2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed Optimization of Clique-wise Coupled Problems

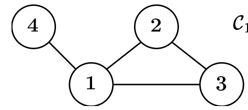
Yuto Watanabe and Kazunori Sakurama

***Abstract*— This study addresses a distributed optimization with a novel class of coupling of variables, called *clique-wise coupling*. A clique is a node set of a complete subgraph of an undirected graph. This setup is an extension of pairwise coupled optimization problems (e.g., consensus optimization) and allows us to handle coupling of variables consisting of more than two agents systematically. To solve this problem, we propose a clique-based linearized ADMM algorithm, which is proved to be distributed. Additionally, we consider objective functions given as a sum of nonsmooth and smooth convex functions and present a more flexible algorithm based on the FLiP-ADMM algorithm. Moreover, we provide convergence theorems of these algorithms. Notably, all the algorithmic parameters and the derived condition in the theorems depend only on local information, which means that each agent can choose the parameters in a distributed manner. Finally, we apply the proposed methods to a consensus optimization problem and demonstrate their effectiveness via numerical experiments.**

I. Introduction

In recent years, distributed optimization has attracted much attention in the control, signal processing, and machine learning communities. In this field, a large body of studies has been dedicated to *pairwise coupled optimization problems*. In this type of problem, every coupling of variables comprises two agents' decision variables corresponding to the communication path between the two agents. The most representative example of this setup is consensus optimization problems, and many studies have presented sophisticated algorithms, such as [1]–[4]. Recently, [5] and [6] have investigated distributed optimization problems with pairwise linear constraints. Their applications are not limited to consensus optimization but contain formation control, distributed model predictive control, and so on. Moreover, in the field of multi-agent control, various coordination tasks (e.g., rendezvous and formation) have been formulated in a pairwise coupled form [7]–[9].

This study addresses a more general form of distributed optimization than the conventional pairwise coupled ones to handle coupling of more than two decision variables. Consider a multi-agent system with n agents over a communication network, expressed by a time-invariant undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with $\mathcal{N} = \{1, \dots, n\}$ and an edge set $\mathcal{E}$. Now, we aim to solve the following optimization



$\mathcal{C}_1 = \{1\},\ \mathcal{C}_2 = \{2\},\ \mathcal{C}_3 = \{3\},\ \mathcal{C}_4 = \{4\},$
$\mathcal{C}_5 = \{1,2\},\ \mathcal{C}_6 = \{2,3\},\ \mathcal{C}_7 = \{1,3\},$
$\mathcal{C}_8 = \{1,4\},\ \mathcal{C}_9 = \{1,2,3\}$

Fig. 1: Example of cliques.

problem, called a *clique-wise coupled optimization problem*, in a distributed manner:

$$\begin{array}{ll} \underset{\substack{x_i \in \mathbb{R}^d,\, i \in \mathcal{N} \\ y_l \in \mathbb{R}^{q_l},\, l \in \mathcal{Q}_\mathcal{G}}}{\text{minimize}} & \displaystyle\sum_{i \in \mathcal{N}} f_i(x_i) + \sum_{l \in \mathcal{Q}_\mathcal{G}} g_l(y_l) \\ \text{subject to} & \underbrace{A_l x_{\mathcal{C}_l} + B_l y_l = c_l \quad \forall l \in \mathcal{Q}_\mathcal{G} \subset \bar{\mathcal{Q}}_\mathcal{G}}_{\text{Clique-wise coupling w.r.t. } \mathcal{C}_l \subset \mathcal{N}} \end{array} \tag{1}$$

with $A_l \in \mathbb{R}^{p_l \times |\mathcal{C}_l| d}$, $B_l \in \mathbb{R}^{p_l \times q_l}$, and $c_l \in \mathbb{R}^{p_l}$, where $f_i : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, $i \in \mathcal{N}$ and $g_l : \mathbb{R}^{q_l} \to \mathbb{R} \cup \{+\infty\}$, $l \in \mathcal{Q}_\mathcal{G}$ are proper, closed, and convex functions (possibly nonsmooth). The vector x_i represents the decision variable of agent i, and y_l represents the variable with respect to clique l. For $x_1, \dots, x_n$, and the set $\mathcal{C}_l = \{j_1, \dots, j_{|\mathcal{C}_l|}\} \subset \mathcal{N}$, let $x_{\mathcal{C}_l}$ denote $x_{\mathcal{C}_l} = [x_{j_1}^\top, \dots, x_{j_{|\mathcal{C}_l|}}^\top]^\top \in \mathbb{R}^{|\mathcal{C}_l| d}$. Here, the set $\mathcal{C}_l$ represents a clique, i.e., a complete subgraph in the graph $\mathcal{G}$ [10]. $\bar{\mathcal{Q}}_\mathcal{G}$ is the index set of all the cliques in $\mathcal{G}$, and $\mathcal{Q}_\mathcal{G}$ is a subset of $\bar{\mathcal{Q}}_\mathcal{G}$. For example, in the undirected graph in Fig. 1, $\bar{\mathcal{Q}}_\mathcal{G} = \{1, \dots, 9\}$ holds, and the cliques $\mathcal{C}_1, \dots, \mathcal{C}_9$ are obtained as shown in Fig. 1. Note that the nodes and edges are always cliques, and hence Problem (1) always contains conventional pairwise coupled optimization problems.

A remarkable benefit of the clique-wise coupling framework is that we can systematically handle variable couplings of more than two agents. Possible applications include consensus optimization [1], [2], [4], [11], formation control [5], [7], robust PCA (with clique-wise trace norm minimization) [12], [13], network lasso (fused lasso, total variation regularization) [14]–[16], multi-task learning [17], and (clique-wise) resource allocation [18], as illustrated in Table I. This table shows concrete formulations of these applications corresponding to Problem (1). Notice that one can also deal with other examples, e.g., semidefinite programming with chordal sparsity [19] and its application for distributed design of decentralized controllers [20], as clique-wise coupled problems with an appropriate transformation.

In this study, we propose a novel distributed algorithm, the Clique-based Linearized ADMM (CL-ADMM) algorithm, for Problem (1) based on the framework of the alternating direction method of multipliers (ADMM) [11], [21], [22] with the linearization technique and localized algorithmic parameters. The proposed method can be implemented with

This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society".

Y. Watanabe and K. Sakurama are with the Department of Systems Science, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606–8501, Japan, y-watanabe@sys.i.kyoto-u.ac.jp, sakurama@i.kyoto-u.ac.jp.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

TABLE I: Practical application examples of Problem (1).

Applications	Constraints	$f_{[l]}$	f_i
Consensus optimization [1]–[4]	$x_{\mathcal{C}_l} - y_l = 0$	Indicator function for $\mathcal{D}_l = \{y_l : \exists \xi \text{ s.t. } y_l = \mathbf{1}_{\lvert\mathcal{C}_l\rvert} \otimes \xi\}$ [1]	$f_i(x_i)$
Robust PCA [12], [13] (clique-wise trace norm minimization)	$Y_j = S_j + L_j \quad \forall j \in \mathcal{C}_l$ [2]	$\theta_l \lVert L_{\mathcal{C}_l} \rVert_*$	$\lVert S_i \rVert_1$
Formation control [5], [7]–[9]	$x_{i,t+1} = A_i x_{i,t} + B_i u_{i,t}$ $y_{l,t} = x_{i,t} - x_{j,t}$	$\frac{1}{2}\sum_{t=1}^{T} \lVert [y_{l,t} - r_{ij} \rVert_{Q_{ij}}^2$	$\frac{1}{2}\sum_{t=1}^{T-1} u_{i,t}^\top R_i u_{i,t}$
Network lasso [14]–[16]	$x_i - x_j - y_l = 0$	$\lambda_{ij} \lVert y_l \rVert$	loss function $\ell_i(x_i)$
Multi-task learning [17]	$x_j - y_{l,j} = 0$ $\forall j \in \mathcal{C}_l = \{j_1, \ldots, j_{\lvert\mathcal{C}_l\rvert}\}$	$\lambda_l \lVert [y_{l,j_1}, \ldots, y_{l,j_{\lvert\mathcal{C}_l\rvert}}] \rVert_*$	loss function $\ell_i(x_i)$
(Clique-wise) resource allocation [18]	$x_{\mathcal{C}_l} - y_l = 0$	Indicator function for $\mathcal{D}_l = \{y_l : \mathbf{1}^\top y_l = N_l > 0,\, y_l \geq 0\}$	$\frac{1}{2}\lVert x_i - x_* \rVert^2$

local communication. Additionally, we consider that the objective functions f_i and g_l are given as a sum of a non-smooth convex and smooth convex functions. Under this setup, we provide a more flexibly implementable algorithm, called the Clique-based Linearized FLiP-ADMM (CL-FLiP-ADMM), based on the Function Linearized Proximal ADMM (FLiP-ADMM) algorithm in [22]. Furthermore, through the convergence analysis of the CL-ADMM and CL-FLiP-ADMM, we prove the exact convergence to an optimal solution under fully localized conditions. Finally, we apply the proposed methods to consensus optimization and demonstrate their effectiveness through numerical experiments. The experimental result of consensus optimization implies that the clique-wise handling of pairwise constraints can enhance the performance.

