A Primal Decomposition Approach to Globally Coupled Aggregative Optimization over Networks

Yuanhanqing Huang and Jianghai Hu

***Abstract*— We consider a class of multi-agent optimization problems, where each agent has a local objective function that depends on its own decision variables and the aggregate of others, and is willing to cooperate with other agents to minimize the sum of the local objectives. After associating each agent with an auxiliary variable and the related local estimates, we conduct primal decomposition to the globally coupled problem and reformulate it so that it can be solved distributedly. Based on the Douglas-Rachford method, an algorithm is proposed which ensures the exact convergence to a solution of the original problem. The proposed method enjoys desirable scalability by only requiring each agent to keep local estimates whose number grows linearly with the number of its neighbors. We illustrate our proposed algorithm by numerical simulations on a commodity distribution problem over a transport network.**

I. Introduction

In a cooperative multi-agent system, there exists a group of agents each of whom has a specific objective function depending on the joint decision profile of all agents, and they cooperate with each other to optimize the sum of their local objectives. Over the past decade, considerable attention and effort have been paid to the consensus optimization problem [1], [2]. There is also some existing work where each agent keeps its own distinct decision variables [3], [4]. In this paper, we restrict our attention to a special case where the influences of other agents' strategies can be represented through some aggregative coupling structures [5]. The aggregative coupling structures have been used to model numerous applications, e.g., network congestion control [6], demand side management in smart grids [7], and charging control of electric vehicles [8]. Besides the aggregative coupling in the objective functions, in many circumstances, the decisions of the agents may be subject to some global resource constraints [9], [10], such as total energy and communication channel capacity [11]. The coupled objectives and strategy sets of the agents are at odds with local privacy concerns and limited scalability. Thus, distributed algorithms are preferred to solve such problems which only allow local exchanges of information.

Our proposed solution to the above problem is inspired by some recent work in non-cooperative games on networks, i.e., the generalized Nash equilibrium problem (GNEP) [12], which has attracted increasing research interest, especially through the avenue of operator splitting [13], [14], [15]. For example, the algorithms proposed in [16], [17] carry out multiple rounds of communication within each iteration. With sufficient rounds of information exchange, the proposed algorithms can converge to an ϵ-neighborhood of a generalized Nash equilibrium (GNE). The authors of [18] design a continuous-time algorithm based on the projected dynamics and non-smooth tracking strategy, which is only applicable if the coupling constraints can be expressed as a system of linear equations. More recently, [19], [20] introduce local estimates of the aggregates of interest, and then leverage the forward-backward splitting and proximal-point algorithms to compute the GNEs, respectively. The authors of [21] further develop an algorithm that can deal with time-varying communication networks by integrating the projected pseudo-gradient scheme with dynamic tracking. The convergence of these methods relies on a proper initialization and an invariance property throughout the algorithm iterations to ensure valid estimates, which make them vulnerable to system noise and malicious attacks.

In this paper, we consider cooperative multi-agent optimization problems that are globally coupled by some aggregates as well as some affine global constraints. We focus on the cases where the aggregates in objectives and global constraints share the same linear functional form. Our main contributions are as follows: a) we present a primal decomposition scheme that converts the original globally coupled problem into local problems among individual agents, subject to some consensus constraints. We show that we can find a minimizer of the original problem by computing a zero of an operator derived from the decomposed problem; b) we use the Douglas-Rachford (DR) splitting method to develop a distributed algorithm for computing a zero of the previously derived operator. The exact convergence of the algorithm can be established without the need for the invariance property.

Basic Notations: For a set of matrices $\{V_i\}_{i\in S}$, we let $\mathrm{blkd}(V_1,\dots,V_{|S|})$ or $\mathrm{blkd}(V_i)_{i\in S}$ denote the diagonal concatenation of these matrices, $[V_1,\dots,V_{|S|}]$ their horizontal stack, and $[V_1;\cdots;V_{|S|}]$ their vertical stack. For a set of vectors $\{v_i\}_{i\in S}$, $[v_i]_{i\in S}$ or $[v_1;\cdots;v_{|S|}]$ denotes their vertical stack. For a vector v and a positive integer i, $[v]_i$ denotes the ith entry of v. Denote $\overline{\mathbb{R}} \coloneqq \mathbb{R}\cup\{+\infty\}$, $\mathbb{R}_+ \coloneqq [0,+\infty)$, and $\mathbb{R}_{++} \coloneqq (0,+\infty)$. $\mathbb{S}^n_+$ (resp. S^n_{++}) represents the set of all $n\times n$ symmetric positive semi-definite (resp. definite) matrices. $\iota_{\mathcal{S}}(x)$ is defined to be the indicator function of a set $\mathcal{S}$, i.e., if $x\in\mathcal{S}$, then $\iota_{\mathcal{S}}(x)=0$; otherwise, $\iota_{\mathcal{S}}(x)=+\infty$. $N_S(x)$ denotes the normal cone to the set $S\subseteq\mathbb{R}^n$ at the point x: if $x\in S$, then $N_S(x)\coloneqq\{u\in\mathbb{R}^n \mid \sup_{z\in S}\langle u, z-x\rangle\le 0\}$; otherwise, $N_S(x)\coloneqq\varnothing$. We use $\rightrightarrows$ to indicate a point-to-set map. For an operator $T:\mathbb{R}^n\rightrightarrows\mathbb{R}^n$, $\mathrm{Zer}(T)\coloneqq\{x\in\mathbb{R}^n \mid$

