Distributed Optimization with Coupling Constraints in Multi-Cluster Networks Based on Dual Proximal Gradient Method

Jianzheng Wang, Guoqiang Hu, Senior Member, IEEE

Abstract—In this work, we consider solving a distributed optimization problem in a multi-agent network with multiple clusters. In each cluster, the involved agents cooperatively optimize a separable composite function with a common decision variable. Meanwhile, a global cost function of the whole network is considered associated with an affine coupling constraint across the clusters. To solve this problem, we propose a cluster-based dual proximal gradient algorithm by resorting to the dual problem, where the global cost function is optimized when the agents in each cluster achieve an agreement on the optimal strategy and the global coupling constraint is satisfied. In addition, the proposed algorithm allows the agents to only communicate with their immediate neighbors with heterogeneous step-sizes. The computational complexity of the proposed algorithm with simple-structured cost functions is discussed and an ergodic convergence rate $O(\frac{1}{T})$ is guaranteed ($T$ is the index of iterations).

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

I. INTRODUCTION

A. Background and Motivation

Distributed optimization problems (DOPs) have drawn much attention in the recent few years due to their wide applications in practical problems, such as task assignment in multi-robot networks [1], machine learning problems [2], and economic dispatch in power systems [3]. In those problems, each agent usually maintains an individual decision variable, and the optimal solution of the whole network is achieved through multiple rounds of local communication and decision making.

To adapt to the arising computational burden and privacy preservation issues in the practical implementations with large-scale data sets, DOPs with multiple agent clusters were discussed recently [4]. Generally speaking, a multi-cluster network corresponds to the conventional multi-agent network where each node is broken into a cluster of sub-nodes such that the overall computation task for the node can be separated effectively [5]. In a multi-cluster-based DOP, to achieve the respective objective of each cluster and the whole network, the state updates and information exchanges are usually analyzed at two levels: cluster level and system level, which brings more complications into the algorithm development than the single-level counterparts. In addition, existing research works in this field expose a significant gap in considering coupling constraints across the clusters, which limits their usages in various coupling constrained problems [6–8]. Therefore, in this work, we aim to develop a distributed optimization algorithm for multi-cluster networks with general affine coupling constraints.

B. Literature Review

Existing research works on DOPs usually assume that the communication graph of multi-agent networks is incomplete due to the inadequate communication infrastructures or communication failures among the agents. To tackle this issue, a valid alternative is applying graph theory in modeling the communication links, by which each agent only needs to access the state information from its immediate neighbors [9–11]. In addition, to deal with the coupling constraints, consensus protocol was applied in [12–14], where certain agreement on the state of the network can be achieved only through local communications. To further minimize the requirement on the acquisition of the global information, some more distributed algorithms were proposed by [15–16], where the step-size of the agents can be determined by local information. Some DOPs dealing with time-varying networks were discussed in [17, 18]. Some distributed algorithms with accelerated convergence rates can be referred to in [19, 20].

However, when we extend the conventional multi-agent networks to the multi-cluster cases, the algorithms in the aforementioned works fail to illuminate the realization mechanisms within and across the clusters. Regarding this, an unconstrained DOP with multiple clusters was investigated in [4], where the agreement of the agents in each cluster is achieved by the consensus protocol. Then, the optimal solution of the whole network is achieved when the leader of a cluster executes update in a sequential order by communicating with the leader of neighbor clusters. With a similar network setting, the authors of [21] further proposed a gossip-based communication mechanism among the leaders with a supermartingale convergence guarantee. Then, a random-sleep updating strategy was proposed in [5], where the followers in each cluster can be in an active or inactive mode based on Bernoulli distribution.

In contrast with the existing works with similar problem setups, the new features of this work are summarized as follows.

• We newly consider a DOP in a multi-cluster network with affine coupling constraints across the clusters, which has a significant gap in both algorithm development and application potential from the cluster-based unconstrained DOPs studied in [4, 5, 21]. In addition, the proposed multi-cluster network model is more distributed than...
those in [4, 5, 21] in the sense that we do not set any leader agent who is designed to access the information of other leaders directly.

- Our considered cluster-based DOPs involve two levels of optimization objectives: intra-cluster consensus and extra-cluster optimization. Specifically, the agreement of each cluster is achieved by the consensus protocol among the agents involved. Then, the optimal solution of the whole network is achieved with the presence of the coupling constraint. Therefore, it can be technically challenging to directly apply the existing algorithms without clusters, as studied in [12–18, 22, 23], due to the heterogeneity of the agents employed by this work. Section III formulates the optimization problem. Section II presents some frequently used definitions and properties of proximal mapping and Fenchel conjugate, which are not considered in [24, 27]. In addition, the proposed CDPG algorithm can avoid the bias error of a near-optimal convergence in [16] with the same convergence rate guarantee.

The overall contributions of this work are summarized as follows.

- We consider a class of DOPs in multi-cluster networks with composite cost functions and general affine coupling constraints. In this problem, the decision of each cluster is based on the consensus of the agents involved. The optimal solution of the whole network is achieved when all the agents only communicate with their immediate neighbors.

- A CDPG algorithm is proposed by resorting to the dual algorithm, and the convergence analysis. The feasibility of the algorithm, if the cost functions are with some simple structures (can be non-smooth), one only need to update the dual variables by some simple operations without any costly inner-loop optimizations, which leads to the reduction of the overall computational complexity. This benefit is closely based on the dual proximal gradient method, as also studied in [16, 24–27]. But differently, we focus on solving DOPs with general affine coupling constraints, which are not considered in [24, 27]. In addition, the proposed CDPG algorithm can avoid the bias error of a near-optimal convergence in [16] with the same convergence rate guarantee.

II. PRELIMINARIES

Some frequently used notations, definitions, and relevant properties of 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 cardinality of set \( A \). \( \mathbb{R}^n \) denotes the \( n \)-dimensional Euclidian space only with non-negative real elements. Operator \((\cdot)^+\) represents the transpose of a matrix. \( \prod_{n \in B} A_n \) and \( \bigcap_{n \in B} A_n \) denote the Cartesian product and intersection of sets \( A_1 \) to \( A_{|B|} \), respectively. \( \text{relint} A \) represents the relative interior of set \( A \). \( \| \cdot \|_1 \) and \( \| \cdot \| \) refer to the \( l_1 \)- and \( l_2 \)-norms, respectively. Define \( \| u \|^2_X = u^\top X u \) with \( X \) a square matrix. \( \otimes \) is Kronecker product. \( \tau_{\max}(A) \) denotes the largest eigenvalue of a positive semi-definite matrix \( A \). \( I_n \) is an \( n \)-dimensional identity matrix and \( O_{n \times m} \) is an \((n \times m)\)-dimensional zero matrix. \( I_n \) and \( 0_n \) denote the \( n \)-dimensional column vectors with all entries being 1 and 0, respectively. Let \( \text{diag}(X_n)_A \) be a diagonal-like matrix with \( X_n \) placed on the diagonal according to index \( n = 1, 2, ..., |A| \), i.e., \( \text{diag}(X_n)_A = \begin{bmatrix} X_1 & O & \cdots & \cdots & O \\ O & X_{|A|} \end{bmatrix} \).

B. Proximal Mapping

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

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

A generalized version of proximal mapping can be defined as

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

with \( X \in \mathbb{R}^{n \times n} \) a positive definite matrix [24].

C. Fenchel Conjugate

\( \psi : \mathbb{R}^n \rightarrow (-\infty, +\infty] \) is a proper function. Then, the Fenchel conjugate of \( \psi \) is defined by \( \psi^\circ(v) = \sup_u \{ v^\top u - \psi(u) \} \), which is convex [28, Sec. 3.3].