The major contributions of this paper are as follows: (a) We propose a highly expressive framework in (1) suitable for distributed optimization and provide a variety of practical examples as shown in Table I; (b) The algorithmic parameters in the CL-ADMM and CL-FLiP-ADMM are also distributed. Moreover, we provide convergence theorems with no global parameter, which means that each agent can choose its parameters in a distributed manner; (c) The CL-ADMM and CL-FLiP-ADMM can alleviate the computational burden because the agents in the same clique can share the computation of the proximal mapping of g_l.

Recently, several studies (e.g., [23]–[26]) have presented distributed algorithms to solve optimization problems with globally coupled constraints. This setup allows us to consider constraints involving all decision variables. However, the existing methods for this setup cannot enjoy our proposed methods' features, such as the contributions (b) and (c) in the above paragraph. Moreover, the clique-wise coupled problem in (1) can be solved more flexibly than the globally coupled problems because the primal problem (1) can be solved directly by well-known methods (e.g., ADMM [11], [21], [22] and Condat-Vũ [27], [28]) in a distributed manner. This is advantageous in terms of ease of solving problems. To apply these well-known methods to the globally coupled problems, we need to introduce additional auxiliary variables (e.g., to take their dual problems to transform them into a consensus optimization). This approach may increase the size of variables and degrade the performance.

Finally, notice that Problem (1) is more general than optimization problems with clique-wise coupled constraints in the authors' paper [18] because we can handle not only constraints but also regularization terms in Problem (1).

The rest of the paper is organized as follows. Section II provides preliminaries. Section III presents the proposed algorithms, and Section IV presents the convergence theorems. In Section V, we apply the proposed methods to consensus optimization. Finally, Section VI concludes our paper.

II. Preliminaries

A. Notation

Let $\mathbb{R}$ and $\mathbb{N}$ be the set of real numbers and that of positive integers, respectively. Let $|\cdot|$ be the number of elements in a countable finite set. Let $I_d \in \mathbb{R}^{d\times d}$ denote the $d \times d$ identity matrix. We omit the subscript of I_d when the dimension is obvious. Let $\mathbf{1}_d = [1, \ldots, 1]^\top \in \mathbb{R}^d$. For a m-dimensional vector $[a_1, \ldots, a_i, \ldots, a_N]^\top \in \mathbb{R}^N$, $\text{diag}(a)$ denotes the diagonal matrix whose ith diagonal entry is a_i. Similarly, for matrices $R_1, \ldots, R_i, \ldots, R_N$, blk-diag$(R_1, \ldots, R_N)$ represents the block diagonal matrix whose ith diagonal block is R_i. For $\mathcal{M} = \{1, \ldots, N\}$, blk-diag$([R_j]_{j\in\mathcal{M}})$ represents blk-diag$(R_1, \ldots, R_N)$. For $v,\, u \in \mathbb{R}^m$, and a positive definite and symmetric matrix $Q \in \mathbb{R}^{m\times m}$, $\langle v, u\rangle_Q := v^\top Q u$ denotes the inner product of v and u with respect to Q. Additionally, we define the norm $\|\cdot\|_Q$ as $\|v\|_Q = \sqrt{\langle v, v\rangle_Q}$ for a vector $v \in \mathbb{R}^m$. When $Q = I_m$, we simply write $\langle\cdot,\cdot\rangle_{I_m}$ and $\|\cdot\|_{I_m}$ as $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively. For $v \in \mathbb{R}^m$, $\|v\|_1$ denotes the ℓ_1 norm of v. For a matrix $R \in \mathbb{R}^{d_1\times d_2}$, $\|R\|_*$ denotes the trace norm of R, i.e., the sum of its singular values. Let $\lambda_{\max}(Q)$ and $\lambda_{\min}(Q)$ be the largest and smallest eigenvalues of Q, respectively. For a vector $\mathbf{v} = [v_1^\top, \ldots, v_j^\top, \ldots, v_N^\top]^\top \in \mathbb{R}^{Nd}$ with vectors $v_1, \ldots, v_j, \ldots, v_N \in \mathbb{R}^d$, $[\mathbf{v}]_j$ represents the operation to

[1] In this formulation, we can include some proximable functions (e.g., ℓ_1 norm regularization) in g_l. For details, see Subsection V-A.

[2] Here, $Y = [Y_1, \ldots, Y_n]$ is a data matrix. For example, Y is a sequence of images, and each column of Y corresponds to each frame. We assume that Y can be decomposed into a sparse matrix $\hat{S} = [S_1, \ldots, S_n]$ and a low-rank matrix $L = [L_1, \ldots, L_n]$.

extract the jth vector v_j from $\mathbf{v}$, that is,

$$[\mathbf{v}]_j = v_j \in \mathbb{R}^d.$$

For a vector $\mathbf{x} = [x_1^\top, \ldots, x_n^\top]^\top \in \mathbb{R}^{nd}$ with $x_1, \ldots, x_n \in \mathbb{R}^d$ and a subset $\mathcal{C} = \{j_1, \ldots, j_{|\mathcal{C}|}\} \subset \mathcal{N}$, let $x_\mathcal{C}$ be $x_\mathcal{C} = [x_{j_1}^\top, \ldots, x_{j_{|\mathcal{C}|}}^\top]^\top \in \mathbb{R}^{|\mathcal{C}|d}$, where $\{j_1, \ldots, j_{|\mathcal{C}|}\}$ is a strictly monotonically increasing sequence. For a differentiable function $f : \mathbb{R}^{nd} \to \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^{nd}$, we write $\nabla f(\mathbf{x}) = \partial f / \partial \mathbf{x}(\mathbf{x})$.

For a proper, closed, and convex function $g : \mathbb{R}^d :\to \mathbb{R} \cup \{+\infty\}$, a positive definite and symmetric matrix $Q \in \mathbb{R}^{d \times d}$, and $x \in \mathbb{R}^d$, the proximal mapping of g with respect to Q is represented by $\mathrm{prox}_g^Q(x) = \arg\min_{y \in \mathbb{R}^d}\{g(y) + (1/2)\|x - y\|_Q^2\}$. When $Q = I_d$, we write $\mathrm{prox}_g^{I_d}(\cdot) = \mathrm{prox}_g(\cdot)$. When the proximal mapping of g can be computed efficiently, the function g is said to be *proximable*.

B. Graph Theory

Here, we provide graph-theoretic concepts. Consider a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with a node set $\mathcal{N} = \{1, \ldots, n\}$ and an edge set $\mathcal{E}$ consisting of pairs $\{i, j\}$ of different nodes $i, j \in \mathcal{N}$. Note that throughout this paper, we consider undirected graphs and do not distinguish $\{i, j\}$ and $\{j, i\}$ for each $\{i, j\} \in \mathcal{E}$. For $i \in \mathcal{N}$ and G, let $\mathcal{N}_i \subset \mathcal{N}$ be the *neighbor set* of node i over $\mathcal{G}$, defined as $\mathcal{N}_i = \{j \in \mathcal{N} : \{i, j\} \in \mathcal{E}\} \cup \{i\}$.

