compact form $M \lambda = 0$, where $M = Q^\perp \otimes \mathbb{K}$ [29]. Then, Problem (P5) is equivalent to

(P6) $\min_{\lambda} \Phi(\lambda) \quad \text{s.t.} \quad M \lambda = 0.$

Let $\lambda^* = [\lambda_1^*, ..., \lambda_N^*] \in \mathbb{R}^{NM}$ be the optimal solution to Problem (P6) with $\lambda_i^* = [\theta_i^*, \mu_i^*]^T$. The Lagrangian function of Problem (P6) can be given by

$L(\lambda, \xi, \bar{x}) = P(\lambda) + Q(\lambda) + \xi^T \lambda \lambda,$

where $\xi = [\xi_1^T, ..., \xi_N^T]^T$, $\xi_i = [\xi_{i1}, ..., \xi_{iS_i}]^T$ with $\xi_{ij} \in \mathbb{R}^B$ being the Lagrangian multiplier associated with the constraint $K \lambda_i - K \lambda_j = 0$, $\bar{x} \in S_i$, and $i \in \{1, 2, ..., |S_i|\}$ is the index. Let $C$ be the set of the saddle points of $L(\lambda, \xi)$. Then any saddle point $(\lambda^*, \xi^*) \in C$ satisfies 38

$L(\lambda, \xi^*) \geq L(\lambda^*, \xi^*) \geq L(\lambda, \xi).$

We aim to seek a saddle point of $L(\lambda, \xi)$, which can be characterized by Karush-Kuhn-Tucker (KKT) conditions [37],

$0 \in \nabla \lambda P(\lambda^*) + \partial \lambda Q(\lambda^*) + M^T \xi^*, \quad M \lambda^* = 0.$

(6)
B. DDPG Algorithm

Based on the previous discussion, the DDPG algorithm for solving Problem (P6) is designed as
\[
\lambda^{t+1} = \arg\min_{\lambda} \left\{ L^{t} - c(\nabla_{\lambda} p_{i}(\lambda^{t}) + M^{T}\xi^{t} + \gamma M^{T}M\lambda^{t}) \right\},
\]
which means
\[
\lambda^{t+1} = \arg\min_{\lambda} \left\{ \nabla_{\lambda} p_{i}(\lambda^{t}) + \frac{\gamma}{2} \sum\mathbf{K}^{T}\mathbf{K}(\lambda^{t} - \lambda_{j}^{t}) \right\}, \forall i \in V,
\]
where \( \mathbf{K} \) is the incidence matrix.

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

Algorithm 1 DDPG Algorithm

1: Initialize \( \lambda^{0}, \xi^{0} \). Determine step sizes \( c, \gamma > 0 \).
2: for \( t = 0, 1, 2, \ldots \) do
3: \hspace{1em} for \( i = 1, 2, \ldots, N \) do (in parallel)
4: \hspace{2em} Update \( \lambda^{t+1} \) based on (9).
5: \hspace{2em} Update \( \xi^{t+1} \) based on (10), \( \forall j \in S_{i} \).
6: \hspace{1em} end for
7: end for
8: Obtain the outputs \( \lambda^{\text{out}} \) and \( \xi^{\text{out}} \) under certain convergence criterion and calculate the primal solution by \( \lambda^{\text{out}} = \arg\min_{x} f_{i}(x_{i}) - x_{i}^{T}H_{i}\lambda^{\text{out}}, \forall i \in V, j \in S_{i} \).

V. MAIN RESULT

The convergence analysis and computational complexity analysis of the DDPG algorithm are conducted in this section.

A. Convergence Analysis

Theorem 1. Suppose that Assumptions 1-3 hold. If the step sizes satisfy
\[
c^{-1} \geq h + \gamma \tau (M^{T}M)
\]
with \( h = \max_{i \in V} \{ h_{i} \} \), then the states generated by Algorithm 1 converge to a primal-dual solution to Problem (P6).

See the proof in Appendix A.