Lemma 1. (Extended Moreau Decomposition [29, Thm. 6.45])

\( \psi : \mathbb{R}^n \rightarrow (-\infty, +\infty] \) is a proper, convex, and closed function. \( \psi^\circ \) is its Fenchel conjugate. Then for some \( v \in \mathbb{R}^n \) and \( \alpha > 0 \), we have

\[
v = \alpha \text{prox}_\psi^\frac{1}{\alpha} [v] + \text{prox}_\psi^\circ [v]. \quad (2)
\]

Lemma 2. [24, 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,

\[
\arg \max_u \{ v^\top u - \psi(u) \} = \nabla \psi^\circ(v) \quad (3)
\]

and \( \nabla \psi^\circ(v) \) is Lipschitz continuous with Lipschitz constant \( \frac{1}{\sigma} \).

III. PROBLEM FORMULATION

The considered network model, problem formulation, and relevant assumptions are introduced in this section.
A. Network Model

Consider a multi-agent network $G$, which is composed of cluster set $V = \{1,\ldots,N\}$. Cluster $i \in V$ is defined by $G_i = \{V_i, E_i\}$ with agent set $V_i = \{1,\ldots,n_i\}$ and undirected edge set $E_i = \{e_1, e_2, \ldots, e_{|E_i|}\} \subseteq \{(j, l) | j \in V_i, l \in V_i, j \neq l\}$ (no self-loop). Then $G$ can be described by agent set $\hat{V} = \{1,\ldots, \sum_{i=1}^{N} n_i\}$ and undirected edge set $E = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_{|E|}\} \subseteq \{(i, j) | i \in \hat{V}, j \in \hat{V}, i \neq j\}$ (no self-loop). In $\hat{V}$, the index of the $j$th agent in cluster $i$ is relabeled by $n_{ij} = \sum_{l=0}^{i-1} n_l + j$, i.e., the agents are relabeled from cluster 1 to $N$ according to the index in each respective cluster. Let $V_i^j = \{l|(j, l) \in E_i\}$ and $V^j = \{l|(k, l) \in E\}$ be the neighborhood sets of the $j$th agent in $G_i$ and the $j$th agent in $G$, respectively. Let $L_i^j \in \mathbb{R}^{n_i \times n_i}$ be the Laplacian matrix of $G_i$, where the $(j, l)$th entry is defined by

$$ [L_i^j]_{jl} = \begin{cases} |V_i^j| & \text{if } j = l \in V_i \\ -1 & \text{if } (j, l) \in E_i \\ 0 & \text{otherwise} \end{cases} \; (4) $$

With a given indexing protocol of the vertices in $V_i$, the index of edges can be decided as follows. For any two distinct edges $e_k = (k_1, k_2) \in E_i$ and $e_v = (v_1, v_2) \in E_i$, if $\min\{k_1, k_2\} > \min\{v_1, v_2\}$, then $k > v$, and vice versa. For the case $\min\{k_1, k_2\} = \min\{v_1, v_2\}$, if $\max\{k_1, k_2\} > \max\{v_1, v_2\}$, then $k > v$, and vice versa.

**Example 1.** Based on the vertex indexing protocol in Fig. 7, the edge indices are given by $e_1 = (1, 2)$, $e_2 = (2, 4)$, $e_3 = (2, 5)$, $e_4 = (3, 4)$, and $e_5 = (4, 5)$.

![Network Example](image.png)

The graph $G_i$ can also be described by an incidence matrix $G_i \in \mathbb{R}^{n_i \times |E_i|}$, where the rows and columns are indexed by vertices and edges, respectively [31]. Based on the above indexing protocol of edges, the incidence matrix of $G_i$ is defined by

$$ [G_i]_{jk} = \begin{cases} 1 & \text{if } e_k = (j, l) \in E_i \text{ and } j > l \\ -1 & \text{if } e_k = (j, l) \in E_i \text{ and } j < l \\ 0 & \text{otherwise} \end{cases} \; (5) $$

Similarly, we define an incidence matrix $G \in \mathbb{R}^{|\hat{V}| \times |E|}$ for $G$, where

$$ [G]_{jk} = \begin{cases} 1 & \text{if } \hat{e}_k = (j, l) \in E \text{ and } j > l \\ -1 & \text{if } \hat{e}_k = (j, l) \in E \text{ and } j < l \\ 0 & \text{otherwise} \end{cases} \; (6) $$

B. The Optimization Problem

Let $H_i(x_i) = \sum_{i \in V} H_i(x_i)$ be the global cost function of network $G$ and $H_i(x_i) = \sum_{j \in V_i} (f_{ij}(x_i) + g_{ij}(x_i))$ be the cost function of cluster $i$, where $f_{ij} + g_{ij}$ is the cost function of the $j$th agent in cluster $i$, $x_i \in \mathbb{R}^M$, $x = [x_1, \ldots, x_N]^T \in \mathbb{R}^{NM}$. The optimization problem of the whole network is formulated as

$$ (P1) \quad \min_x \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(x_i) + g_{ij}(x_i)) $$

subject to $Ax \leq b$, \; (7)

with $A \in \mathbb{R}^{B \times NM}$, $b \in \mathbb{R}^B$. An illustrative communication topology of the network is shown in Fig. 2.

**Remark 1.** The composite cost function $f_{ij} + g_{ij}$ is a generalization of many cost functions in practical problems, such as resource allocation problems [25], regularization problems [22], and support vector machines [33]. For instance, $g_{ij}$ can be a regularization penalty term, such as $l_1$- or $l_2$-norms. Alternatively, we can find a local feasible region $X_{ij} \subseteq \mathbb{R}^M$ for the $j$th agent in cluster $i$, where $X_{ij}$ is non-empty, convex, and closed. Then, we can let $g_{ij}$ be an indicator function $1_{X_{ij}}$, where $1_{X_{ij}}(x_i) = \begin{cases} 0 & \text{if } x_i \in X_{ij} \\ \infty & \text{otherwise} \end{cases}$ [24].

**Remark 2.** For the comparison purpose, we consider a conventional composite DOP

$$ (P1+) \quad \min_x \sum_{i \in V} (f_i(x_i) + g_i(x_i)) $$

subject to $Ax \leq b$, \; (8)

where $f_i + g_i$ is the cost function of agent $i$. Then, compared with Problem (P1+), the new features of the multi-cluster based Problem (P1) are explained as follows.

- Problem (P1) can be viewed as a generalization of Problem (P1+) by expanding the agent who manages $x_i$ into cluster $i$ with distributed agents. In case there is only one agent in each cluster, Problem (P1) is equivalent to Problem (P1+).
- To optimize the global cost function, the agents in each cluster need to generate a consensual decision, e.g., $x_i$, which is involved by the coupling constraint $Ax \leq b$. Therefore, in the distributed setup, the interactions among the agents within each cluster and across different clusters should be considered simultaneously.
Assumption 1. The edges in $G$ and $G_i$ are undirected: Any two distinct vertices in $G$ and $G_i$ are linked by at least one path, $\forall i \in V$.

Assumption 2. Both $f_{ij}: \mathbb{R}^M \to (-\infty, +\infty]$ and $g_{ij}: \mathbb{R}^M \to (-\infty, +\infty]$ are proper, convex, and closed extended real-valued functions; $f_{ij}$ is $\sigma_{ij}$-strongly convex, $\sigma_{ij} > 0$, $\forall i \in V$, $j \in V_i$.

The assumptions in Assumption 2 are often discussed in composite optimization problems [16][24][25][34][37].

Assumption 3. (Constraint Qualification [38]) There exists an $\bar{x} \in \text{relint } D$ such that $A\bar{x} < b$, where $D = \prod_{i \in V} \bigcap_{j \in V_i} D_{ij}$ with $D_{ij}$ the domain of $f_{ij} + g_{ij}$, $\forall i \in V$, $j \in V_i$.

IV. Main Result

In this section, we propose a CDPG algorithm for solving the problem of interest, discuss the computational complexity with simple-structured cost functions, and further conduct the convergence analysis.