For an undirected graph $\mathcal{G}$, consider a set $\mathcal{C} \subset \mathcal{N}$. For $\mathcal{C}$ and $\mathcal{E}$, let $\mathcal{E}|_\mathcal{C}$ be $\mathcal{E}|_\mathcal{C} = \{\{i, j\} \in \mathcal{E} : i, j \in \mathcal{C}\}$. We call $\mathcal{G}|_\mathcal{C} = (\mathcal{C}, \mathcal{E}|_\mathcal{C})$ a subgraph induced by $\mathcal{C}$. If $\mathcal{G}|_\mathcal{C}$ is complete, $\mathcal{C}$ is called a *clique* in $\mathcal{G}$. We define $\bar{\mathcal{Q}}_\mathcal{G} = \{1, 2, \ldots, q\}$ as the set of indices of all the cliques in $\mathcal{G}$. For $\bar{\mathcal{Q}}_\mathcal{G}$, the set $\mathcal{Q}_\mathcal{G}$ represents a subset of $\bar{\mathcal{Q}}_\mathcal{G}$. If a clique $\mathcal{C}$ is not contained by any other cliques, $\mathcal{C}$ is said to be *maximal*. Let $\mathcal{Q}_\mathcal{G}^{\max} (\subset \bar{\mathcal{Q}}_\mathcal{G})$ be the set of indices of all the maximal cliques in $\mathcal{G}$. For $i \in \mathcal{N}$, we define $\bar{\mathcal{Q}}_\mathcal{G}^i$ as an index set of all the cliques containing i, that is, $\bar{\mathcal{Q}}_\mathcal{G}^i = \{l \in \bar{\mathcal{Q}}_\mathcal{G} : i \in \mathcal{C}_l\}$. For each $i \in \mathcal{N}$, $\mathcal{N}_i$, and $\mathcal{C}_l$, $l \in \bar{\mathcal{Q}}_\mathcal{G}^i$,

$$\mathcal{N}_i = \bigcup_{l \in \bar{\mathcal{Q}}_\mathcal{G}^i} \mathcal{C}_l. \tag{2}$$

holds [9]. Note that agent i can independently compute the cliques that it belongs to, i.e., $\mathcal{C}_l$, $l \in \bar{\mathcal{Q}}_\mathcal{G}^i$, from the undirected subgraph $(\mathcal{N}_i, \mathcal{E}_i)$ with $\mathcal{E}_i = \{\{i, j\} \in \mathcal{E} : j \in \mathcal{N}_i\}$.

III. Algorithm Description

A. Clique-based Linearized ADMM Algorithm

The proposed algorithm to solve (1), the Clique-based Linearized ADMM (CL-ADMM) algorithm, is illustrated in Algorithm 1. This algorithm is based on the linearized ADMM algorithm [21], [22] and can be implemented in a distributed manner from (2). Here, let $\mathcal{Q}_\mathcal{G}^i := \bar{\mathcal{Q}}_\mathcal{G}^i \cap \mathcal{Q}_\mathcal{G}$. Besides, α_i, $i \in \mathcal{N}$ is an agent-wise algorithmic parameter, and β_l, γ_l, and $\phi_l > 0$, $l \in \mathcal{Q}_\mathcal{G}$ are clique-wise parameters. In Algorithm (1), y_l^k represents the estimate of an optimal y_l, and u_l^k represents the dual variable for $l \in \mathcal{Q}_\mathcal{G}$ at the kth iteration. In the x_i-update in (3a), $\pi_l : \mathcal{C}_l \to \{1, \ldots, |\mathcal{C}_l|\}$ is the one-to-one mapping satisfying $\pi_l(i_j) = j$ for $\mathcal{C}_l = \{i_1, \ldots, i_j, \ldots, i_{|\mathcal{C}_l|}\}$ with $1 \le i_1 < i_j < i_{|\mathcal{C}_l|} \le n$. Note that if the y_l-subproblem in (3b) has multiple solutions, the agents in $\mathcal{C}_l$ must choose the same value as y_l^{k+1}. In addition, $\bigcup_{l \in \mathcal{Q}_\mathcal{G}} \mathcal{C}_l = \mathcal{N}$ is assumed here.

Algorithm 1 Clique-based Linearized ADMM (CL-ADMM) Algorithm

Require: x_i^0, $\alpha_i > 0$, y_l^0, u_l^0, $\beta_l > 0$, $\gamma_l > 0$, and $\phi_l > 0$ for all $l \in \mathcal{Q}_\mathcal{G}^i$ $(:= \bar{\mathcal{Q}}_\mathcal{G}^i \cap \mathcal{Q}_\mathcal{G})$.

for $k = 0, 1, \ldots$ **do**

  Update x_i^k by

$$x_i^{k+1} = \mathrm{prox}_{\alpha_i f_i}\big(x_i^k - \alpha_i \textstyle\sum_{l \in \mathcal{Q}_\mathcal{G}^i} \big[A_l^\top u_l^k + \gamma_l A_l^\top \left(A_l x_{\mathcal{C}_l}^k + B_l y_l^k - c_l\right)\big]_{\pi_l(i)}\big). \tag{3a}$$

  Send x_i^{k+1} to all agents in $\mathcal{N}_i \setminus \{i\}$.

  Update y_l^k and u_l^k for all $l \in \mathcal{Q}_\mathcal{G}^i$ by

$$y_l^{k+1} = \mathrm{prox}_{\beta_l g_l}\big(y_l^k - \beta_l \big(B_l^\top u_l^k + \gamma_l B_l^\top \big(A_l x_{\mathcal{C}_l}^{k+1} + B_l y_l^k - c_l\big)\big)\big) \tag{3b}$$

$$u_l^{k+1} = u_l^k + \phi_l \gamma_l \left(A_l x_{\mathcal{C}_l}^{k+1} + B_l y_l^{k+1} - c_l\right). \tag{3c}$$

end for

Example 1: Consider $\mathcal{C}_l = \{2, 3, 5\}$ and $y_l = [y_{l,1}^\top, y_{l,2}^\top, y_{l,3}^\top]^\top \in \mathbb{R}^{|\mathcal{C}_l|d}$. Then, $\pi_l(2) = 1$, $\pi_l(3) = 2$, and $\pi_l(5) = 3$ hold. Besides, for y_l, $[y_l]_{\pi_l(2)} = y_{l,1}$, $[y_l]_{\pi_l(3)} = y_{l,2}$, and $[y_l]_{\pi_l(5)} = y_{l,3}$ are obtained.

We provide an interpretation of Algorithm 1. First, we give the aggregated form of Problem (1) as follows:

$$\begin{aligned} \underset{\mathbf{x}, \mathbf{y}}{\text{minimize}} \quad & F(\mathbf{x}) + G(\mathbf{y}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}, \end{aligned} \tag{4}$$

where $F(\mathbf{x}) = \sum_{i \in \mathcal{N}} f_i(x_i)$, $G(\mathbf{y}) = \sum_{l \in \mathcal{Q}_\mathcal{G}} g_l(y_l)$, $\mathbf{x} = [x_1^\top, \ldots, x_n^\top]^\top$, $\mathbf{y} = [y_1^\top, \ldots, y_{|\mathcal{Q}_G|}^\top]^\top$, $\mathbf{A} = \text{blk-diag}([A_l]_{l \in \mathcal{Q}_\mathcal{G}})$, $\mathbf{B} = \text{blk-diag}([B_l]_{l \in \mathcal{Q}_\mathcal{G}})$, and $\mathbf{c} = [c_1^\top, \ldots, c_{|\mathcal{Q}_G|}^\top]^\top$. Now, $\mathbf{W}$ is defined as the matrix satisfying the following relationships for any $\mathbf{x} \in \mathbb{R}^{dn}$:

$$\begin{aligned} &\mathbf{W} = \left[W_1^\top, \ldots, W_{|\mathcal{Q}_G|}^\top\right]^\top \in \mathbb{R}^{\left(\sum_{l \in \mathcal{Q}_\mathcal{G}} |\mathcal{C}_l|\right) d \times nd}, \\ &W_l \mathbf{x} = x_{\mathcal{C}_l} \quad \forall l \in \mathcal{Q}_\mathcal{G}. \end{aligned} \tag{5}$$