This work was supported by the National Science Foundation under Grant No. 2014816.

The authors are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907, USA {huan1282, jianghai}@purdue.edu

$Tx \ni \mathbf{0}\}$ and $\mathrm{Fix}(T) := \{x \in \mathbb{R}^n \mid Tx \ni x\}$ denote its zero set and fixed point set, respectively.

II. Problem Formulation And Preliminaries

A. Cooperative Multi-Agent Optimization Problem

We consider a group of agents indexed by $\mathcal{N} = \{1, \dots, N\}$, where each agent $i \in \mathcal{N}$ shall choose its decision variables x_i from its local feasible set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$. The goal of each agent i is to minimize its objective $J_i(x_i, s_i(x_{-i}))$, which depends both on its own local decision x_i and the aggregate of other agents' decisions $s_i(x_{-i})$. We let the vector x_{-i} represent the vertical stack of other agents' decisions. The aggregate $s_i : \mathbb{R}^{n_{-i}} \to \mathbb{R}^l$ is assumed to be of the form $s_i(x_{-i}) := \sum_{j \in \mathcal{N}_{-i}} A_j x_j$, where $A_j \in \mathbb{R}^{l \times n_j}$, $n_{-i} := \sum_{j \in \mathcal{N}_{-i}} n_j$, and $\mathcal{N}_{-i}$ denotes the set of all agents except i. Besides the local feasible sets, the decisions of all agents should also satisfy the global constraints given by $\tilde{\mathcal{X}} := \{x \in \mathcal{X} \mid \sum_{i \in \mathcal{N}} A_i x_i \le c\}$, where $x := [x_1; \cdots; x_N]$, $\mathcal{X} := \prod_{i \in \mathcal{N}} \mathcal{X}_i$, and $c \in \mathbb{R}^l$ is a constant vector denoting the total availability of l global resources shared among all involved agents.

The setting above gives rise to the specific formulations of the multi-agent optimization problem we are going to study. In this problem, this group of agents aim to cooperatively solve the following convex optimization problem:

$$\begin{cases} \underset{x_i \in \mathcal{X}_i, i \in \mathcal{N}}{\text{minimize}} & J(x) := \sum_{i \in \mathcal{N}} J_i(x_i, s_i(x_{-i})) \\ \text{subject to} & \sum_{i \in \mathcal{N}} A_i x_i \le c. \end{cases} \tag{1}$$

Assumption 1. *(Existence of Subgradient) For each agent $i \in \mathcal{N}$, $J_i(x_i, s_i) : \mathcal{X}_i \times \mathbb{R}^l \to \overline{\mathbb{R}}$ is an extended-real-valued closed convex proper (CCP) function in x_i and s_i.*

Assumption 2. *(Feasible Sets) Each local feasible set $\mathcal{X}_i$ is nonempty, closed and convex. The set $\tilde{\mathcal{X}}$ is nonempty and satisfies Slater's constraint qualification [22, Sec. 3.2].*

Under Assumptions 1 and 2, the problem (1) admits a nonempty minimizer set if and only if there exists a multiplier $\lambda \in \mathbb{R}^l$ such that the Karush-Kuhn-Tucker (KKT) system below holds [22, Sec. 1.3]:

$$\begin{aligned} &\partial_{x_i} J(x) + A_i^T \lambda + N_{\mathcal{X}_i}(x_i) \ni \mathbf{0}, \ \forall i \in \mathcal{N}, \\ &\mathbf{0} \le \lambda \perp c - \textstyle\sum_{i \in \mathcal{N}} A_i x_i \ge \mathbf{0}, \end{aligned} \tag{2}$$

where the subgradient can be more explicitly written as $\partial_{x_i} J(x) = \partial_{x_i} J_i(x_i, s_i(x_{-i})) + \sum_{j \in \mathcal{N}_{-i}} A_i^T \partial_{s_j} J_j(x_j, s_j(x_{-j}))$.