Theorem 2. Suppose that Assumptions 1-3 hold and the step sizes are selected based on (14). By Algorithm 1, for certain \( (\lambda^{\ast}, \xi^{\ast}) \in C \), we have \( \|\Phi(\lambda^{t+1}) - \Phi(\lambda^{\ast})\| \leq O(\frac{1}{t}) \) and \( \|\xi^{\ast}\| 2^{T+1} \leq O(\frac{1}{t}) \), where \( \lambda^{T+1} = \frac{1}{1+T} \sum_{t=0}^{T} \lambda^{t+1} \) and \( T \in \mathbb{N}_{+} \).

See the proof in Appendix B.

B. Computational Complexity Analysis with Simple-Structured Cost Functions

In this work, the concept of computational complexity measures the total amount of basic operations required, which is dominated by the iteration complexity and computational cost per iteration [39].

In the following, we first discuss the iteration complexity of (9). Specifically, to apply (9), one needs to compute (i) \( \nabla_{p_{i}} \) and (ii) the proximal mapping of \( g_{i}, i \in V \). For (i), \( \nabla_{p_{i}} \) can be efficiently obtained given that \( f_{i} \) is simple-structured and, consequently, \( \nabla_{p_{i}} f_{i} \) can be analytically derived, e.g., \( f_{i} \) is a quadratic function [37, 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 efficiently obtained [38], we have \( \text{prox}_{g_{i}} = \text{prox}_{g_{i} + I_{X_{i}}} \), where

3This case is based on that \( g_{i} + I_{X_{i}} \) is with certain simple structure, which is often the assumption in the works on proximal gradient method. See some frequently used formulas in [31, Sec. 6.3] and applications in [40, Sec. 7].
prox_{g_i + I_{X_i}} \gamma c can be obtained by calculating prox_{g_i + I_{X_i}} \gamma c based on Lemma 1. Then, by decomposing $\lambda_i^{t+1}$, (9) can be modified into

$$\theta_i^{t+1} = \theta_i^t - c \nabla_{\theta_i} p_i(\lambda_i^t) + \sum_{j \in S_i} \xi_i - \sum_{j \in S_i} \xi_i + \gamma \sum_{j \in V_i} (\theta_i^t - \theta_j^t)$$

and

$$\mu_i^{t+1} = \text{prox}_{g_i} [\mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t)]$$

$$= \mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t) - c \text{prox}_{g_i}^{-1} \left[ \mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t) \right] c$$

(15)

where $g_i^t = (g_i + I_{X_i})^{\infty} = g_i + I_{X_i}$ due to the convexity and lower semi-continuity of $g_i + I_{X_i}$ and $(g_i + I_{X_i})^{\infty}$ is the biconjugate of $g_i + I_{X_i}$ [37, Sec. 3.3.2]. Then, the calculation of the proximal mapping of $(g_i + I_{X_i})$ can be avoided as shown in (15), which can reduce the iteration complexity if the proximal mapping of $g_i + I_{X_i}$ is easier to obtain. For instance, in an LASSO problem with penalty $g_i(x_i) = ||x_i||_1$ and $X_i = \mathbb{R}^M$, the proximal mapping of $I_1$-norm is a soft thresholding operator with analytical solution [31, Sec. 6.3].

In addition, if $g_i = 0$, (15) can be written as $\mu_i^{t+1} = \mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t) - c \text{prox}_{g_i}^{-1} \left[ \mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t) \right] c - c \Pi_{X_i} \left[ \mu_i^t - c \nabla_{\mu_i} p_i(\lambda_i^t) \right] c$, where $\Pi_{X_i} [\cdot]$ is a Euclidean projection onto $X_i$ [40, Sec. 1.2].

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 a Euclidean e-norm: $g_i(x_i) = ||x_i||_e$, $X_i = \mathbb{R}^M$. Then, we can have $g_i(\lambda_i) = g_i^t(\mu_i) = \{0$ if $\mu_i \in W_i = \{0$ if $\lambda_i \in Y_i = +\infty$ otherwise $\}$, $Y_i = \mathbb{R}^B \times W_i$. The second equality holds by computing the conjugate of a norm [37, Sec. 3.3.1]. Then, in (9), the proximal mapping of $g_i$ is a Euclidean projection onto $Y_i$ [40, Sec. 1.2].