Then, the Lagrangian function is obtained as follows:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{u}) = F(\mathbf{x}) + G(\mathbf{y}) + \langle \mathbf{u}, \mathbf{A}\mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c} \rangle. \tag{6}$$

By applying the linearized ADMM algorithm [21], [22] to the problem in (4), we obtain the following algorithm, where the scalar-valued constants in the normal linearized ADMM are replaced by the diagonal matrices $\mathbf{D}_\alpha$, $\mathbf{D}_\beta$, $\mathbf{\Gamma}$, and $\mathbf{\Phi}$:

$$\mathbf{x}^{k+1} = \mathrm{prox}_F^{\mathbf{D}_\alpha^{-1}}(\mathbf{x}^k - \mathbf{D}_\alpha \mathbf{W}^\top \mathbf{A}^\top (\mathbf{u}^k + \mathbf{\Gamma}(\mathbf{A}\mathbf{W}\mathbf{x}^k + \mathbf{B}\mathbf{y}^k - \mathbf{c}))) \tag{7a}$$

$$\mathbf{y}^{k+1} = \mathrm{prox}_G^{\mathbf{D}_\beta^{-1}}(\mathbf{y}^k - \mathbf{D}_\beta \mathbf{B}^\top (\mathbf{u}^k + \mathbf{\Gamma}(\mathbf{A}\mathbf{W}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^k - \mathbf{c}))) \tag{7b}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{\Phi}\mathbf{\Gamma}(\mathbf{A}\mathbf{W}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c}). \tag{7c}$$

Then, by setting

$$\mathbf{D}_\alpha = \text{blk-diag}([\alpha_i I_d]_{i\in\mathcal{N}}),\ \mathbf{D}_\beta = \text{blk-diag}([\beta_l I_{q_l}]_{j\in\mathcal{Q}_\mathcal{G}}),$$
$$\mathbf{\Gamma} = \text{blk-diag}([\gamma_l I_{p_l}]_{j\in\mathcal{Q}_\mathcal{G}}),\ \mathbf{\Phi} = \text{blk-diag}([\phi_l I_{p_l}]_{j\in\mathcal{Q}_\mathcal{G}})$$

with $\alpha_i > 0$, $i \in \mathcal{N}$ and β_l, γ_l, $\phi_l > 0$, $l \in \mathcal{Q}_\mathcal{G}$, we obtain the CL-ADMM in Algorithm 1 from Algorithm (7). This follows from the following proposition.

Proposition 1: Consider $i \in \mathcal{N}$, $\mathcal{C}_l$, $l \in \mathcal{Q}^i_\mathcal{G}$, $\mathbf{W}$ in (5), the mapping $\pi_l : \mathcal{C}_l \to \{1,\dots,|\mathcal{C}_l|\}$, and $\mathbf{v} = [v_1^\top,\dots,v_l^\top,\dots,v_{|\mathcal{Q}_\mathcal{G}|}^\top]^\top$ with any $v_l \in \mathbb{R}^{|\mathcal{C}_l|d}$. Then, it holds that $[\mathbf{W}^\top\mathbf{v}]_i = \sum_{l\in\mathcal{Q}^i_\mathcal{G}}[v_l]_{\pi_l(i)} \in \mathbb{R}^d$.

Proof: From (5), the (j,i) block of $W_l \in \mathbb{R}^{|\mathcal{C}_l|d\times nd}$ can be written as $[W_l]_{ji} = w_{l,ji}I_d$, where

$$w_{l,ji} = \begin{cases} 1, & i\in\mathcal{C}_l \text{ and } \pi_l(i) = j \\ 0, & \text{otherwise}\end{cases}. \tag{8}$$

Additionally, we have $\mathbf{W}^\top\mathbf{u} = \sum_{l\in\mathcal{Q}_\mathcal{G}}(W_l)^\top u_l$. Then, for the ith block $[\mathbf{W}^\top\mathbf{v}]_i \in \mathbb{R}^d$ of $\mathbf{W}^\top\mathbf{v}$, we obtain $[\mathbf{W}^\top\mathbf{v}]_i = \sum_{l\in\mathcal{Q}_\mathcal{G}}\left(\sum_{j=1}^{|\mathcal{C}_l|}[W_l]_{ji}[v_l]_j\right) = \sum_{l\in\mathcal{Q}^i_\mathcal{G}}[v_l]_{\pi_l(i)}$ because $[W^l]_{ji}[v^l]_j = [v^l]_{\pi_l(i)}$ holds for j satisfying $\pi_l(i) = j$, and $[W^l]_{ji}[v^l]_j = 0$ holds otherwise. ■

Remark 1: One of the advantages of the CL-ADMM in Algorithm 1 is that the agents in the same clique $\mathcal{C}_l$ can share the computation of the y_l-subproblem in (3b). Hence, we can alleviate computational burdens per iteration by allocating the computation.

Remark 2: The proposed algorithm is based on the linearized ADMM, which is essential for its distributed implementation. This is because the augmented Lagrangian of (6) can be separated into an agent-wise form by eliminating its coupled terms by the linearization technique.

B. Clique-based Linearized FLiP-ADMM Algorithm

We provide a more flexible algorithm, the Clique-based Linearized FLiP-ADMM (CL-FLiP-ADMM), than Algorithm 1 based on the FLiP-ADMM in [22]. Suppose that the objective functions f_i and g_l can be separated as

$$f_i = f_i^1 + f_i^2, \quad g_l = g_l^1 + g_l^2, \tag{9}$$

where f_i^1 and g_l^1 are proper, closed, and convex functions, and f_i^2 and g_l^2 are proper, closed, convex, and smooth functions.

Now, we present the Clique-based Linearized FLiP-ADMM (CL-FLiP-ADMM) algorithm in Algorithm 2. This algorithm can be implemented distributedly in the same manner as Algorithm 1. When $f_i^2 = g_l^2 = 0$, this algorithm is reduced to Algorithm 1. Notably, this algorithm allows us to avoid computing the proximal map involving f_i^2 and g_l^2.

Note that although the FLiP-ADMM algorithm with the linearization technique is named as doubly linearized ADMM in [22], we refer to Algorithm 2 as CL-FLiP-ADMM for the sake of consistency with Algorithm 1.

IV. Convergence Analysis

This section presents the key convergence theorems of the CL-ADMM in Algorithm 1 and CL-FLiP-ADMM in Algorithm 2. We now assume the following assumption.

Algorithm 2 Clique-based Linearized FLiP-ADMM (CL-FLiP-ADMM) Algorithm

Require: x_i^0, $\alpha_i > 0$, y_l^0, u_l^0, $\beta_l > 0$, $\gamma_l > 0$, and $\phi_l > 0$ for all $l \in \mathcal{Q}^i_\mathcal{G}$ $(:= \bar{\mathcal{Q}}^i_\mathcal{G} \cap \mathcal{Q}_\mathcal{G})$.

for $k = 0, 1, \dots$ **do**

  Update x_i^k by

$$x_i^{k+1} = \text{prox}_{\alpha_i f_i^1}\big(x_i^k - \alpha_i\big(\nabla f_i^2(x_i^k) + \sum_{l\in\mathcal{Q}^i_\mathcal{G}}\big[A_l^\top u_l^k + \gamma_l A_l^\top\big(A_l x^k_{\mathcal{C}_l} + B_l y_l^k - c_l\big)\big]_{\pi_l(i)}\big)\big). \tag{10a}$$

  Send x_i^{k+1} to all agents in $\mathcal{N}_i \setminus \{i\}$.