Assumption 3. *The set of minimizers of (1), i.e., $\{x \mid J(x) < +\infty, J(x) \le J(x'), \forall x' \in \tilde{\mathcal{X}}\}$, is nonempty.*

A (typically sparse) communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is assumed to exist to implement local information exchanges, where $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of directed edges. Let E be the cardinality of $\mathcal{E}$. We denote by (i, j) a directed edge with agent i as its tail and agent j as its head. Each agent can communicate with its neighbors through arbitrators on the incident edges and then update its local decision variables accordingly. For brevity of notation, define the sets of agent i's in- and out-neighbors as $\mathcal{N}_i^+ := \{j \in \mathcal{N} \mid (j, i) \in \mathcal{E}\}$ and $\mathcal{N}_i^- := \{j \in \mathcal{N} \mid (i, j) \in \mathcal{E}\}$, the cardinalities of which are denoted by N_i^+ and N_i^-, respectively. Note that although the multipliers we are going to introduce are defined in a directed fashion, we assume each node can send messages to both its in- and out-neighbors, and $\mathcal{G}$ should satisfy the following assumption.

Assumption 4. *(Communicability) The underlying communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is undirected and connected. Besides, it has no self-loops and $\forall i \in \mathcal{N}$, $\mathcal{N}_i^- \neq \emptyset$.*

B. The Douglas-Rachford (DR) Splitting Method

A set-valued operator $\mathcal{T} : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is called maximally monotone if $\forall (x, u) \in \mathrm{gra}(\mathcal{T})$ and $\forall (y, v) \in \mathrm{gra}(\mathcal{T})$, $\langle x - y, u - v \rangle \ge 0$, and its graph is not properly contained in the graph of any other monotone operators. By leveraging the resolvent of $\mathcal{T}$, i.e., $J_{\mathcal{T}} := (I + \mathcal{T})^{-1}$, the proximal point iteration can generate a sequence converging to a zero of $\mathcal{T}$ [23, Sec. 23.4]. However, this method has a drawback that the resolvent of $\mathcal{T}$ often can not be easily evaluated. Another potential difficulty is that in a network with multiple agents, $J_{\mathcal{T}}$ usually cannot be implemented distributedly.

The DR splitting method decomposes the operator $\mathcal{T}$ as the sum of two other operators $\mathcal{A}$ and $\mathcal{B}$, whose resolvents are easier to evaluate. This can thus alleviate the computational burden by evaluating $J_{\mathcal{A}} := (I + \mathcal{A})^{-1}$ and $J_{\mathcal{B}} := (I + \mathcal{B})^{-1}$ instead. Moreover, if $\mathrm{Zer}(\mathcal{T}) \neq \emptyset$, the convergence of the DR method can be obtained under very mild assumptions, i.e., $\mathcal{A}$ and $\mathcal{B}$ are set-valued and maximally monotone [23, Sec. 26.3].

A common strategy to solve networked problems is that, rather than focusing on $\mathcal{T}(\psi) \ni \mathbf{0}$, we consider $\Phi^{-1}\mathcal{T}(\psi) \ni \mathbf{0}$. Here, Φ is a positive definite matrix, called the design matrix, which is introduced to facilitate the distributed implementation. By applying the DR method to the splitting $\Phi^{-1}\mathcal{A} + \Phi^{-1}\mathcal{B}$, we obtain the following updating steps:

$$\begin{aligned} &\text{Calculate } J_{\Phi^{-1}\mathcal{A}} : \psi^{(k+1)} := J_{\Phi^{-1}\mathcal{A}}(\tilde{\psi}^{(k)}); \\ &\text{R-R updates} : \hat{\psi}^{(k+1)} := 2 \cdot \psi^{(k+1)} - \tilde{\psi}^{(k)}; \\ &\text{Calculate } J_{\Phi^{-1}\mathcal{B}} : \bar{\psi}^{(k+1)} := J_{\Phi^{-1}\mathcal{B}}(\hat{\psi}^{(k+1)}); \\ &\text{K-M updates} : \tilde{\psi}^{(k+1)} := \tilde{\psi}^{(k)} + 2\gamma^{(k)}(\bar{\psi}^{(k+1)} - \psi^{(k+1)}). \end{aligned} \tag{3}$$