A. Dual Problem

To realize distributed computation, we decouple the variable of clusters by defining $y_i \in \mathbb{R}^M$ as the estimate of $x_i$ by the $j$th agent in cluster $i$. Then, the collection of the estimates in cluster $i$ can be $y_i = [y_{i1}^T, ..., y_{in_i}^T]^T \in \mathbb{R}^{n_iM}$ and the collection of the overall estimates can be $y = [y_1^T, ..., y_N^T]^T \in \mathbb{R}^{N \sum_{i \in V} n_i M}$.

Based on Problem (P1), the estimates in each cluster should reach a consensus. Then by the consensus protocol in cluster $i$: $x_i = y_{i1} = ... = y_{in_i}$, the global constraint can be represented by

$$Ax \leq b \iff \sum_{i \in V} A_i x_i \leq b \iff \sum_{i \in V} A_i y_i \leq b \iff \sum_{i \in V} A_i y_i \leq b \iff \sum_{i \in V} A_i y_i \leq b \iff \sum_{i \in V} A_i y_i \leq b,$$

where $A_i = \frac{1}{n_i} \otimes A_i \in \mathbb{R}^{B \times n_i M}$ and $A = [A_1, ..., A_N] \in \mathbb{R}^{B \times \sum_{i \in V} n_i M}$ with $A_i \in \mathbb{R}^{B \times M}$ the $i$th column block of $A$ (i.e., $A_i = [A_{i1}, ..., A_{in_i}]$). Essentially, (9) reconstructs the coupling constraint in (P1) with the augmented variables without affecting the nature of the constraint when certain consensus is achieved in each cluster.

Note that the consensus constraint of $y_{ij}$ in cluster $i$ can be equivalently written as $L_i^T y_i = 0$, where $L_i = L_i^T \otimes \mathbf{1}_M \in \mathbb{R}^{n_i M \times M}$ is the augmented Laplacian matrix of $G_i$. Then the consensus-based optimization problem of the whole network can be formulated as

$$(P2) \quad \min_{y} \quad \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + g_{ij}(z_{ij}))$$

subject to

$$y_{ij} = z_{ij}, \quad \forall i \in V, j \in V_i, \quad (10)$$
$$L_i^T y_i = 0, \quad \forall i \in V, \quad (11)$$
$$Ay \leq b, \quad \forall i \in V, j \in V_i, \quad (12)$$

where $z_{ij} \in \mathbb{R}^M$ is a slack variable.

The Lagrangian function of Problem (P2) can be given by

$$L(y, z, \mu, \nu, \phi) = \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + g_{ij}(z_{ij}) + \mu_{ij}^T (y_{ij} - z_{ij})) + \nu_i^T L_i y_i + \phi^T (Ay - b)$$
$$= \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + g_{ij}(z_{ij}) + \mu_{ij}^T (y_{ij} - z_{ij})) + \nu_i^T L_i y_i + \phi^T (A_i y_i - \nu_i \phi^T b)$$
$$= \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + g_{ij}(z_{ij}) + \mu_{ij}^T (y_{ij} - z_{ij})) + \nu_i^T L_i y_i + \phi^T (A_i y_i - \nu_i \phi^T b), \quad (13)$$

where $\mu_{ij} \in \mathbb{R}^M$ and $\nu_i \in \mathbb{R}^{n_i M}$ and $\phi \in \mathbb{R}^B$ are Lagrangian multipliers,

$$\mathbf{z}_i = [z_{i1}^T, ..., z_{in_i}^T]^T, \quad \mathbf{z} = [\mathbf{z}_1^T, ..., \mathbf{z}_N^T]^T,$$
$$\mathbf{\mu} = [\mathbf{\mu}_1, ..., \mathbf{\mu}_n], \quad \mathbf{\mu} = [\mathbf{\mu}_1, ..., \mathbf{\mu}_N],$$
$$\mathbf{\nu} = [\mathbf{\nu}_1, ..., \mathbf{\nu}_N], \quad \sum_{i \in V} \mathbf{\kappa}_i = 1, \quad \sum_{i \in V} \mathbf{\eta}_i = 1,$$

$A_{ij} \in \mathbb{R}^{B \times M}$ is the $j$th column block of $A_i$ (i.e., $A_i = [A_{i1}, ..., A_{in_i}]$), and $L_i^T \in \mathbb{R}^{n_i M \times n_i M}$ is the $j$th column block of $L_i$ (i.e., $L_i = [L_{i1}, ..., L_{in_i}]$).

Then the dual function can be obtained by

$$W(\mathbf{\mu}, \mathbf{\nu}, \phi) = \min_{y, \mathbf{\mu}, \mathbf{\nu}, \phi} \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + g_{ij}(z_{ij}) + \mu_{ij}^T (y_{ij} - z_{ij})) + \nu_i^T L_i y_i + \phi^T (A_i y_i - \nu_i \phi^T b)$$
$$= \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(y_{ij}) + (\mu_{ij} + \nu_j) y_{ij} - \nu_i \phi^T b)$$
$$- \kappa_i \eta_i \phi^T - \nu_i \phi_{ij}^T$$
$$= \sum_{i \in V} \sum_{j \in V_i} (-f_{ij} (\mu_{ij} - L_i^T \nu_i + A_{ij}^T \phi))$$
$$- \kappa_i \eta_i \phi^T - \nu_i \phi_{ij}^T,$$

where

$$\mathbf{H}_{ij} = [-\mathbf{L}_i^T, -\mathbf{A}_{ij}^T], \quad \mathbf{E}_{ij} = [0^T, \mathbf{0}_{n_i M}, \kappa_i \eta_i \mathbf{b}^T],$$
$$\mathbf{F}_{ij} = [-\mathbf{L}_i^T, \mathbf{O}_{M \times n_i M}, \mathbf{O}_{M \times B}], \quad \phi_{ij} = [\mathbf{\mu}_j, \mathbf{\nu}_i, \phi]^T.$$

Hence, the dual problem of Problem (P2) is

$$(P3) \quad \min_{\phi \geq 0} \sum_{i \in V} \sum_{j \in V_i} (f_{ij}(\mathbf{H}_{ij} \phi_{ij}) + \mathbf{E}_{ij} \phi_{ij} + g_{ij}^T (\mathbf{F}_{ij} \phi_{ij})), \quad (14)$$

where $\phi = [\phi_1^T, ..., \phi_N^T]^T, \phi_{ij} = [\phi_{i1}^T, ..., \phi_{in_i}^T]^T$. Since considering the constraint $\phi \geq 0$ is equivalent to accommodating indicator function $I_{R^B}(\phi) = I_{R^B}(N_{ij} \phi_{ij})$ into the cost function with $N_{ij} = [\mathbf{O}_{B \times (M+n_i M)} I_B]$, then Problem (P3) can be rewritten as

$$(P4) \quad \min_{\phi} \sum_{i \in V} \sum_{j \in V_i} \phi_{ij} \phi_{ij} + \sum_{i \in V} \sum_{j \in V_i} \phi_{ij} \phi_{ij},$$

where
where
\[ p_{ij}(\theta_{ij}) = f_{ij}^T(H_{ij}(\theta_{ij}) + E_{ij}\theta_{ij}), \quad (15) \]
\[ q_{ij}(\theta_{ij}) = g_{ij}^T(F_{ij}\theta_{ij}) + I_{B_{ij}}(N_{ij}\theta_{ij}). \quad (16) \]