  Update y_l^k and u_l^k for all $l \in \mathcal{Q}^i_\mathcal{G}$ by

$$y_l^{k+1} = \text{prox}_{\beta_l g_l^1}\big(y_l^k - \beta_l\big(\nabla g_l^2(y_l^k) + B_l^\top u_l^k + \gamma_l B_l^\top\big(A_l x^{k+1}_{\mathcal{C}_l} + B_l y_l^k - c_l\big)\big)\big) \tag{10b}$$

$$u_l^{k+1} = u_l^k + \phi_l\gamma_l\left(A_l x^{k+1}_{\mathcal{C}_l} + B_l y_l^{k+1} - c_l\right). \tag{10c}$$

end for

Assumption 1: The following statements hold:

(a) $f_i : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$, $i \in \mathcal{N}$ is proper, closed, and convex. Additionally, for f_i of the form in (9), $f_i^2 : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable, and the gradient of f_i^2 is $L_{f_i^2}$-Lipschitz continuous, i.e., $\|\nabla f_i^2(x_i) - \nabla f_i^2(z_i)\| \le L_{f_i^2}\|x_i - z_i\|$ holds for any x_i, $z_i \in \mathbb{R}^d$ and some $L_{f_i^2} \ge 0$.
(b) $g_l : \mathbb{R}^{q_l} \to \mathbb{R}\cup\{+\infty\}$, $l \in \mathcal{Q}_\mathcal{G}$ is proper, closed, and convex. Additionally, for g_l of the form in (9), $g_l^2 : \mathbb{R}^{q_l} \to \mathbb{R}$ is convex and differentiable, and the gradient of g_l^2 is $L_{g_l^2}$-Lipschitz continuous with some $L_{g_l^2} \ge 0$.
(c) The Lagrangian $\mathcal{L}$ in (6) has a saddle point.
(d) $\bigcup_{l\in\mathcal{Q}_\mathcal{G}}\mathcal{C}_l = \mathcal{N}$ holds for $\mathcal{Q}_\mathcal{G}$.

First, we provide the following theorem. In this theorem, each agent can check the conditions in a distributed fashion. Regarding the choice of the parameters, a detailed discussion is available in Chapter 8 of [22].

Theorem 1: Consider Algorithms 1 and 2. Assume that Assumption 1 is satisfied. Assume that for all $i \in \mathcal{N}$, all $l \in \mathcal{Q}_\mathcal{G}$, and some $\varepsilon_l \in (0, 2-\phi_l)$, these inequalities hold:

$$\alpha_i^{-1} \ge \sum_{l\in\mathcal{Q}^i_\mathcal{G}}\gamma_l\lambda_{\max}(A_l^\top A_l) + L_{f_i^2}, \tag{11a}$$

$$\beta_l^{-1} - \gamma_l\lambda_{\max}(B_l^\top B_l) \ge 0, \tag{11b}$$

$$\gamma_l\left(1 - \frac{(1-\phi_l)^2}{2-\phi_l-\varepsilon_l}\right)B_l^\top B_l + Q_l \succeq 3L_{g_l^2}I_{q_l}, \tag{11c}$$

where $Q_l = \beta_l^{-1}I_{q_l} - \gamma_l B_l^\top B_l$. Then, $\lim_{k\to\infty}(F(\mathbf{x}^k) + G(\mathbf{y}^k)) = (F(\mathbf{x}^*) + G(\mathbf{y}^*))$ and $\lim_{k\to\infty}(\mathbf{AWx}^k + \mathbf{By}^k - \mathbf{c}) = 0$ hold, where $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*)$ is a saddle point of the Lagrangian function $\mathcal{L}$.

Proof: We prove Theorem 1 based on Theorem 6 in [22]. Note that the supporting lemmas for this proof are provided in the preprint [29] due to the space limit.

First, we give preliminaries of the proof. Let $F(\mathbf{x}) = F^1(\mathbf{x}) + F^2(\mathbf{x})$ and $G(\mathbf{y}) = G^1(\mathbf{y}) + G^2(\mathbf{y})$, where $F^1(\mathbf{x}) = \sum_{i\in\mathcal{N}} f_i^1(x_i)$, $F^2(\mathbf{x}) = \sum_{i\in\mathcal{N}} f_i^2(x_i)$, $G^1(\mathbf{y}) =$

$\sum_{l\in\mathcal{Q}_\mathcal{G}} g_l^1(y_l)$, and $G^2(\mathbf{y}) = \sum_{l\in\mathcal{Q}_\mathcal{G}} g_l^2(y_l)$. Now, we consider the following FLiP-ADMM algorithm:

$$\mathbf{x}^{k+1} \in \underset{\mathbf{x}}{\operatorname{argmin}}\{F^1(\mathbf{x}) + \langle \nabla F^2(\mathbf{x}^k) + \mathbf{W}^\top\mathbf{A}^\top\mathbf{u}^k, \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{AWx} + \mathbf{By}^k - \mathbf{c}\|_\mathbf{\Gamma}^2 + \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^k\right\|_\mathbf{P}^2\} \quad (12a)$$

$$\mathbf{y}^{k+1} \in \underset{\mathbf{y}}{\operatorname{argmin}}\{G^1(\mathbf{y}) + \langle \nabla G^2(\mathbf{y}^k) + \mathbf{B}^\top\mathbf{u}^k, \mathbf{y}\rangle + \frac{1}{2}\|\mathbf{AWx}^{k+1} + \mathbf{By} - \mathbf{c}\|_\mathbf{\Gamma}^2 + \frac{1}{2}\|\mathbf{y} - \mathbf{y}^k\|_\mathbf{Q}^2\} \quad (12b)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{\Phi\Gamma}(\mathbf{AWx}^{k+1} + \mathbf{By}^{k+1} - \mathbf{c}). \quad (12c)$$

Now, suppose $\mathbf{P}$ and $\mathbf{Q}$ are given as $\mathbf{P} = \mathbf{D}_\alpha^{-1} - \mathbf{W}^\top\mathbf{A}^\top\mathbf{\Gamma}\mathbf{AW}$ and $\mathbf{Q} = \text{blk-diag}([Q_l]_{l\in\mathcal{Q}_\mathcal{G}}) = \mathbf{D}_\beta^{-1} - \mathbf{B}^\top\mathbf{\Gamma}\mathbf{B}$, respectively. Then, (12) is equivalent to Algorithm 2. Additionally, when $F^2 = G^2 = 0$, (12) is reduced to Algorithm 1. Note that under Assumption 1a–b, the x_i- and y_l- subproblems in (3a), (10a), (3b), and (10b) are always well-defined (see Proposition 12.15 in [30]). Moreover, let $\mathbf{w}^k := [\mathbf{x}^{k\top}, \mathbf{y}^{k\top}, \mathbf{u}^{k\top}]^\top$ and $\mathbf{w}^* := [\mathbf{x}^{*\top}, \mathbf{y}^{*\top}, \mathbf{u}^{*\top}]^\top$. Furthermore, we define

$$M_0 = \frac{1}{2}\text{blk-diag}(\mathbf{P},\, \mathbf{B}^\top\mathbf{\Gamma}\mathbf{B} + \mathbf{Q},\, \mathbf{\Gamma}^{-1}\mathbf{\Phi}^{-1}),$$

$$M_1 = \frac{1}{2}\text{blk-diag}(0,\, \mathbf{L}_{G^2} + \mathbf{Q},\, \mathbf{\Theta}(\mathbf{\Phi}^{-1})^2\mathbf{\Gamma}^{-1}),$$

$$M_2 = \frac{1}{2}\text{blk-diag}(\mathbf{P} - \mathbf{L}_{F^2},\, \mathbf{B}^\top\mathbf{\Gamma}(I - \mathbf{\Theta}^{-1}(I-\mathbf{\Phi})^2)\mathbf{B} + \mathbf{Q} - 3\mathbf{L}_{G^2},\, (\mathbf{\Phi}^{-1})^2\mathbf{\Gamma}^{-1}(2I - \mathbf{\Phi} - \Theta)), \quad (13)$$

where $\mathbf{L}_{F^2} = \text{blk-diag}([L_{f_i^2} I_d]_{i\in\mathcal{N}})$, $\mathbf{L}_{G^2} = \text{blk-diag}([L_{g_l^2} I_{q_l}]_{l\in\mathcal{Q}_\mathcal{G}})$, and $\mathbf{\Theta} = \text{blk-diag}([\theta_l I_{p_l}])$ with $\theta_l = 2 - \phi_l - \varepsilon_l > 0$, $l \in \mathcal{Q}_\mathcal{G}$. Finally, we define the Lyapunov function V^k as

$$V^k = \|\mathbf{w}^k - \mathbf{w}^*\|_{M_0}^2 + \|\mathbf{w}^k - \mathbf{w}^{k-1}\|_{M_1}^2. \quad (14)$$