In addition to the aforementioned conditions, if $(\gamma^{(k)})_{k \in \mathbb{N}}$ is a sequence in $[0, 1]$ satisfying $\sum_{k \in \mathbb{N}} \gamma^{(k)}(1 - \gamma^{(k)}) = +\infty$, then the sequence $(\psi^{(k)})_{k \in \mathbb{N}}$ generated by (3) will converge to a zero of $\mathcal{T}$ [23, Thm. 26.11]. The second and fourth steps in (3) are trivial and the major workload resides in the first and third steps. We use $\hat{\psi}^{(k)}$ to denote the results of the reflected resolvent (R-R) updates $R_{\Phi^{-1}\mathcal{A}}(\tilde{\psi}^{(k)}) = 2\psi^{(k)} - \tilde{\psi}^{(k)}$ and $\tilde{\psi}^{(k+1)}$ the results of the Krasnoselskij-Mann (K-M) updates $\tilde{\psi}^{(k)} + 2\gamma^{(k)}(\bar{\psi}^{(k+1)} - \psi^{(k+1)})$. This set of notations is used throughout for brevity, and similar notations are defined for decisions $x^{(k)}$ $(\hat{x}^{(k)}, \bar{x}^{(k)}, \tilde{x}^{(k)})$, multipliers $\lambda^{(k)}$ $(\hat{\lambda}^{(k)}, \bar{\lambda}^{(k)}, \tilde{\lambda}^{(k)})$, etc.

III. Distributed Algorithm Using Primal Decomposition

A. Primal Decomposition of the Problem

Given the multi-agent optimization problem (1), we introduce for each agent i a variable $\sigma_i := \sum_{j \in \mathcal{N}_{-i}} A_j x_j \in \mathbb{R}^l$ to

track the aggregate of other agents. The global problem (1) can be equivalently recast into the following problems:

$$(\forall i \in \mathcal{N}) \begin{cases} \underset{x_i \in X_i, \sigma_i}{\text{minimize}} & J_i(x_i, \sigma_i) \\ \text{subject to} & A_i x_i + \sigma_i \le c; \end{cases} \tag{4}$$
$$\text{all agents collectively satisfy: } \sigma_i = \textstyle\sum_{j \in \mathcal{N}_{-i}} A_j x_j, \forall i \in \mathcal{N}.$$

We refer the readers to [24, Lemma 1] for a detailed discussion about the equivalence between (1) and (4). However, the current formulation of σ_i is still globally dependent, while we need to estimate it with only local communications. To this end, we associate a weight matrix $W \in \mathbb{R}^{N\times N}$ with $\mathcal{G}$ that satisfies the conditions in Assumption 4. If there is a directed edge from agent i to agent j ($j \neq i$), W_{ij} is assigned some proper negative value; otherwise, $W_{ij} = 0$. The diagonal entries are set to be $W_{ii} = -\sum_{j\in\mathcal{N}} W_{ij} = -\sum_{j\in\mathcal{N}_i^-} W_{ij} > 0$. By Assumption 4, the defined weight matrix W has an eigenvalue 0 with multiplicity 1, with an associated eigenvector $\mathbf{1}_N$. Furthermore, we endow each agent i with an auxiliary variable $y_i \in \mathbb{R}^l$. With the introduction of $\{y_i\}_{i\in\mathcal{N}}$ and weight matrix W, we can decompose the problem (4) as follows:

$$\begin{cases} \underset{x_i \in X_i, \sigma_i, y_i}{\text{minimize}} \; J_i(x_i, \sigma_i) \\ \text{subject to} \;\; A_i x_i + \sigma_i \le c \\ \qquad (N-1)A_i x_i - \sigma_i = W_{ii} y_i + \textstyle\sum_{j\in\mathcal{N}_i^+} W_{ji} y_j \\ \qquad A_i x_i + \sigma_i = A_j x_j + \sigma_j, \forall j \in \mathcal{N}_i^+. \end{cases} \tag{5}$$

A proof of the equivalence between (4) and (5) can be found in [24, Appendix B]. Besides its own variables, the problem (5) of agent i only involves the auxiliary variables of its in-neighbors. Hence, (5) is a locally dependent problem. To facilitate the distributed implementation, each agent i is assumed to keep a local estimate for the auxiliary variable of each of its in-neighbors. Let y_{ji} denote the local estimate of y_j kept by agent i. With these local estimates, we can rewrite the preceding problem (5) as:

$$\begin{cases} \underset{x_i \in X_i, \sigma_i, \boldsymbol{y}_i}{\text{minimize}} \; J_i(x_i, \sigma_i) \\ \text{subject to} \;\; A_i x_i + \sigma_i \le c \\ \qquad (N-1)A_i x_i - \sigma_i = W_{ii} y_i + \textstyle\sum_{j\in\mathcal{N}_i^+} W_{ji} y_{ji} \\ \qquad A_i x_i + \sigma_i = A_j x_j + \sigma_j, \;\; y_{ji} = y_j, \forall j \in \mathcal{N}_i^+ \end{cases} \tag{6}$$