Note that \( \theta_{ij} \) is coupled among the agents with the common components \( \nu_i \) and \( \phi \). To separate a separable structure for the cost function in (P4), we define \( \lambda_{ij} = [\mu_{ij}^T, \gamma_{ij}^T, \theta_{ij}^T]^T \), where \( \gamma_{ij} \) and \( \theta_{ij} \) are the local estimates of \( \nu_i \) and \( \phi \) by the \( j \)th agent in cluster \( i \), respectively. In addition, we define
\[ \gamma_i = [\gamma_{i1}^T, \ldots, \gamma_{i_n}^T]^T, \gamma = [\gamma_1^T, \ldots, \gamma_N^T]^T, \]
\[ \theta_i = [\theta_{i1}^T, \ldots, \theta_{in}^T]^T, \theta = [\theta_1^T, \ldots, \theta_N^T]^T, \]
\[ \lambda_i = [\lambda_{i1}^T, \ldots, \lambda_{in}^T]^T, \lambda = [\lambda_1^T, \ldots, \lambda_N^T]^T, \]
\[ M_{ij} = [O_{n, M \times M}, I_{n, M}, O_{n, M \times B}], \]
\[ M_i = I_{n_i} \otimes M_{ij}, M = \text{diag}[M_i], \]
\[ N_i = I_{n_i} \otimes N_{ij}, N = \text{diag}[N_i]. \]

Then we have \( \gamma_{ij} = M_{ij} \lambda_i, \theta_{ij} = N_{ij} \lambda_i, \gamma_i = M_i \lambda_i, \gamma = M \lambda, \theta_i = N_i \lambda_i, \) and \( \theta = N \lambda \). For convenience purposes, we relabel \( \theta_{ij} \) by defining \( \theta_{n_{ij}} = \theta_{ij} \). Then the consensus of \( \gamma_{ij} \) in \( V_i \) and \( \theta_{n_{ij}} \) in \( V \) can be characterized by
\[ \gamma_{il} - \gamma_{ij} = 0, \quad \forall l \in S_i, \quad (17) \]
\[ \theta_{k} - \theta_{n_{ij}} = 0, \quad \forall k \in S_{ij}, \quad (18) \]
respectively, where \( S_i = \{ l| (j, l) \in E_{ij}, l > j \} \), \( S_{ij} = \{ k| (n_j, k) \in E_k, k > n_j \} \), \( \forall i \in V, j \in V_i \).

**Example 2.** Assume that the cluster setting in Example 7 follows \( \tilde{V} = \{ 1, 2, 3, 4, 5 \}, V_1 = \{ 1 \}, V_2 = \{ 2 \}, \)
\( V_3 = \{ 1, 2 \} \). Then \( S_{i1} = \emptyset, S_{i2} = \emptyset, S_1 = S_{i1} = \{ 2 \}, \) \( S_2 = S_{i2} = \emptyset, S_{i1} = \{ 1, 2 \} \), \( S_1 = S_{i1} = \{ 1 \} \), \( S_2 = S_{i2} = \emptyset \). An empty set of \( S_{ij} \) or \( S_{ij} \) means no consensus constraint exists in (17) or (18), respectively.

Note that (17) and (18) can be written in compact forms with the help of the incidence matrix [31]. By defining \( G_i = G_i^T \otimes I_{n_{M_i}} \), (17) can be represented by \( G_i \gamma_i = G_i M_i \lambda_i = 0 \). In addition, one can construct \( G \gamma = G M \lambda = 0 \) by including all clusters with \( G = \text{diag}[G_i] \). Similarly, (18) can be represented by \( G \theta = G N \lambda = 0 \), where \( \theta = [\theta_1^T, \ldots, \theta_N^T] \) and \( G = G^T \otimes I_B \). Then, by defining \( Z = [M^T G^T, N^T G^T]^T \), (17) and (18) can be jointly represented by \( Z \lambda = 0 \).

Consider a quadratic term
\[ ||Z\lambda||_{D[\pi]}^2 = \sum_{i \in V} \sum_{j \in V_i} \pi_{ij} \sum_{l \in S_{ij}} (\gamma_{il} - \gamma_{ij})^2 + \sum_{i \in V} \sum_{j \in V_i} \pi_{ij} \sum_{k \in S_{ij}} (\theta_k - \theta_{n_{ij}})^2, \]
where
\[ D[\pi] = \begin{bmatrix} \text{diag}[V_i] & 0 \\ 0 & \text{diag}[V_i] \end{bmatrix}, \]
\[ V_i = \begin{bmatrix} \pi_{i1} I_{S_{i1} n_{M}} & \cdots \\ \pi_{i2} I_{S_{i2} n_{M}} & \cdots \\ \vdots & \vdots \\ \pi_{in} I_{S_{in} n_{M}} & \cdots \end{bmatrix}, \]
with \( \pi_{ij} > 0 \) a weighting factor, \( i \in V, j \in V_i \).

Then, a consensus-based penalized dual problem of Problem (P2) can be given by
\[ \min_\lambda \Phi(\lambda) = P(\lambda) + Q(\lambda), \quad \text{subject to } Z \lambda = 0, \quad (20) \]
\[ P(\lambda) = \sum_{i \in V} \sum_{j \in V_i} p_{ij} (\lambda_{ij}), \quad (21) \]
\[ Q(\lambda) = \sum_{i \in V} \sum_{j \in V_i} q_{ij} (\lambda_{ij}), \quad (22) \]
\[ p_{ij} (\lambda_{ij}) = f_{ij}^T (H_{ij}(\lambda_{ij}) + E_{ij} \lambda_{ij}), \quad (23) \]
\[ q_{ij} (\lambda_{ij}) = g_{ij}^T (F_{ij} \lambda_{ij}) + I_{B_{ij}} (N_{ij} \lambda_{ij}). \quad (24) \]

\( \Phi \) is convex based on the definition of Fenchel conjugate.

**B. Distributed Optimization Algorithm Development**

The Lagrangian function of Problem (P5) can be given by
\[ \mathcal{L}(\lambda, \omega) = P(\lambda) + Q(\lambda) + \frac{1}{2} ||Z\lambda||_{D[\pi]}^2 + \omega^T Z \lambda = \sum_{i \in V} \sum_{j \in V_i} p_{ij} (\lambda_{ij}) + \sum_{i \in V} \sum_{j \in V_i} q_{ij} (\lambda_{ij}), \quad (25) \]
\[ + \frac{1}{2} \sum_{i \in V} \sum_{j \in V_i} \pi_{ij} \sum_{l \in S_{ij}} (\gamma_{il} - \gamma_{ij})^2 + \frac{1}{2} \sum_{i \in V} \sum_{j \in V_i} \sum_{k \in S_{ij}} (\theta_k - \theta_{n_{ij}})^2, \]
\[ + \sum_{i \in V} \sum_{j \in V_i} \sum_{k \in S_{ij}} \xi_{ijl} (\gamma_{il} - \gamma_{ij}) + \sum_{i \in V} \sum_{j \in V_i} \sum_{k \in S_{ij}} \xi_{ijk} (\theta_k - \theta_{n_{ij}}), \]
where
\[ \xi_{ijl} \in \mathbb{R}^{n_{M}}, \xi_{ijl} = [\xi_{ijl,1}, \ldots, \xi_{ijl,n_{ij}}]^T, l_{(\cdot)} \in S_{ij}, \]
\[ \xi_{ijk} \in \mathbb{R}^{n_{B}}, \xi_{ijk} = [\xi_{ijk,1}, \ldots, \xi_{ijk,n_{ij}}]^T, k_{(\cdot)} \in S_{ij}, \]
\[ \xi_i = [\xi_{i1}, \ldots, \xi_{in_i}]^T, \xi = [\xi_1^T, \ldots, \xi_N^T]^T, \]
\[ \xi_i = [\xi_{i1}, \ldots, \xi_{in_i}]^T, \xi = [\xi_1^T, \ldots, \xi_N^T]^T, \omega = [\gamma_i^T, \theta_i^T]^T. \]
Here, \( \omega \) is the collection of the Lagrangian multipliers. Sequence \{1, \ldots, \{ S_{ij} \} \} is decided by: \( \forall m, l \in S_{ij}, l_m > l_i \), then \( m > n \), and vice versa. Similarly, in \{1, \ldots, \{ S_{ij} \} \}, \( \forall k_m, k_i \in S_{ij}, k_m > k_i \), then \( m > n \), and vice versa.