With this in mind, we prove Theorem 1 as follows. By the equation $\|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{M_0}^2 = \|\mathbf{w}^k - \mathbf{w}^*\|_{M_0}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_0}^2 + 2\langle \mathbf{w}^{k+1} - \mathbf{w}^k, \mathbf{w}^{k+1} - \mathbf{w}^*\rangle_{M_0}$, we have

$$V^{k+1} = V^k - \|\mathbf{w}^k - \mathbf{w}^{k-1}\|_{M_1}^2 + \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_1}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_0}^2 + 2\langle \mathbf{w}^{k+1} - \mathbf{w}^k, \mathbf{w}^{k+1} - \mathbf{w}^*\rangle_{M_0} \quad (15)$$

for V^k in (14). Then, Lemma 4 and Lemma 5 in [29] yield

$$\begin{aligned}
V^{k+1} &\le V^k - \|\mathbf{w}^k - \mathbf{w}^{k-1}\|_{M_1}^2 + \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_1}^2 \\
&- \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_0}^2 + 2\langle \mathbf{w}^{k+1} - \mathbf{w}^k, \mathbf{w}^{k+1} - \mathbf{w}^*\rangle_{M_0} \\
&+ \textstyle\sum_{i\in\mathcal{N}} \frac{L_{f_i^2}}{2}\|x_i^{k+1} - x_i^k\|^2 + \sum_{l\in\mathcal{Q}_\mathcal{G}} \frac{L_{g_l^2}}{2}\|y_l^{k+1} - y_l^k\|^2 \\
&- 2\langle \mathbf{w}^{k+1} - \mathbf{w}^k, \mathbf{w}^{k+1} - \mathbf{w}^*\rangle_{M_0} \\
&+ \|\mathbf{u}^{k+1} - \mathbf{u}^k\|_{(I-\mathbf{\Phi}^{-1})\mathbf{\Phi}^{-1}\mathbf{\Gamma}^{-1}}^2 \\
&+ \frac{1}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|_{\mathbf{L}_{G^2} - \mathbf{Q} + \mathbf{B}^\top\mathbf{\Theta}^{-1}(I-\mathbf{\Phi})^2\mathbf{\Gamma}\mathbf{B}}^2 \\
&+ \frac{1}{2}\|\mathbf{y}^k - \mathbf{y}^{k-1}\|_{\mathbf{L}_{G^2}+\mathbf{Q}}^2 + \frac{1}{2}\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_{\mathbf{\Theta}\mathbf{\Gamma}^{-1}(\mathbf{\Phi}^{-1})^2}^2 \\
&- \left(\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{u}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*)\right) \\
&= V^k - \|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_2}^2 \\
&\quad - \left(\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{u}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*)\right),
\end{aligned}$$

for any $\mathbf{\Theta} \succ 0$. Consequently, from this inequality, Lemmas 1 and 2 in [29] and $\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{u}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*) \ge 0$, we obtain $\lim_{k\to\infty}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_2}^2 = 0$ and $\lim_{k\to\infty}(\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{u}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*)) = 0$. Then, By $\|\mathbf{w}^{k+1} - \mathbf{w}^k\|_{M_2}^2 \to 0$, we have $\lim_{k\to\infty}(\mathbf{u}^{k+1} - \mathbf{u}^k) = 0$ Therefore, $\lim_{k\to\infty}(F(\mathbf{x}^k) + G(\mathbf{y}^k)) = (F(\mathbf{x}^*) + G(\mathbf{y}^*))$ and $\lim_{k\to\infty}(\mathbf{AWx}^k + \mathbf{By}^k - \mathbf{c}) = 0$ are achieved. ■

Moreover, when both $\mathbf{A}$ and $\mathbf{B}$ are the identity matrix, we can prove the convergence with fully agent-wise algorithmic parameters and convergence conditions. In this case, the agents can independently choose their algorithmic parameters without any cooperation.

Theorem 2: Assume that $\mathbf{A} = \mathbf{B} = I$. Consider that in Algorithms 1 and 2, we replace γ_l, ϕ_l, and β_l by the diagonal matrices $\text{diag}(\gamma_{\mathcal{C}_l}) \otimes I_d$ with $\gamma = [\gamma_1, \ldots, \gamma_n]^\top \in \mathbb{R}^n$, $\text{diag}(\phi_{\mathcal{C}_l}) \otimes I_d$ with $\phi = [\phi_1, \ldots, \phi_n]^\top \in \mathbb{R}^n$, and $\text{diag}(\beta_{\mathcal{C}_l}) \otimes I_d$ with $\beta = [\beta_1, \ldots, \beta_n]^\top \in \mathbb{R}^n$, respectively. Additionally, in (3b) and (10b), we replace $\text{prox}_{\beta_l g_l}(\cdot)$ with $\text{prox}_{g_l}^{(\text{diag}(\beta_{\mathcal{C}_l})\otimes I_d)^{-1}}(\cdot)$. Assume that Assumption 1 is satisfied. Assume that for all $i \in \mathcal{N}$ and some $\varepsilon_i \in (0, 2 - \phi_i)$, the following inequalities are satisfied:

$$\alpha_i^{-1} \ge \gamma_i|\mathcal{Q}_\mathcal{G}^i| + L_{f_i}, \quad \beta_i^{-1} - \gamma_i \ge 0,$$

$$\gamma_i\left(1 - \frac{(1-\phi_i)^2}{2 - \phi_i - \varepsilon_i}\right) I_d + Q_i \succeq 3\left(\max_{l\in\mathcal{Q}_\mathcal{G}^i} L_{g_l^2}\right) I_d,$$

where $Q_i = (\beta_i^{-1} - \gamma_i)I_d$. Then, $\lim_{k\to\infty}(F(\mathbf{x}^k) + G(\mathbf{y}^k)) = (F(\mathbf{x}^*) + G(\mathbf{y}^*))$ and $\lim_{k\to\infty}(\mathbf{Wx}^k - \mathbf{y}^k) = 0$ hold.

Proof: We can prove Theorem 2 similarly to Theorem 1 by modifying Lemma 5 in [29]. Due to the space limit, we omit the proof. ■

V. Application to Consensus Optimization

A. Problem Setting

Now, we consider the following problem:

$$\begin{aligned}
\underset{x_1\ldots,x_n\in\mathbb{R}^d}{\text{minimize}} \quad & \sum_{i\in\mathcal{N}}(f_i(x_i) + h_i(x_i)) \\
\text{subject to} \quad & x_i = x_j \quad \forall\{i,j\} \in \mathcal{E},
\end{aligned} \quad (16)$$

where f_i and h_i for $i \in \mathcal{N}$ are proper, closed, and convex. When $\mathcal{G}$ is connected, the problem (16) can be reformulated in the form of Problem (1) with $A_l = -B_l = I_{|\mathcal{C}_l|d}$, $c_l = 0$, and

$$g_l(y_l) = \sum_{j\in\mathcal{C}_l} \frac{1}{|\mathcal{Q}_\mathcal{G}^j|} h_j([y_l]_{\pi_l(j)}) + \delta_{\mathcal{D}_l}(y_l), \quad (17)$$

where $\mathcal{D}_l := \{y_l \in \mathbb{R}^{|\mathcal{C}_l|d} : \exists\xi \in \mathbb{R}^d \text{ s.t. } y_l = \mathbf{1}_{|\mathcal{C}_l|} \otimes \xi\}$ for $l \in \mathcal{Q}_\mathcal{G}$, and $\delta_{\mathcal{D}_l}(y_l)$ is the indicator function for $\mathcal{D}_l$, i.e., $\delta_{\mathcal{D}_l}$ satisfies $\delta_{\mathcal{D}_l}(y_l) = 0$ for $y_l \in \mathcal{D}_l$ and $\delta_{\mathcal{D}_l}(y_l) = \infty$ for $y_l \notin \mathcal{D}_l$. This can be verified by Proposition 4.2 in [9].