where $\boldsymbol{y}_i := [y_i; [y_{ji}]_{j\in\mathcal{N}_i^+}]$. By this reformulation, the coupling with other agents is sandboxed inside the consensus constraints between each auxiliary variable y_i and the local estimate y_{ij} kept by each out-neighbor j. As will be shown later, these consensus constraints can be enforced by some simple updating steps, making (6) more tractable than the one in (5). It is worth highlighting that the local variables kept by each agent i has dimension $n_i + l(2+N_i^+)$ which merely grows linearly with the number of its in-neighbors and allows better scalability in a sparsely connected communication network.

To study agent i's local optimization problem (6), we consider its associated Lagrangian $\mathcal{L}_i$ defined as follows:

$$\begin{aligned} \mathcal{L}_i &= J_i(x_i, \sigma_i) + \iota_{X_i}(x_i) + \iota_{Q_i}(x_i, \sigma_i) + \iota_{\mathcal{F}_i}(x_i, \sigma_i, \boldsymbol{y}_i) \\ &\quad + \textstyle\sum_{j\in\mathcal{N}_i^+} \big(\lambda_{ji}^T (A_i x_i + \sigma_i - A_j x_j - \sigma_j) + \mu_{ji}^T (y_{ji} - y_j)\big), \end{aligned}$$

where λ_{ji} is the Lagrange multiplier of the constraint $A_i x_i + \sigma_i - A_j x_j - \sigma_j = 0$; μ_{ji} is the Lagrange multiplier incentivizing the consensus between y_{ji} and y_j; $Q_i := \{(x_i, \sigma_i) \mid A_i x_i + \sigma_i \le c\}$ denotes the local feasible set corresponding to the first constraint of (6); and $\mathcal{F}_i := \{(x_i, \sigma_i, \boldsymbol{y}_i) \mid (N-1)A_i x_i - \sigma_i = W_{ii} y_i + \sum_{j\in\mathcal{N}_i^+} W_{ji} y_{ji}\}$ denotes the local feasible set corresponding to the second constraint of (6).

Summing the Lagrangians of all agents, we obtain the Lagrangian for the network optimization problem given by:

$$\begin{aligned} \mathcal{L}_{\text{net}} &= \sum_{i\in\mathcal{N}} \big(J_i(x_i, \sigma_i) + \iota_{X_i}(x_i) + \iota_{Q_i}(x_i, \sigma_i) + \iota_{\mathcal{F}_i}(x_i, \sigma_i, \boldsymbol{y}_i)\big) \\ &\quad + \sum_{(j,i)\in\mathcal{E}} \big(\lambda_{ji}^T (A_i x_i + \sigma_i - A_j x_j - \sigma_j) + \mu_{ji}^T (y_{ji} - y_j)\big). \end{aligned} \tag{7}$$

As will be shown later in the paper, by finding a saddle point of the network Lagrangian, we can obtain a solution of the original problem (1). For convenience, we shall write $\{\boldsymbol{y}_i\}$ in replacement of the more cumbersome notation $\{\boldsymbol{y}_i\}_{i\in\mathcal{N}}$ and similarly for other variables on nodes and edges (e.g. $\{\mu_{ji}\}$ in replacement of $\{\mu_{ji}\}_{(j,i)\in\mathcal{E}_g}$), unless otherwise specified.

Note that the structure of $\sum_{(j,i)\in\mathcal{E}} \mu_{ji}^T (y_{ji} - y_j)$ in the network Lagrangian defined above, allows us to organize $\{\boldsymbol{y}_i\}$ and $\{\mu_{ji}\}$ into a single vector $\omega := [\omega_1; \cdots; \omega_i; \cdots; \omega_N]$ with $\omega_i := [y_i; \mu_{ij_1}; y_{ij_1}; \cdots; \mu_{ij_{N_i^-}}; y_{ij_{N_i^-}}]$, and construct a constant symmetric matrix $M_y := \text{blkd}(M_{y_i})_{i\in\mathcal{N}} \in \mathbb{R}^{(2E+N)\times(2E+N)}$ accordingly such that the consensus term can be written as $\sum_{(j,i)\in\mathcal{E}} \mu_{ji}^T (y_{ji} - y_j) = \frac{1}{2}\omega^T (M_y \otimes I_l)\omega$.