**Example 3.** Based on the graph in Example 1 and the cluster setting in Example 2, we have \( \xi_1 = [\xi_{12}, \xi_{13}, \xi_{14}, \xi_{15}]^T, \xi = [\xi_{12}, \xi_{13}, \xi_{14}, \xi_{15}]^T, \]
\( \xi_3 = \xi_{314}, \xi_{315} \), and \( \xi_4 = \xi_{415} \).
Let $C$ be the set of the saddle points of $\mathcal{L}(\lambda, \omega)$. Then, any saddle point $(\lambda^*, \omega^*) \in C$ satisfies:

$$\mathcal{L}(\lambda, \omega^*) \geq \mathcal{L}(\lambda^*, \omega^*) \geq \mathcal{L}(\lambda^*, \omega).$$

We aim to seek a saddle point of $\mathcal{L}(\lambda, \omega)$, which can be characterized by Karush-Kuhn-Tucker (KKT) conditions:

$$0 \in \nabla_\lambda P(\lambda^*) + \partial_\lambda Q(\lambda^*) + Z^T \omega^* + Z^T D[\pi]Z \lambda^*, \quad \text{and} \quad Z \lambda^* = 0.$$ 

(28)

Then the proposed CDPG algorithm for solving Problem (P5) is designed as:

$$\lambda^{t+1} = \text{prox}_{\Omega}[\lambda^t - S[c](\nabla_\lambda P(\lambda^t) + Z^T \omega^t + Z^T D[\pi]Z \lambda^t)],$$

$$\omega^{t+1} = \omega^t + D[\pi]Z \lambda^{t+1},$$

(29)

(30)

where

$$S[c] = \text{diag}(C_i)_V,$$

$$C_i = \begin{bmatrix}
    c_{i1} & 0 & \ldots & 0 \\
    0 & c_{i2} & \ldots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \ldots & c_{in_i}
\end{bmatrix} \otimes I_{M+n_i+M_B}.$$ 

(32)

Based on (24), we have:

$$q_{ij}(\lambda_{ij}) = g_{ij}^c(\mu_{ij}) + I_{R^B}(\theta_{ij}) = g_{ij}^c(\mu_{ij}) + I_{R^B}(\gamma_{ij}) + I_{R^B}(\theta_{ij}).$$

(33)

Then one can have [28 Thm. 6.6.]

$$\text{prox}_{q_{ij}} = \text{prox}_{g_{ij}^c} \times \text{prox}_{\gamma_{ij}^c} \times \text{prox}_{\theta_{ij}^c}.$$ 

(34)

Based on (34), (29) and (30) can be decomposed into:

$$\nu_{ij}^{t+1} = \text{prox}_{\mu_{ij}}[\mu_{ij}^{t+1} - c_{ij} \nabla_{\mu_{ij}} p_{ij}(\lambda_{ij}^t)],$$

$$\gamma_{ij}^{t+1} = \gamma_{ij} - c_{ij} \nabla_{\gamma_{ij}} p_{ij}(\lambda_{ij}^t) - \sum_{k \in S_{ij}} \xi_{ik}^t,$$

$$\xi_{lk}^{t+1} = \xi_{lk}^t + \pi_{lk}^t \sum_{k \in S_{ij}} (\theta_{lk}^t - \theta_{lk}) + \sum_{l \in S_{ij}} (\gamma_{lk}^t - \gamma_{lk}),$$

$$\theta_{ij}^{t+1} = \text{prox}_{\theta_{ij}}[\theta_{ij}^{t+1} - \frac{\pi_{ij}^t}{\pi_{ij}} \theta_{ij}],$$

$$\nu_{ij}^{t+1} = \nu_{ij}^t + \sum_{k \in S_{ij}} (\theta_{ij}^{k+1} - \theta_{ij}^k) + \pi_{ij}^t \sum_{k \in S_{ij}} (\theta_{ij}^k - \theta_{ij}^k),$$

(35)

(36)

(37)

(38)

(39)

due to the separability of $P$ and $Q$, where $\pi$ is the relabeled $\pi$ by formula $\pi_{n_{ij}} = \pi_{ij}, S_{ij}^t = \{l | (i, l) \in E_i, l < j\},$ $\bar{S}_{ij}^t = \{k | (m_{ij}, k) \in E_i, k < n_{ij}\},$ and $P_{R^B}[\cdot]$ is a Euclidean projection onto $R^B$, $\forall i \in V_j, \forall l \in V_i$. (36) and (37) hold since the proximal mapping of an indicator function is equivalent to a Euclidean projection (the projection onto $R^{n_i+M_i}$ in (36) is omitted) [41 Sec. 1.2].

**Remark 3.** To apply (32) to (27), one needs to compute $\nabla_{\mu_{ij}} p_{ij}(\lambda_{ij}), \nabla_{\gamma_{ij}} p_{ij}(\lambda_{ij}),$ and $\nabla_{\theta_{ij}} p_{ij}(\lambda_{ij})$. By Lemma 2 we have:

$$\nabla_{\lambda_{ij}} p_{ij}(\lambda_{ij}) = \nabla_{\mu_{ij}} p_{ij}(\lambda_{ij}) + \nabla_{\gamma_{ij}} p_{ij}(\lambda_{ij}) + \nabla_{\theta_{ij}} p_{ij}(\lambda_{ij}) = \mathbf{H}_{ij}^\top \mathbf{H}_{ij} \lambda_{ij}^t + \mathbf{E}_{ij}^\top,$$

(39)

which can be completed by some numerical methods only with local information. On the other hand, if the proximal mapping of $g_{ij}$ is easier to obtain, the computation of (36) can be further simplified by employing Lemma 7, which gives:

$$\theta_{ij}^{t+1} = \frac{1}{\pi_{ij}} \boldsymbol{\theta}_{ij} - c_{ij} \nabla_{\theta_{ij}} p_{ij}(\lambda_{ij}^t) - \pi_{ij} \sum_{k \in S_{ij}} (\theta_{ij}^k - \theta_{ij}^k).$$

(40)

**Remark 4.** The considered inequality-constrained DOP can cover the equality-constrained scenarios by removing the constraint $\phi \geq 0$ in Problem (P3) based on KKT conditions, which leads to $q_{ij}(\lambda_{ij}) = g_{ij}^c(\mu_{ij})$ in Problem (P5) by removing the indicator function $I_{R^B}(\mu_{ij})$ from the cost function. Then, the updating law (36) becomes:

$$\theta_{ij}^{t+1} = \theta_{ij}^t - c_{ij} \nabla_{\theta_{ij}} p_{ij}(\lambda_{ij}^t) - \sum_{k \in S_{ij}} \xi_{ik}^t + \sum_{k' \in S_{ij}} \xi_{ik'}^t,$$

(41)

by removing the Euclidean projection.

In the following, we discuss how to recover the optimal primal solution $y^*$. By the saddle point property:

$$L(y^*, z^*, \mu, \nu, \phi) \leq L(y^*, z^*, \mu^*, \nu^*, \phi^*) \leq L(y, z, \mu^*, \nu^*, \phi^*),$$

(42)

the optimal primal variable can be obtained by the second inequality in (43): $y^* = \arg \min_y L(y, z, \mu^*, \nu^*, \phi^*)$. Then by decomposing $y$ and omitting the constant terms in $L$, we have:

$$y_{ij}^* = \arg \min_{y_{ij}} f_{ij}(y_{ij}) + (\mu_{ij}^* + \mathbf{L}_{ij}^\top \nu^* + \mathbf{A}_{ij} \phi^*)^\top y_{ij}$$

(43)

$$= \arg \min_{y_{ij}} f_{ij}(y_{ij}) + (\mu_{ij}^* + \mathbf{L}_{ij}^\top \gamma_{ij}^* + \mathbf{A}_{ij} \theta_{ij}^*)^\top y_{ij},$$

(44)

where we use $\gamma_{ij}^* = \nu^*_i$ and $\theta_{ij}^* = \phi^*$ since $\gamma_{ij}^*$ and $\theta_{ij}^*$ are the optimal local estimates of $\nu^*_i$ and $\phi^*$, respectively, $i \in V_j, j \in V_i$. The detailed computation procedure of the CDPG algorithm is summarized in Algorithm [1].
Algorithm 1 CDPG algorithm