For the g_l in (17) and $\mathcal{D}_l$, the following proposition states that the proximal mapping of g_l in (17) is proximable if that of $(\sum_{j\in\mathcal{C}_l} \frac{1}{|\mathcal{Q}_\mathcal{G}^j|} h_j)(\cdot)$ is proximable.

Proposition 2: Assume that the functions $r_j : \mathbb{R}^d \to \mathbb{R}$, $j \in \mathcal{C}_l$ are proper, closed, and convex functions. Let

$\bar{r}_l : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and $s_l : \mathbb{R}^{|\mathcal{C}_l|d} \to \mathbb{R} \cup \{+\infty\}$ be $\bar{r}_l(z) = \sum_{j\in\mathcal{C}_l} r_j(z)$ for $z \in \mathbb{R}^d$ and

$$s_l(x_{\mathcal{C}_l}) = \sum_{j\in\mathcal{C}_l} r_j(x_j) + \delta_{\mathcal{D}_l}(x_{\mathcal{C}_l}),$$

respectively. Then,

$$\mathrm{prox}_{s_l}(x_{\mathcal{C}_l}) = \mathbf{1}_{|\mathcal{C}_l|} \otimes \mathrm{prox}_{\frac{1}{|\mathcal{C}_l|}\bar{r}_l}\left(\frac{1}{|\mathcal{C}_l|}(\mathbf{1}_{|\mathcal{C}_l|} \otimes I_d)^\top x_{\mathcal{C}_l}\right)$$

holds for $\bar{r}_l$ and any $x_{\mathcal{C}_l} \in \mathbb{R}^{|\mathcal{C}_l|d}$.

Proof: From the definition of $\mathcal{D}_l$, we have $\mathrm{prox}_{s_l}(x_{\mathcal{C}_l}) = \arg\min_{v\in\mathbb{R}^{|\mathcal{C}_l|d}}\{\frac{1}{2}\|x_{\mathcal{C}_l} - z\|^2 + s_l(v)\} = \arg\min_{\xi\in\mathbb{R}^d}\{\sum_{j\in\mathcal{C}_l}\frac{1}{2}\|x_j - \xi\|^2 + \bar{r}_l(\xi)\}$. Then, we obtain $\xi^* \in \arg\min_{\xi\in\mathbb{R}^d}\{\sum_{j\in\mathcal{C}_l}\frac{1}{2}\|x_j - \xi\|^2 + \bar{r}_l(\xi)\} \Leftrightarrow 0 \in |\mathcal{C}_l|\{\xi^* - \frac{1}{|\mathcal{C}_l|}\sum_{j\in\mathcal{C}_l} x_j + \frac{1}{|\mathcal{C}_l|}\partial\bar{r}_l(\xi^*)\} \Leftrightarrow \frac{1}{|\mathcal{C}_l|}\sum_{j\in\mathcal{C}_l} x_j \in (\mathbb{I} + \frac{1}{|\mathcal{C}_l|}\partial\bar{r}_l)(\xi^*) \Leftrightarrow \xi^* \in (\mathbb{I} + \frac{1}{|\mathcal{C}_l|}\partial\bar{r}_l)^{-1}(\frac{1}{|\mathcal{C}_l|}\sum_{j\in\mathcal{C}_l} x_j)$, where $\mathbb{I}$ is the identity operator. Therefore, Proposition 2 follows from Proposition 16.44 in [30]. ■

B. Numerical Experiment

Through a numerical experiment, we demonstrate the effectiveness of the proposed CL-ADMM and CL-FLiP-ADMM.

Consider a multi-agent system with $n = 50$ agents. Assume that the communication network $\mathcal{G}$ is given as a connected time-invariant undirected graph, where each edge is generated with a probability of 0.1. We consider that the multi-agent system solves the consensus optimization problem in (16) with

$$f_i(x_i) = \frac{1}{2}\|\Psi_i x_i - b_i\|^2, \quad h_i(x_i) = \lambda_i\|x_i\|_1, \tag{18}$$

where $\Psi_i = I_d + 0.1\Omega_i \in \mathbb{R}^{d\times d}$, $b_i \in \mathbb{R}^d$, $i \in \mathcal{N}$, and $\lambda_i = \lambda = 0.001$ for all $i \in \mathcal{N}$. For all $i \in \mathcal{N}$, each entry of Ω_i and b_i is generated by the standard normal distribution. Note that f_i in (18) is $\lambda_{\max}(\Psi_i^\top\Psi_i)$-smooth, and the functions g_l in (17) is proximable for h_i in (18).

Now, to verify the effectiveness of the proposed clique-based algorithms, we conduct simulations for the CL-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$, CL-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{E}$, CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$, CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{E}$, and PG-EXTRA [2], given as follows:

$$\mathbf{x}^{k+1} = \mathrm{prox}_{\eta\lambda\|\cdot\|_1}((W_\mathrm{m} \otimes I_d)\mathbf{x}^k - \eta\nabla F(\mathbf{x}^k) - \mathbf{v}^k)$$
$$\mathbf{v}^{k+1} = \mathbf{v}^k + \frac{I_{nd} - W_\mathrm{m} \otimes I_d}{2}\mathbf{x}^k,$$

where $W_\mathrm{m} \in \mathbb{R}^{n\times n}$ is a mixing matrix of $\mathcal{G}$. Note that when $\mathcal{G}$ is connected, $\cap_{l\in\mathcal{Q}_\mathcal{G}}\{\mathbf{x} \in \mathbb{R}^{nd} : x_{\mathcal{C}_l} \in \mathcal{D}_l\} = \{\mathbf{x} \in \mathbb{R}^{nd} : x_1 = \cdots = x_n\}$ is satisfied for $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ from Proposition 4.2 in [9], and hence an optimal solution can be obtained by the CL-ADMM and CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$.

The algorithmic parameters are given as follows. For the CL-ADMM algorithms with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ and $\mathcal{E}$, we set $\gamma_l = \phi_l = \beta_l = 1$ for all $l \in \mathcal{Q}_\mathcal{G}$, and $\alpha_i = 1/|\mathcal{Q}_\mathcal{G}^i|$ for each $i \in \mathcal{N}$. For the CL-FLiP-ADMM algorithms with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ and $\mathcal{E}$, we set $f_i^1(x_i) = 0$, $f_i^2(x_i) = f_i(x_i)$, $g_l^1(y_l) = g_l(y_l)$, $g_l^2(y_l) = 0$, $\gamma_l = \phi_l = \beta_l = 1$ for all $l \in \mathcal{Q}_\mathcal{G}$, and $\alpha_i = 1/(\lambda_{\max}(\Psi_i^\top\Psi_i) + |\mathcal{Q}_\mathcal{G}^i|)$ for each $i \in \mathcal{N}$. For the PG-EXTRA, we set $\eta = 0.9(1 + \lambda_{\min}(W_\mathrm{m}))/\lambda_{\max}(\mathbf{\Psi}^\top\mathbf{\Psi})$ with $\mathbf{\Psi} = \text{blk-diag}(\Psi_1, \ldots, \Psi_n)$ and $W_\mathrm{m} = I_n - \frac{1}{\max_{i\in\mathcal{N}}|\mathcal{N}_i| - 1}L_\mathcal{G}$, where $L_\mathcal{G}$ is the graph Laplacian matrix of the graph $\mathcal{G}$.