3) Case 3: If $g_i$ is with certain complicated structure, as a general method, we can construct a strongly convex non-smooth $g_i$ (e.g., shift a strongly convex component of $f_i$ to $g_i$). Then, rewrite (9) by the definition of proximal mapping, which gives

$$\lambda_i^{t+1} = \arg \min_{\lambda_i} \lambda_i - \lambda_i + \sum_{j \in S_i} K_i \xi_j - \sum_{j \in S_i} K_i \xi_j + \sum_{j \in V_i} K_i (\lambda_i^t - \lambda_j^t)$$

To obtain the result, one can utilize a gradient descent method by computing $\nabla_{\lambda_i} (\lambda_i^t + \lambda_i)^{\infty}(F_{\lambda_i}) = F^T \nabla_{\lambda_i} (\lambda_i^t + \lambda_i)^{\infty}(F_{\lambda_i}) = F^T \arg \max_u (F_{\lambda_i})^T u - (\lambda_i + \Pi_{X_i}(u))$ with the help of Lemma 2. In this case, the update of $\lambda_i$ requires an inner-loop optimization to compute the gradient of $q_i$, which can be completed by agent $i$ locally.

In Cases 1 and 2, (9) can only involve some simple operations (e.g., addition, multiplication, and Euclidean projection with iteration complexity $O(1)$) without any costly inner-loop optimization of primal or other auxiliary variables, which results in an overall iteration complexity $O(1)$ with an ergodic convergence error $\epsilon$ in dual function value (see Thm. 2). In addition, note that the computational cost per iteration is linear in the dimension of $\lambda$ and $\xi$ [39], then the overall computational complexity of the DDPG algorithm can be $O(NM + NB) + O(\frac{B \sum_{i \in V} |S_i|}{\epsilon})$.

VI. NUMERICAL RESULT

In this section, we demonstrate the feasibility of the DDPG algorithm by solving a social welfare optimization problem in an electricity market, where the overall cost function is the sum of the objective function of all the agents [41].

A. Simulation Setup

Define $V_{UC}$ and $V_{user}$ as the sets of utility companies (UCs) and users, respectively. Define $x = [x_{UC1}^T, ..., x_{UCn}^T, x_{user1}^T, ..., x_{userm}^T]^T$, where $x_{UC}$ is the energy generation quantity of UC $i$ and $x_{user}$ is the demand of user $j$, $i \in V_{UC}$, $j \in V_{user}$. $\phi_i(x_{UC})$ and $\omega_j(x_{user})$ are the cost function of UC $i$ and the utility function of user $j$, respectively. Then the social welfare optimization problem of the market can be formulated as

$$\min_{x} \sum_{i \in V_{UC}} \phi(x_{UC}) - \sum_{j \in V_{user}} \omega_j(x_{user})$$

s.t.

$$\begin{align*}
\sum_{i \in V_{UC}} x_{UC}^i &= \sum_{j \in V_{user}} x_{user}^j, \\
x_{UC}^i &\in X_{UC}, x_{user}^j \in X_{user}, \forall i \in V_{UC}, j \in V_{user}, \end{align*}$$

(16)

where $\phi_i(x_{UC}) = \delta_i(x_{UC})^2 + \zeta_i x_{UC}^2 + \beta_i$, $\omega_j(x_{user}) = \chi_j x_{user}^2 - \pi_j x_{user}^2$, with $\delta_i, \zeta_i, \beta_i, \chi_j$, and $\pi_j$ being parameters, whose values are set in Table 1 [42]. (16) is the supply-demand balance constraint. $x_{UC}^i = [0, x_{UC,\text{max}}]$, and $x_{user}^j = [0, x_{user,\text{max}}]$ are local constraints with $x_{UC,\text{max}}, x_{user,\text{max}} > 0$. Define $A = [I_{V_{UC}}, -I_{V_{user}}]$. Then (16) is equivalent to $Ax = 0$.