1: Initialize $X^0$, $\omega^0$. Determine step-sizes $c_{ij}, \pi_{ij} > 0$, $\forall i \in V, j \in V_j$.
2: for $t = 0, 1, 2, \ldots$ do
3: for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, n_i$ do (in parallel)
4: Update $\nabla_{\omega_{ij}} P_{ij}(\lambda_{ij}^t)$, $\nabla_{\lambda_{ij}} P_{ij}(\lambda_{ij}^t)$, and $\nabla_{\theta_{ij}} P_{ij}(\lambda_{ij}^t)$ based on (43).
5: Update $\mu_{ij}^{t+1}$ based on (35) or (41)-(42).
6: Intra-cluster interaction:
7: Update $\gamma_{ij}^{t+1}$ and $\xi_{ij}^{t+1}$ based on (36) and (38), respectively.
8: Extra-cluster interaction:
9: Update $\theta_{ij}^{t+1}$ and $\zeta_{ij}^{t+1}$ based on (37) and (39), respectively.
10: end for
11: end for
12: Obtain the output of dual variables $\mu_{ij}, \gamma_{ij}, \xi_{ij}, \theta_{ij},$ and $\zeta_{ij}, \forall i \in V, j \in V_j$. Calculate the primal solution based on (44).

C. Convergence Analysis

Lemma 3. With Assumption 2, the Lipschitz constant of $\nabla_{\lambda_{ij}} P_{ij}(\lambda_{ij})$ is given by $h_{ij} = \frac{\|H_i^\perp\|}{\sigma_{ij}}$, $\forall i \in V, j \in V_j$.

See the proof in Appendix A.

In the following, we let $D[\pi]$ and $S[c]$ be the inverse matrices of $D[\pi]$ and $S[c]$, respectively. In addition, define

$$S[h] = \text{diag}[W_i]_{ij},$$

$$W_i = \begin{bmatrix} h_{i1} & & \cdots & \text{O} \\ & \ddots & \cdots & \text{O} \\ & & h_{i_{n_i}} & \end{bmatrix} \otimes I_{M+n_iM+M_B}.$$ (45)

Theorem 1. Suppose that Assumptions 2,3 hold. Let $0 < c_{ij} \leq \frac{1}{h_{ij} + H_{\max}(Z \cdot D[\pi]Z^T)}$, $\forall i \in V, j \in V_j$. By Algorithm 1, for any $(\lambda^*, \omega^*) \in C$, we have

$$|\Phi(\lambda^{T+1}) - \Phi(\lambda^*)| \leq \frac{\Theta}{T+1},$$

$$\|\omega^*\|Z\lambda^{T+1}\| \leq \frac{\Theta}{T+1},$$

where $\Theta = \|\omega^*\|Z^T D[\pi]|Z| + \|\omega^0\|Z^T D[\pi]|Z| + \lambda^* - \lambda^0 \|Z^T D[\pi]|Z|^{T+1} = \frac{1}{T+1} \sum_{t=0}^T \lambda^{t+1}, T \in \mathbb{N}_+.$

See the proof in Appendix B.

D. Computational Complexity with Simple-Structured Cost Functions

By the proposed CDPG algorithm, (35) to (37) may require some inner-loop optimizations to compute $\nabla P_{ij}$ as discussed in Remark 3 and the proximal mapping of $g_{ij}^\ast$. In the following, we will discuss the CDPG algorithm with some simple-structured cost functions, where the computational complexity can be reduced. For the gradient-based iterative algorithms, the computational complexity is dominated by iteration complexity and computational cost per iteration [42].

First, we consider the iteration complexity of (35) in the following cases.

- If $f_{ij}$ is a simple-structured smooth function (e.g., $f_{ij}$ is quadratic), $\nabla f_{ij}$ (also $\nabla P_{ij}$ by (35)) can be obtained efficiently [38, Sec. 3.3.1]. Widely discussed optimization problems with smooth + nonsmooth cost functions can be referred to in [25,32,33].

- In (35), the proximal mapping of $g_{ij}^\ast$ can be efficiently obtained if $g_{ij}$ is simple-structured. For example, consider a regularization problem, where the penalty is a Euclidean $\epsilon$-norm: $g_{ij}(x_{ij}) = \|x_{ij}\|_\epsilon$ (e.g., it can be an LASSO problem if $\epsilon = 1$ [32]). Then, we can have

$$g_{ij}^\ast(\mu_{ij}) = I_{V_j}(\mu_{ij}) = \begin{cases} 0 & \text{if } \mu_{ij} \in Y_{ij}, \\ +\infty & \text{otherwise} \end{cases},$$ (49)

where $Y_{ij} = \{v \in \mathbb{R}^M | \|v\|_\epsilon \leq 1 \}$ with $\|\cdot\|_\epsilon$ being the dual norm of $\|\cdot\|_\epsilon$. The first equality holds by computing the conjugate of an $\epsilon$-norm [38, Sec. 3.3.1]. Then in (35), the proximal mapping of $g_{ij}^\ast$ is a Euclidean projection onto $Y_{ij}$ [41, Sec. 1.2].

Based on the above discussion, provided that $f_{ij}$ and $g_{ij}$ are some simple-structured functions, (35) only requires some simple operations (e.g., addition, multiplication, and Euclidean projection) with iteration complexity $O(1)$ without any costly inner-loop optimization. Then the overall iteration complexity of (35) can be $O\left(\frac{1}{T}\right)$ with $T$ being the convergence error (see Thm. 1). Note that for gradient-based iterative algorithms, the computational cost per iteration is linear in dimension [42]. Then, the overall computational complexity of (35) can be $O\left(\frac{1}{T}\right)$.

Meanwhile, given that $f_{ij}$ is simple-structured, (36)-(39) can also be computed efficiently only with some simple operations with iteration complexity $O(1)$. Then the overall computational complexity of the CDPG algorithm can be obtained by counting the overall dimension of (35)-(39), which gives $O\left(\frac{M|V|}{\epsilon} + O\left(\sum_{i \in V} V_i^2 M^2\right) + O\left(\frac{B|V|}{\epsilon}\right) + O\left(\sum_{i \in V} \sum_{r \in e} |S_{ij}| n_i M\right) \epsilon^{-1} + O\left(\sum_{i \in V} \sum_{r \in e} |S_{ij}| B\right)\right)$.

V. Motivating Applications and Numerical Simulation

A. Social Welfare Optimization in Commodity Market

In this simulation, we consider a social welfare optimization problem in a commodity market [43]. In this market, we aim to supply certain amount of commodities to multiple consumer regions such that the utility function of the whole consumer community is optimized (the transportation cost is assumed to be negligible).

The utility function of different regions can be obtained by some learning machines based on the regional information [44]. Due to the possibly large-scale data sets and privacy perseveration issues, it can be inefficient or even infeasible to transmit the whole data sets among the machines. Therefore, distributed learning framework can be employed [45], as introduced as follows.

- Distributed learning machines are established, who collect the data in different areas and generate the utility functions based on the local data and learning algorithms.
• The utility function of each region can be settled by ensemble method [46], e.g., take the average of all the generated functions in each region.

Then, the machines decide the optimal commodity supply strategy in a distributed manner to optimize the utility function of the whole community.

![Diagram](image)

Fig. 3. Communication typology of learning machines.