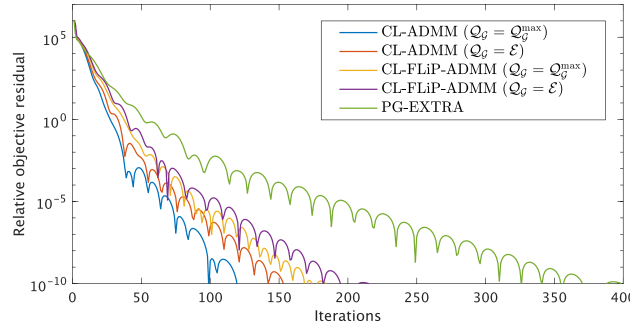


Fig. 2: Plot of relative objective residuals $|(F(\mathbf{x}^k)+G(\mathbf{x}^k)) - (F(\mathbf{x}^*) - G(\mathbf{x}^*))|/(F(\mathbf{x}^*) + G(\mathbf{x}^*))$ against the number of iteration under the CL-ADMM and CL-FLiP-ADMM algorithms with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ and $\mathcal{E}$, and PG-EXTRA.

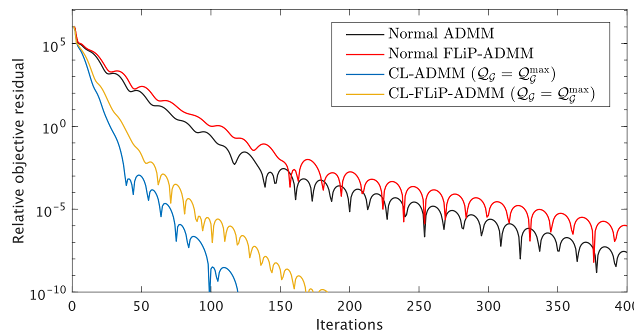


Fig. 3: Plot of relative objective residuals $|(F(\mathbf{x}^k)+G(\mathbf{x}^k)) - (F(\mathbf{x}^*) + G(\mathbf{x}^*))|/(F(\mathbf{x}^*) + G(\mathbf{x}^*))$ against the number of iteration under the normal ADMM, normal FLiP-ADMM, and the CL-ADMM and CL-FLiP-ADMM algorithms with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$.

Fig. 2 plots the relative objective residuals $|(F(\mathbf{x}^k) + G(\mathbf{x}^k)) - (F(\mathbf{x}^*) + G(\mathbf{x}^*))|/(F(\mathbf{x}^*) + G(\mathbf{x}^*))$ versus iterations. As shown in Fig. 2, all the methods successfully converge to an optimal solution with a tiny error, and the CL-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ outpaces the others. Additionally, although the CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ is slower than the CL-ADMM algorithms, it outperforms the CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{E}$ and PG-EXTRA. Moreover, Fig. 2 indicates that handling the consensus constraint on a clique basis, i.e., setting not $\mathcal{Q}_\mathcal{G} = \mathcal{E}$ but $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ can enhance the performance.

For further comparison, we also run the normal ADMM and FLiP-ADMM algorithms on the same problem, and Fig. 3 plots the relative objective residual versus iterations of the normal ADMM and normal FLiP-ADMM with the results of the CL-ADMM and CL-FLiP-ADMM with $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$ in Fig. 2. Here, the normal ADMM and FLiP-ADMM correspond to special cases of Algorithms 1 and 2 in which all the parameters γ_l, ϕ_l, β_l, and α_i are common among all $l \in \mathcal{Q}_\mathcal{G}$ and all $i \in \mathcal{N}$. For the normal ADMM, we set $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$, $\gamma_l = \phi_l = \beta_l = 1$ for all $l \in \mathcal{Q}_\mathcal{G}$, and $\alpha_i = 1/\max_{i\in\mathcal{N}} |\mathcal{Q}_\mathcal{G}^i|$ for all $i \in \mathcal{N}$. For the normal FLiP-ADMM, we set $\mathcal{Q}_\mathcal{G} = \mathcal{Q}_\mathcal{G}^{\max}$, $\gamma_l = \phi_l = \beta_l = 1$ for all $l \in \mathcal{Q}_\mathcal{G}$, and $\alpha_i = 1/\max_{i\in\mathcal{N}}(\lambda_{\max}(\Psi_i^\top \Psi_i) + |\mathcal{Q}_\mathcal{G}^i|)$ for all $i \in \mathcal{N}$. From Fig. 3, the normal ADMM and normal FLiP-ADMM are much slower than the CL-ADMM and CL-FLiP-ADMM. This implies that thanks to the localized algorithmic parameters, the proposed methods are not only fully distributed but perform better than the normal ADMM and FLiP-ADMM.

These results highlight the effectiveness of the proposed methods.

Remark 3: The clique-wise handling of pairwise constraints (e.g., the consensus constraint in (16)) tends to outperform the pairwise one, in particular, when G is not dense, and the initial value of $\mathbf{x}^k$ is far from its optimal value. When the clique-wise handling does not outperform the edge-based one, the clique-wise coupled framework is still meaningful because the agents can share a part of their computation (see Remark 1).

VI. Conclusion

This paper addressed a novel framework for distributed optimization, clique-wise coupled optimization problems. We proposed a distributed ADMM and FLiP-ADMM algorithms based on cliques and proved the convergence theorems with no global parameter. Moreover, we applied the proposed methods to a consensus optimization problem and demonstrated their effectiveness via numerical experiments. A future direction is to extend existing methods for pairwise coupled distributed optimization/control problems to a more general framework from the viewpoint of clique-wise coupling.

Appendix

References

[1] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automat. Contr.*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6013–6023, Nov. 2015.

[3] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Apr. 2017.

[4] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4494–4506, Jan. 2019.

[5] P. Latafat, N. M. Freris, and P. Patrinos, "A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization," *IEEE Trans. Automat. Contr.*, vol. 64, no. 10, pp. 4050–4065, Oct. 2019.

[6] H. Li, E. Su, C. Wang, J. Liu, Z. Zheng, Z. Wang, and D. Xia, "A primal-dual forward-backward splitting algorithm for distributed convex optimization," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 1, pp. 278–284, Feb. 2023.

[7] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[8] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton Univ. Press, 2009.

[9] K. Sakurama and T. Sugie, "Generalized coordination of multi-robot systems," *Found. Trends® Syst. Control*, vol. 9, no. 1, pp. 1–170, 2021.

[10] B. Bollobás, *Modern Graph Theory*. Springer Science & Business Media, 1998.

[11] D. P. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall Englewood Cliffs, NJ, 1989.

[12] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3047–3064, Jan. 2012.

[13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, Jun. 2011.

[14] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1, pp. 259–268, Nov. 1992.

[15] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proc. Vol.*, vol. 45, no. 16, pp. 83–88, Mar. 2012.

[16] D. Hallac, J. Leskovec, and S. Boyd, "Network Lasso: Clustering and optimization in large graphs," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 387–396.

[17] Y. Zhang and Q. Yang, "An overview of multi-task learning," *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, Sep. 2017.

[18] Y. Watanabe and K. Sakurama, "Accelerated distributed projected gradient descent for convex optimization with clique-wise coupled constraints," in *Proc. 22nd IFAC World Congr.*, 2023, (Accepted).

[19] L. Vandenberghe, M. S. Andersen *et al.*, "Chordal graphs and semidefinite optimization," *Foundations and Trends® in Optimization*, vol. 1, no. 4, pp. 241–433, 2015.

[20] Y. Zheng, M. Kamgarpour, A. Sootla, and A. Papachristodoulou, "Distributed design for decentralized control using chordal decomposition and admm," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 614–626, 2019.

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

[22] E. K. Ryu and W. Yin, *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators*. Cambridge Univ. Press, 2022.

[23] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3719–3734, Jul. 2016.

[24] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, Jul. 2020.

[25] I. Notarnicola and G. Notarstefano, "Constraint-coupled distributed optimization: A relaxation and duality approach," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 483–492, Mar. 2020.

[26] X. Wu, H. Wang, and J. Lu, "Distributed optimization with coupling constraints," *IEEE Trans. Automat. Contr.*, vol. 68, no. 3, pp. 1847–1854, Mar. 2023.

[27] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, Aug. 2013.

[28] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, no. 3, pp. 667–681, Apr. 2013.

[29] Y. Watanabe and K. Sakurama, "Distributed optimization of clique-wise coupled problems," *arXiv preprint arXiv:2304.10904*, 2023.

[30] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2011.