To show the performance of Algorithm 1, we consider the communication topology shown in Fig. 1. Similar to the derivation procedure of (1), the Lagrangian function of Problem (P7) can be obtained as

$$\begin{align*}
L(x, z, \eta, \mu) &= \sum_{i \in V_{UC}} \phi_i(x_{UC}^i) + \sum_{j \in V_{user}} \omega_j(x_{user}^j) + \sum_{j \in V_{user}} (-\omega_j(x_{user}) + \nu_j x_{user}^j) + \eta^T A x + \sum_{i \in V_{UC}} \mu_i (x_{UC}^i - x_{UC}) + \sum_{j \in V_{user}} \rho_j (x_{user} - x_{user})^j, \end{align*}$$

where $z = [z_{UC}^1, ..., z_{UC}^n, z_{user}^1, ..., z_{user}^m]^T$ is a slack...
variable and both \( \eta \) and \( \mu = [\mu_1^{\text{UC}}, ..., \mu_i^{\text{UC}}, \mu_i^{\text{user}}, ..., \mu_{|V_{\text{user}}|}^{\text{user}}] ^T \) are dual variables. Define \( \theta = [\theta_1^{\text{UC}}, ..., \theta_i^{\text{UC}}, \theta_i^{\text{user}}, ..., \theta_{|V_{\text{user}}|}^{\text{user}}] ^T \), which contains the local estimates of \( \eta \).

Let \( \xi = [\xi_1^{\text{UC, user}}, \xi_1^{\text{UC, user}}, ..., \xi_i^{\text{UC, user}}, ..., \xi_{|V_{\text{user}}|}^{\text{UC, user}}] ^T \) be the Lagrangian multiplier defined in (1). For instance, \( \xi_i^{\text{UC, user}} \) is the multiplier associated with constraint \( \theta_i^{\text{UC}} = \theta_i^{\text{user}} \). With some direct calculations, the optimal solution to Problem (P7) is given by \( x^* = [0, 150, 48.5, 50.2, 51.3]^T \).

B. Simulation Result

The simulation result is shown in Figs. 2(a) to 2(c). Fig. 2(a) depicts the trajectory of dual variables \( \theta \) and \( \mu \). It can be seen that all the elements in \( \theta \) converge to \( \eta^* = -8.1 \) while \( \mu \) converges to \( \mu^* = [-0.61, 2.34, 0, 0, 0]^T \). One can check that the optimal primal solution at the saddle point of \( L \) is \( x^* = \operatorname{arg min}_x L(x, z, \eta^*, \mu^*) = [0, 150, 48.5, 50.2, 51.3]^T \), which means that the lower bound and upper bound of \( x_1^{\text{UC}} \) and \( x_2^{\text{UC}} \) are activated, respectively, while other variables reach interior optimal solutions. Fig. 2(b) depicts the trajectory of \( \xi \). Fig. 2(c) shows that the value of dual function \( \Phi(\lambda) \) (as defined in Problem (P6)) converges to approximately 756.53.

VII. CONCLUSION

In this work, we considered solving a composite DOP with affine coupling constraints. A DDPG algorithm was proposed by resorting to the dual problem. Compared with the existing research works with similar problem setups, we showed that if the cost functions are with some simple structures, one only needs to update the dual variables by some simple operations, which leads to the reduction of overall computational complexity.

APPENDIX

A. Proof of Lemma 3

By Lemma 2, \( \nabla f_{i}^{\phi} \) is Lipschitz continuous with Lipschitz constant \( \frac{1}{\alpha_i} \), which means \( \| \nabla f_i^{\phi}(H_v) - \nabla f_i^{\phi}(H_u) \| \leq \frac{1}{\alpha_i} \| H_v - H_u \| \). Then, \( \nabla \lambda, f_i^{\phi}(H, \lambda_i) \) is Lipschitz continuous with constant \( h_i \) and, therefore, \( \nabla \lambda, f_i^{\phi}(H, \lambda_i) = \nabla f_i^{\phi}(H, \lambda_i) + k_i E \) is also Lipschitz continuous with constant \( h_i \).