To solve the multi-agent optimization problem, we need to find the stationary points of the network Lagrangian (7). By taking the subgradient of $\mathcal{L}_{\text{net}}$ w.r.t. each variable and reversing the sign of the rows corresponding to the dual variables, we can derive the set-valued operator $\mathcal{T}$ given by:

$$\begin{aligned} \mathcal{T} : \psi \mapsto{}& \partial\big(\textstyle\sum_{i\in\mathcal{N}} J_i(x_i, \sigma_i) + \iota_{X_i}(x_i) + \iota_{Q_i}(x_i, \sigma_i) \\ &+ \iota_{\mathcal{F}_i}(x_i, \sigma_i, \boldsymbol{y}_i)\big) + (2D + \bar{M}_y')\psi, \\ \text{with } 2D + \bar{M}_y' :={}& \begin{bmatrix} \mathbf{0} & \mathbf{0} & A^T B_l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B_l & \mathbf{0} \\ -B_l^T A & -B_l^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & M_y' \end{bmatrix}, \end{aligned} \tag{8}$$

where $A := \text{blkd}(A_i)_{i\in\mathcal{N}}$; B is the incidence matrix of the communication graph $\mathcal{G}$; $B_l := (B \otimes I_l)$; $M_y' := \text{blkd}(M_{y_i}')_{i\in\mathcal{N}}$ with M_{y_i}' being a skew-symmetric matrix generated by reversing the sign of the even rows of M_{y_i}; $\bar{M}_y'$ embeds M_y' to the lower right corner of a zero matrix; $2D$ is the skew-symmetric matrix enforcing the consensus of $\{A_i x_i + \sigma_i\}$; σ and λ are the stack vectors of $\{\sigma_i\}$ and $\{\lambda_{ji}\}$; ψ denotes the stack of the these variables, i.e. $\psi := [x; \sigma; \lambda; \omega]$.

Theorem 1. *Suppose Assumptions 1, 2, and 4 hold. Then for any zero point $[x^*; \sigma^*; \lambda^*; \omega^*] \in Zer(\mathcal{T})$, x^* is a minimizer of (1). Conversely, for any minimizer $x^\dagger$ of the problem (1), we can choose proper $\sigma^\dagger$, $\lambda^\dagger$, and $\omega^\dagger$, such that $[x^\dagger; \sigma^\dagger; \lambda^\dagger; \omega^\dagger] \in Zer(\mathcal{T})$.*

Proof: See [24, Appendix B].

B. Operator Splitting and Distributed Algorithm

We can split the operator $\mathcal{T}$ into two operators $\mathcal{A}$ and $\mathcal{B}$, and construct a design matrix Φ, which are given by:

$$\mathcal{A}:\psi\mapsto\partial(\sum_{i\in\mathcal{N}}J_i(x_i,\sigma_i))+(D+\bar{M}_y')\psi,$$
$$\mathcal{B}:\psi\mapsto\partial(\sum_{i\in\mathcal{N}}\iota_{\mathcal{X}_i}(x_i)+\iota_{\mathcal{Q}_i}(x_i,\sigma_i)+\iota_{\mathcal{F}_i}(x_i,\sigma_i,\boldsymbol{y}_i))+D\psi,$$
$$\Phi:=\begin{bmatrix}\boldsymbol{\tau}_1^{-1}&\mathbf{0}&-\frac{1}{2}A^TB_l&\mathbf{0}\\ \mathbf{0}&\boldsymbol{\tau}_2^{-1}&-\frac{1}{2}B_l&\mathbf{0}\\ -\frac{1}{2}B_l^TA&-\frac{1}{2}B_l^T&\boldsymbol{\tau}_3^{-1}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\boldsymbol{\tau}_4^{-1}\end{bmatrix},$$

where $\boldsymbol{\tau}_1:=\mathrm{blkd}(\tau_{11}I_n,\ldots,\tau_{1N}I_n)$ with the scalars $\tau_{1i}>0$ for $i\in\mathcal{N}$; similarly for $\boldsymbol{\tau}_2$, $\boldsymbol{\tau}_3$, and $\boldsymbol{\tau}_4$. These step sizes can be chosen based on the Gershgorin circle theorem [25] to guarantee $\Phi\in\mathbb{S}_{++}$. The operator $\mathcal{A}$ is maximally monotone since in Assumption 1, each objective J_i is assumed to be jointly convex in x_i and σ_i, and $D+\bar{M}_y'$ is skew-symmetric. The maximal monotonicity of $\mathcal{B}$ can be similarly established.