Based on the above discussion, we let \( f_{ij}(x_i) \) be the utility function generated by the \( j \)-th machine in region \( i \). Then, the utility function of region \( i \) is settled as \( f_i(x_i) = \frac{1}{n_i} \sum_{j \in V_i} f_{ij}(x_i) \) [47]. Hence, the social welfare optimization problem can be formulated as

\[
\text{(P6): } \max_{x \in X} H(x) \\
\text{subject to } Ax \leq b.
\]

Here, \( H(x) = \sum_{i \in V} f_i(x_i) \), \( x = [x_1, \ldots, x_{|V|}]^T \), \( A = 1_{|V|}^T \), \( b \) is total quantity of commodities in store. \( X = \prod_{i \in V} \prod_{j \in V_i} X_{ij} \) with \( X_{ij} = [\underline{x}_{ij}, \bar{x}_{ij}] \), where \( \underline{x}_{ij} \) and \( \bar{x}_{ij} \) are the lower and upper bounds of demand, respectively. In addition, the utility function obtained by the machines is assumed to be in a quadratic form \( f_{ij}(x_i) = \omega_{ij} x_i^2 + \varsigma_{ij} x_i + \zeta_{ij} \) [43]. \( b \) is set as 5. The detailed communication typology of the machines and other parameter settings are shown in Fig. 3 and Table I, respectively.

By some direct calculations, the optimal solution to Problem (P6) is \( x^* = [3.33, 0.1, 1.67]^T \) (the lower bound of \( x_2 \) is activated). To characterize the relative convergence error, we define \( \delta = \left| \frac{L(\omega, \omega^*) - H(x^*)}{H(x^*)} \right| \) with certain non-zero \( H(x^*) \). The simulation result is shown in Fig. 4. Fig. 4(a) shows that the iteration tends to the steady state asymptotically. Fig. 4(b) depicts the trajectory of \( \omega \). The trajectory of convergence error is shown in Fig. 4(c).

### B. Economic Emission Dispatch Optimization in Energy Market

In this simulation, we consider an economic emission dispatch problem in an energy market. In this market, we consider multiple energy generation companies (GENCOs) and regulation entities (REs), where the latter ones can be some policy makers for the environment’s benefit and are responsible for regulating the pollutant emissions [48]. Specifically, two sorts of emissions are considered: sulfur dioxide (SO\(_2\)) and oxides of nitrogen (NO\(_x\)), which are regulated by possibly different REs. Certain GENCO and its REs are assumed to jointly make decisions to optimize the output of the GENCO. Hence, we treat the GENCO and its REs as a cluster.

In this problem, the objective function of cluster \( i \) is designed as \( f_i(x_i) = \chi_i C_i(x_i) + (1 - \chi_i) \delta_i (E_i^S(x_i) + E_i^N(x_i)) \) [49]. Here, \( x_i \in X_i = [\underline{x}_i, \bar{x}_i] \) is the quantity of energy generation with \( \underline{x}_i \) and \( \bar{x}_i \) being the lower and upper generation limits, respectively. \( C_i, E_i^S \), and \( E_i^N \) are the fuel cost, emission quantity of SO\(_2\), and emission quantity of NO\(_x\), respectively.

\[ \delta_i > 0 \text{ and } \chi_i \in (0, 1) \] are the penalty price of emission and weighting factor, respectively. The detailed fuel cost and emission quantity functions are given by

\[
C_i(x_i) = \alpha_{i,1} x_i^2 + \alpha_{i,2} x_i + \alpha_{i,3},
\]

\[
E_i^S(x_i) = \beta_{i,1} x_i^2 + \beta_{i,2} x_i + \beta_{i,3},
\]

\[
E_i^N(x_i) = \rho_{i,1} \exp(\rho_{i,2} x_i) + \rho_{i,3} x_i + \rho_{i,4},
\]

where \( \alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}, \beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \rho_{i,1}, \rho_{i,2}, \rho_{i,3}, \) and \( \rho_{i,4} \) are parameters. The communication typology of the market is designed in Fig. 5. Then, the economic emission dispatch problem of the whole market can be formulated as

\[
\text{(P7): } \min_{x \in X} \sum_{i \in V} f_i(x_i)
\]
algorithm was demonstrated by two motivating applications in the simulation.

APPENDIX

A. Proof of Lemma 3

By Lemma 2, \( \nabla f_{ij}^\omega \) is Lipschitz continuous with Lipschitz constant \( \frac{1}{\sigma_{ij}} \), which means

\[
\| \nabla_v f_{ij}^\omega (H_{ij} \mathbf{v}) - \nabla_u f_{ij}^\omega (H_{ij} \mathbf{u}) \| \\
= \| H_{ij}^\top \nabla_{H_{ij} \mathbf{v}} f_{ij}^\omega (H_{ij} \mathbf{v}) - H_{ij}^\top \nabla_{H_{ij} \mathbf{u}} f_{ij}^\omega (H_{ij} \mathbf{u}) \| \\
\leq \| H_{ij}^\top \| \| \nabla_{H_{ij} \mathbf{v}} f_{ij}^\omega (H_{ij} \mathbf{v}) - \nabla_{H_{ij} \mathbf{u}} f_{ij}^\omega (H_{ij} \mathbf{u}) \| \\
\leq \frac{H_{ij}^\top}{\sigma_{ij}} \| \mathbf{v} - \mathbf{u} \| = h_{ij} \| \mathbf{v} - \mathbf{u} \|. \\
\tag{53}
\]

implies \( \nabla_{\lambda_{ij}} f_{ij}^\omega (H_{ij} \lambda_{ij}) \) is Lipschitz continuous with constant \( h_{ij} \), which means

\( \nabla_{\lambda_{ij}} f_{ij}^\omega (H_{ij} \lambda_{ij}) = \nabla_{\lambda_{ij}} f_{ij}^\omega (H_{ij} \lambda_{ij}) + E^\top \) is also Lipschitz continuous with constant \( h_{ij} \).

B. Proof of Theorem 1

By the first-order optimality condition of (29) in terms of (1), we have

\[
0 \in \partial_{\lambda} Q(\lambda^{t+1}) + S[\mathcal{C}] (\lambda^{t+1} - \lambda^t) + \nabla_\lambda P(\lambda^t) + Z^\top D[\sigma] Z \omega^t \\
+ Z^\top D[\sigma] Z \omega^{t+1} - Z^\top D[\sigma] Z \lambda^{t+1}. \\
\tag{54}
\]

From the convexity of \( Q(\lambda) \), we have

\[
Q(\lambda) - Q(\lambda^{t+1}) \geq (\lambda - \lambda^{t+1})^\top S[\mathcal{C}] (\lambda - \lambda^{t+1}) \\
- (\lambda - \lambda^{t+1})^\top \nabla_\lambda P(\lambda^t) - (\lambda - \lambda^{t+1})^\top Z^\top \omega^{t+1} \\
+ (\lambda - \lambda^{t+1})^\top Z^\top D[\sigma] Z (\lambda^{t+1} - \lambda^t). \\
\tag{55}
\]

From the convexity and Lipschitz continuous differentiability of \( p_{ij} \), we have

\[
(\lambda - \lambda^{t+1})^\top \nabla_\lambda P(\lambda^t) = \sum_{i \in V} \sum_{j \in V_i} (\lambda_i - \lambda_i^{t+1})^\top \nabla_{\lambda_{ij}} p_{ij}(\lambda_{ij})
\]

VI. CONCLUSION

In this work, we considered a DOP in a multi-cluster network with an affine coupling constraint. In this problem, each cluster can make its decision based on the consensus protocol among the agents involved. To achieve the optimal solution of the whole network, a CDPG algorithm was proposed, where each agent can make updates with some simple-structured communications. The computational complexity with some simple-structured cost functions were discussed. The performance of the CDPG
By (56), we have
\[ \Phi(\lambda) = P(\lambda) - P(\lambda^{t+1}) + \|\lambda^t - \lambda^{t+1}\|_2^2 S[h]. \]  
(56)

Therefore, by multiplying the both sides of (57) by \((\omega - \omega^{t+1})^T\), we have
\[ (\omega - \omega^{t+1})^T \bar{D}[\pi](\omega^t - \omega^{t+1}) + (\omega - \omega^{t+1})^T Z \lambda^{t+1} = 0. \]  
(58)