B. Proof of Theorem 7

By the first-order optimality condition of (4), we have
\[
0 \in \partial \lambda Q(\lambda^{t+1}) + c^{-1}(\lambda^{t+1} - \lambda^*) + \nabla \lambda P(\lambda^t) + M^T \xi^t + \gamma M^T M \xi^{t+1}.
\]

By the convexity of \( Q(\lambda) \), we have
\[
Q(\lambda) - Q(\lambda^{t+1}) \geq \gamma c^{-1} \langle \lambda - \lambda^{t+1} \rangle^T (\lambda - \lambda^{t+1}) + c^{-1} \langle \lambda - \lambda^{t+1} \rangle^T M^T \xi^{t+1} + \gamma \langle \lambda - \lambda^{t+1} \rangle^T M^T M(\lambda^{t+1} - \lambda^t).
\]

By the convexity and Lipschitz continuous differentiability of \( p_i \), we have
\[
\begin{align*}
& (\lambda - \lambda^{t+1})^T \nabla \lambda P(\lambda^t) \\
= & \sum_{i \in V} (\lambda_i - \lambda_i^t)^T \nabla \lambda p_i (\lambda_i^t) + \sum_{i \in V} (\lambda_i - \lambda_i^t)^T \nabla \lambda p_i (\lambda_i^t) \\
\leq & \sum_{i \in V} (\lambda_i - \lambda_i^{t+1})^T \xi_i + \sum_{i \in V} (\lambda_i - \lambda_i^{t+1})^T (\lambda_i - \lambda_i^t) \\
& + \sum_{i \in V} \frac{h_i}{2} \| \lambda_i - \lambda_i^{t+1} \|^2 \\
\leq & P(\lambda) - P(\lambda^{t+1}) + \frac{h}{2} \| \lambda^t - \lambda^{t+1} \|^2.
\end{align*}
\]

By (8), we have \( \lambda^t = \gamma^{-1} (\xi^t - \xi^{t+1}) + M \lambda^{t+1} \). Multiplying the both sides of the above equation by \( (\xi - \xi^{t+1})^T \) gives
\[
\gamma^{-1} (\xi - \xi^{t+1})^T (\xi^t - \xi^{t+1}) + (\xi - \xi^{t+1})^T M \lambda^{t+1} = 0.
\]

By adding (18) and (19) together from the both sides, we have
\[
\begin{align*}
& \Phi(\lambda^{t+1}) - \Phi(\lambda) \leq -c^{-1}(\lambda - \lambda^{t+1})^T (\lambda - \lambda^{t+1}) \\
& \quad + (\lambda - \lambda^{t+1})^T M^T \xi^{t+1} + \frac{h}{2} \| \lambda^t - \lambda^{t+1} \|^2 \\
& - \gamma (\lambda - \lambda^{t+1})^T M^T M(\lambda^{t+1} - \lambda^t) \\
& = -c^{-1}(\lambda - \lambda^{t+1})^T (\lambda - \lambda^{t+1}) \\
& - \gamma (\xi - \xi^{t+1})^T (\xi^t - \xi^{t+1}) - (\xi - \xi^{t+1})^T M \lambda^{t+1} + \xi^{t+1}^T M \lambda^{t+1} + M \lambda^{t+1} - \lambda^t \\
& \quad \leq \gamma (\lambda - \lambda^{t+1})^T M^T M(\lambda^{t+1} - \lambda^t).
\end{align*}
\]
Then we can have \(\lambda \in \partial_{\kappa} \lambda(0) + \nabla_{\kappa} P(\lambda) + M^T \xi\). Therefore, \((\lambda, \xi)\) is a saddle point defined by (5) and (6). Define \(a_n = ||\lambda - \lambda_n||^2 + \frac{2}{\gamma} I_n^N + \frac{\gamma}{2} \xi - \xi_n||^2\). Then \(\lim_{n \to \infty} a_n = 0\). Since \(\lim_{n \to \infty} a_n\) exists, we have \(\lim_{n \to \infty} a_n = \lim_{n \to \infty} a_n = 0\), which means \(\lambda_n \to \lambda\) and \(\xi_n \to \xi\). This proves the theorem.

C. Proof of Theorem 2

Note that (21) holds for all \(\lambda\) and \(\xi\). By letting \(\lambda = \lambda^*\) and \(\xi = 2||\xi||M_{X}^{T+1}\) in (21), we have

\[\Phi(\lambda^{T+1}) - \Phi(\lambda^*) + 2||\xi||M_{X}^{T+1} - \lambda^* - 1\frac{I_n^N + \gamma \xi - \xi^{T+1}}{||\xi||}||^2 - \frac{1}{2\gamma} \leq 2||\xi||M_{X}^{T+1} - ||\xi||^2\]

where (25) is considered. Summing up above inequality over \(t = 0, 1, ..., T\), we get

\[\sum_{t=0}^{T} \left( \Phi(\lambda^{T+1}) - \Phi(\lambda^*) + 2||\xi||M_{X}^{T+1} \right) \leq \sum_{t=0}^{T} \left( \Phi(\lambda^{T+1}) - \Phi(\lambda^*) + 2||\xi||M_{X}^{T+1} \right) \leq \sum_{t=0}^{T} \left( \Phi(\lambda^{T+1}) - \Phi(\lambda^*) + 2||\xi||M_{X}^{T+1} \right) \leq \Theta \]

where the first inequality is from the convexity of \(\Phi\) and the third inequality is from Cauchy-Schwarz inequality. Therefore,

\[\Phi(\lambda^{T+1}) - \Phi(\lambda^*) \leq \frac{\Theta}{T+1} - 2||\xi||M_{X}^{T+1} \leq \frac{\Theta}{T+1} \]

By letting \(\lambda = \lambda^{T+1}\) in (23), we have

\[\Phi(\lambda^{T+1}) - \Phi(\lambda^*) \geq -2||\xi||M_{X}^{T+1} \]

By combining (26) and (27), we have

\[\Phi(\lambda^{T+1}) - \Phi(\lambda^*) \leq \frac{\Theta}{T+1} \leq \frac{\Theta}{T+1} \]

Then with (27), we can further have

\[\Phi(\lambda^{T+1}) - \Phi(\lambda^*) \geq -\frac{\Theta}{T+1} \]

The proof is completed based on (26), (28), and (29).

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

[8] 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.

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

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

[11] 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.

[12] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, 2009.

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

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

[15] D. Yuan, S. Xu, and H. Zhao, “Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 6, pp. 1715–1724, 2011.

[16] I. Necsoa, 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] I. Necsoa 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.

[22] 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.

[23] 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.

[24] J. Wang and G. Hu, “Composite optimization with coupling constraints via dual proximal gradient method with applications to asynchronous networks,” International Journal of Robust and Nonlinear Control, 2022, DOI: 10.1002/rnc.6048.

[25] Y. Su, Q. Wang, and C. Sun, “Distributed primal-dual method for convex optimization with coupled constraints,” IEEE Transactions on Signal Processing, 2021.

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

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

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

[29] D. V. Dimarogonas and K. H. Johansson, “Stability analysis for multi-agent systems using the incidence matrix: Quantized communication and formation control,” Automatica, vol. 46, no. 4, pp. 695–700, 2010.

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

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.

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

[39] S. Bubeck, “Convex optimization: Algorithms and complexity,” Foundations and Trends® in Machine Learning, vol. 8, no. 3-4, pp. 231–357, 2015.

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

[41] J. Wang, Y. Pang, and G. Hu, “Social profit optimization with demand response management in electricity market: A multi-timescale leader-following approach,” IEEE Transactions on Control Systems Technology, 2022, DOI: 10.1109/TCST.2022.3154654.

[42] H. Pourbabak, J. Luo, T. Chen, and W. Su, “A novel consensus-based distributed algorithm for economic dispatch based on local estimation of power mismatch,” IEEE Transactions on Smart Grid, vol. 9, no. 6, pp. 5930–5942, 2017.

[43] R. A. Horn, N. H. Rhee, and S. Wasin, “Eigenvalue inequalities and equalities,” Linear Algebra and its Applications, vol. 270, no. 1-3, pp. 29–44, 1998.

[44] R. G. Bartle and D. R. Sherbert, Introduction to real analysis, 2000, vol. 2.

[45] R. T. Rockafellar, “Convex analysis,” in Convex analysis. Princeton university press, 1997.