By adding (55) and (56) together from the both sides, we have
\[ \Phi(\lambda^{t+1}) - \Phi(\lambda) \leq - (\lambda - \lambda^{t+1})^T S[c](\lambda^t - \lambda^{t+1}) + (\lambda - \lambda^{t+1})^T Z \omega^{t+1} + \|\lambda^t - \lambda^{t+1}\|_2^2 S[h] 
+ (\lambda - \lambda^{t+1})^T Z^T D[\pi] Z(\lambda^t - \lambda^{t+1}) 
= - (\lambda - \lambda^{t+1})^T S[c](\lambda^t - \lambda^{t+1}) 
- (\omega - \omega^{t+1})^T D[\pi](\omega^t - \omega^{t+1}) 
- (\omega - \omega^{t+1})^T Z \lambda^{t+1} + (\omega - \omega^{t+1})^T Z \lambda 
- (\omega^{t+1})^T Z \lambda^{t+1} + ||\lambda^t - \lambda^{t+1}\|_2^2 S[h] 
+ (\lambda - \lambda^{t+1})^T Z^T D[\pi] Z(\lambda^t - \lambda^{t+1}) 
= ||\lambda^t - \lambda^{t+1}\|_2^2 S[c] 
- ||\lambda^t - \lambda^{t+1}\|_2^2 S[h] 
- ||\lambda^t - \lambda^{t+1}\|_2^2 D[\pi] 
- ||\omega - \omega^{t+1}\|_2^2 D[\pi] 
+ (\omega^{t+1})^T Z \lambda - \omega^T Z \lambda^{t+1} + ||\lambda^t - \lambda^{t+1}\|_2^2 S[h] 
+ ||\lambda^t - \lambda^{t+1}\|_2^2 Z^T D[\pi] Z 
= ||\lambda^t - \lambda^{t+1}\|_2^2 S[c] - \frac{1}{2} Z^T D[\pi] Z 
- ||\lambda^t - \lambda^{t+1}\|_2^2 S[h] - \frac{1}{2} Z^T D[\pi] Z 
- ||\lambda^t - \lambda^{t+1}\|_2^2 D[\pi] Z 
+ ||\lambda^t - \lambda^{t+1}\|_2^2 Z^T D[\pi] Z 
= ||\lambda^t - \lambda^{t+1}\|_2^2 S[c] - \frac{1}{2} Z^T D[\pi] Z 
+ ||\lambda^t - \lambda^{t+1}\|_2^2 S[h] - \frac{1}{2} Z^T D[\pi] Z 
+ ||\lambda^t - \lambda^{t+1}\|_2^2 D[\pi] Z 
+ ||\lambda^t - \lambda^{t+1}\|_2^2 Z^T D[\pi] Z. 
(60)

where the first inequality is from the convexity of \(\Phi\) and the third inequality is from Cauchy-Schwarz inequality. Therefore,
\[ \Phi(\lambda^{t+1}) - \Phi(\lambda) \leq \frac{1}{T+1}(\|\omega^*\|_2^2 D[\pi] + ||\omega^0\|_2^2 D[\pi] 
+ ||\lambda^t - \lambda^0\|_2^2 S[c] - \frac{1}{2} Z^T D[\pi] Z 
+ ||\lambda^t - \lambda^0\|_2^2 S[h] - \frac{1}{2} Z^T D[\pi] Z 
+ ||\lambda^t - \lambda^0\|_2^2 D[\pi] Z 
+ ||\lambda^t - \lambda^0\|_2^2 Z^T D[\pi] Z). 
(61)\]
Based on (25), (26) and (28), we have
\[ \Phi(\lambda) - \Phi(\lambda^*) + \omega^T Z \lambda \geq 0. \]

Letting \( \lambda = \lambda^{T+1} \) in (63) gives
\[ \Phi(\lambda^{T+1}) - \Phi(\lambda^*) \geq -\omega^* \| Z \lambda^{T+1} \|. \]

By combining the first inequality in (62) and (64), we have
\[ \| \omega^* \| \| Z \lambda^{T+1} \| \leq \frac{\Theta}{T+1}. \]

By (64) and (65), we have
\[ \Phi(\lambda^{T+1}) - \Phi(\lambda^*) \geq -\frac{\Theta}{T+1}. \]

By combining (62), (65) and (66), (47) and (48) are proved.

2) Scenario 2: If \( Z \lambda^{T+1} = 0 \), we let \( \lambda = \lambda^* \) and \( \omega = 2\omega^* \) in (59), which directly gives
\[ \Phi(\lambda^{T+1}) - \Phi(\lambda^*) \leq \frac{\Theta}{T+1} \]
by the same derivation process of (62). Then, considering that \( \| Z \lambda^{T+1} \| = 0 \) and \( \Phi(\lambda^{T+1}) - \Phi(\lambda^*) \geq -\| \omega^* \| \| Z \lambda^{T+1} \| = 0 \), (47) and (48) hold as well.

REFERENCES

[1] L. Luo, N. Chakraborty, and K. Sycara, “Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks,” IEEE Transactions on Robotics, vol. 31, no. 1, pp. 19–30, 2014.

[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] F. Guo, C. Wen, J. Mao, G. Li, and Y.-D. Song, “A distributed hierarchical algorithm for multi-cluster constrained optimization,” Automatica, vol. 77, pp. 230–238, 2017.

[5] C.-X. Shi and G.-H. Yang, “Multi-cluster distributed optimization via random sleep strategy,” Journal of the Franklin Institute, vol. 356, no. 10, pp. 5353–5377, 2019.

[6] J. Zhou, C. Wang, Y. Li, P. Wang, C. Li, P. Lu, and L. Mo, “A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security,” Applied Mathematical Modelling, vol. 45, pp. 684–704, 2017.

[7] I. Notarnicola, M. Franceschelli, and G. Notarstefano, “A duality-based approach for distributed min-max optimization with application to demand side management,” in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016, pp. 1877–1882.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] A. Simoniello 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.

[14] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, “Fully distributed algorithms for convex optimization problems,” SIAM Journal on Optimization, vol. 20, no. 6, pp. 3260–3279, 2010.

[15] I. Necoara, 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.

[16] J. Wang and G. Hu, “Distributed discrete-time optimization with coupling constraints based on dual proximal gradient method in multi-agent networks,” arXiv preprint arXiv:2108.10652, 2021.

[17] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, “Dual decomposition for multi-agent distributed optimization with coupling constraints,” Automatica, vol. 84, pp. 149–158, 2017.

[18] B. Yang and M. Johansson, “Distributed optimization and games: A tutorial overview,” Networked Control Systems, pp. 109–148, 2010.

[19] I. Necoara and V. Nedelcu, “On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems,” Automatica, vol. 55, pp. 209–215, 2015.

[20] S. A. Alghunaim, K. Yuan, and A. H. Sayed, “A proximal dual consensus admm method for multi-agent constrained optimization,” IEEE Transactions on Automatic Control, vol. 65, no. 11, pp. 4554–4567, 2019.

[21] D.-K. Li, C.-X. Shi, and G.-H. Yang, “Gossip-based distributed hierarchical algorithm for multi-cluster constrained optimization,” IET Control Theory & Applications, vol. 13, no. 15, pp. 2346–2355, 2019.

[22] X. Li, G. Feng, and L. Xie, “Distributed proximal algorithms for multi-agent optimization with coupled inequality constraints,” IEEE Transactions on Automatic Control, 2020.
[23] I. Notarnicola and G. Notarstefano, “A duality-based approach for distributed optimization with coupling constraints,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 14 326–14 331, 2017.

[24] ———, “Asynchronous distributed optimization via randomized dual proximal gradient,” IEEE Transactions on Automatic Control, vol. 62, no. 5, pp. 2095–2106, 2016.

[25] 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.

[26] 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.

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

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

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

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

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

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

[40] M. A. Hanson, “On sufficiency of the kuhn-tucker conditions,” Journal of Mathematical Analysis and Applications, vol. 80, no. 2, pp. 545–550, 1981.