As suggested in the DR splitting (3), we next evaluate the analytical expressions for $J_{\Phi^{-1}\mathcal{A}}$ and $J_{\Phi^{-1}\mathcal{B}}$. For brevity, let $\lambda_{iB}^{(k)}:=\sum_{j\in\mathcal{N}_i^+}\lambda_{ji}^{(k)}-\sum_{j\in\mathcal{N}_i^-}\lambda_{ij}^{(k)}$, which can be obtained through the communications among agent i and its incident edges. The variable $\hat{\lambda}_{iB}^{(k)}$ is defined similarly.

For $\psi^{(k+1)}:=J_{\Phi^{-1}\mathcal{A}}(\tilde{\psi}^{(k)})$, it corresponds to the inclusion $(\Phi+\mathcal{A})\psi^{(k+1)}\ni\Phi\tilde{\psi}^{(k)}$. Each agent i can update its local decisions x_i and local estimates σ_i by solving the following problem using local information $\tilde{x}_i^{(k)}$, $\tilde{\sigma}_i^{(k)}$ and the dual information $\tilde{\lambda}_{iB}^{(k)}$ from its incident edges:

$$\begin{aligned}\underset{x_i,\sigma_i}{\text{minimize}}\ &J_i(x_i,\sigma_i)+\tfrac{1}{2}(\tilde{\lambda}_{iB}^{(k)})^T(A_ix_i+\sigma_i)\\&+\tfrac{1}{2\tau_{1i}}\|x_i-\tilde{x}_i^{(k)}\|_2^2+\tfrac{1}{2\tau_{2i}}\|\sigma_i-\tilde{\sigma}_i^{(k)}\|_2^2.\end{aligned}\tag{9}$$

After both incident agents solve (9), the dual variable $\lambda_{ji}^{(k+1)}$ maintained by the edge (j,i) can be updated by:

$$\lambda_{ji}^{(k+1)}=\tilde{\lambda}_{ji}^{(k)}+\tfrac{1}{2}\tau_{3ji}\cdot\big(A_i\hat{x}_i^{(k+1)}+\hat{\sigma}_i^{(k+1)}-A_j\hat{x}_j^{(k+1)}-\hat{\sigma}_j^{(k+1)}\big),\tag{10}$$

where $\hat{x}_i^{(k+1)}$ and $\hat{\sigma}_i^{(k+1)}$ are the results of $R_{\Phi^{-1}\mathcal{A}}$ as shown in (3). The update of $\omega^{(k+1)}$ is given by $(\boldsymbol{\tau}_4^{-1}+M_y')\cdot\omega^{(k+1)}=\boldsymbol{\tau}_4^{-1}\tilde{\omega}^{(k)}$, which is independent of the updates of x, σ and λ. To have the analytical solution to it, notice that:

$$\begin{aligned}&\mu_{ij}^{(k+1)}+\tau_{4i}(y_i^{(k+1)}-y_{ij}^{(k+1)})=\tilde{\mu}_{ij}^{(k)},&&\text{edge }(i,j);\\&y_i^{(k+1)}+\tau_{4i}(-\textstyle\sum_{j\in\mathcal{N}_i^-}\mu_{ij}^{(k+1)})=\tilde{y}_i^{(k)},&&\text{agent }i;\\&y_{ij}^{(k+1)}+\tau_{4i}\mu_{ij}^{(k+1)}=\tilde{y}_{ij}^{(k)},&&\text{agent }j.\end{aligned}\tag{11}$$

The above computation is restricted to the agent i, its out-edges, and its out-neighbors. We can thus derive the analytical expressions for the local updates of auxiliary variables and their dual variables, which are given in Subroutine 1.

For $\bar{\psi}^{(k+1)}:=J_{\Phi^{-1}\mathcal{B}}(\hat{\psi}^{(k+1)})$, the inclusions of $\bar{x},\bar{\sigma}$, and $\bar{y}$ are coupled locally, and each agent i should update them by solving the following local system of inclusions:

$$\begin{aligned}&\partial_{x_i}\iota_{\mathcal{Q}_i}(\bar{x}_i^{(k+1)},\bar{\sigma}_i^{(k+1)})+\partial_{x_i}\iota_{\mathcal{F}_i}(\bar{x}_i^{(k+1)},\bar{\sigma}_i^{(k+1)},\bar{\boldsymbol{y}}_i^{(k+1)})\\&\quad+\partial_{x_i}\iota_{\mathcal{X}_i}(\bar{x}_i^{(k+1)})+\tfrac{1}{2}A_i^T\hat{\lambda}_i^{(k+1)}+\tfrac{1}{\tau_{1i}}(\bar{x}_i^{(k+1)}-\hat{x}_i^{(k+1)})\ni\mathbf{0}\\&\partial_{\sigma_i}\iota_{\mathcal{Q}_i}(\bar{x}_i^{(k+1)},\bar{\sigma}_i^{(k+1)})+\partial_{\sigma_i}\iota_{\mathcal{F}_i}(\bar{x}_i^{(k+1)},\bar{\sigma}_i^{(k+1)},\bar{\boldsymbol{y}}_i^{(k+1)})\\&\quad+\tfrac{1}{2}\hat{\lambda}_i^{(k+1)}+\tfrac{1}{\tau_{2i}}(\bar{\sigma}_i^{(k+1)}-\hat{\sigma}_i^{(k+1)})\ni\mathbf{0}\\&\partial_{\boldsymbol{y}_i}\iota_{\mathcal{F}_i}(\bar{x}_i^{(k+1)},\bar{\sigma}_i^{(k+1)},\bar{\boldsymbol{y}}_i^{(k+1)})+\tfrac{1}{\tau_{4i}}(\bar{\boldsymbol{y}}_i^{(k+1)}-\hat{\boldsymbol{y}}_i^{(k+1)})\ni\mathbf{0}.\end{aligned}\tag{12}$$

Define $M_{\mathcal{F}_i}:=[(N-1)A_i,-I_l,-W_{ii}\otimes I_l,-[W_{ji}]_{j\in\mathcal{N}_i^+}^T\otimes I_l]\in\mathbb{R}^{l\times(n_i+l(N_i^++2))}$, and $\check{x}_i^{(k+1)}:=\hat{x}_i^{(k+1)}-\frac{\tau_{1i}}{2}A_i^T\hat{\lambda}_i^{(k+1)}$ and $\check{\sigma}_i^{(k+1)}:=\hat{\sigma}_i^{(k+1)}-\frac{\tau_{2i}}{2}\hat{\lambda}_i^{(k+1)}$. Then, finding zeros of (12) is equivalent to solving the following constrained minimization problem:

$$\begin{cases}\underset{x_i\in\mathcal{X}_i,\sigma_i,\boldsymbol{y}_i}{\text{minimize}}&\frac{1}{2\tau_{1i}}\|x_i-\check{x}_i^{(k+1)}\|^2+\frac{1}{2\tau_{2i}}\|\sigma_i-\check{\sigma}_i^{(k+1)}\|^2\\&\qquad+\frac{1}{2\tau_{4i}}\|\boldsymbol{y}_i-\hat{\boldsymbol{y}}_i^{(k+1)}\|^2\\\text{subject to}&M_{\mathcal{F}_i}\cdot[x_i;\sigma_i;\boldsymbol{y}_i]=\mathbf{0},A_ix_i+\sigma_i\le c\end{cases}.\tag{13}$$

For the variables maintained by the edges, the dual variable $\bar{\mu}$ remains the same, while $\bar{\lambda}_{ji}$ is updated by:

$$\begin{aligned}\bar{\lambda}_{ji}^{(k+1)}=\hat{\lambda}_{ji}^{(k+1)}+\tau_{3ji}\cdot\big(&A_i\bar{x}_i^{(k+1)}+\bar{\sigma}_i^{(k+1)}-A_j\bar{x}_j^{(k+1)}\\&-\bar{\sigma}_j^{(k+1)}-\tfrac{1}{2}(A_i\hat{x}_i^{(k+1)}+\hat{\sigma}_i^{(k+1)}-A_j\hat{x}_j^{(k+1)}-\hat{\sigma}_j^{(k+1)})\big).\end{aligned}\tag{14}$$

Overall, Algorithm 2 summarizes the proposed algorithm for finding a minimizer of (1) based on the DR framework (3).

Subroutine 1: Auxiliary-Consensus-Update (ACU)

Input: $\{\tilde{\boldsymbol{y}}_i^{(k)}\}$, $\{\tilde{\mu}_{ji}^{(k)}\}$;
for *agent* $i\in\mathcal{N}$ **do**
  For each $j\in\mathcal{N}_i^-$, receive $\tilde{\mu}_{ij}^{(k)}$ and $\tilde{y}_{ij}^{(k)}$;
  $y_i^{(k+1)}:=\frac{1+\tau_{4i}^2}{1+\tau_{4i}^2(1+N_i^-)}(\tilde{y}_i^{(k)}+\frac{\tau_{4i}}{1+\tau_{4i}^2}\sum_{j\in\mathcal{N}_i^-}(\tilde{\mu}_{ij}^{(k)}+\tau_{4i}\tilde{y}_{ij}^{(k)}))$;
end
for *edge* $(j,i)\in\mathcal{E}$ **do**
  Receive $y_j^{(k+1)}$ from its tail and $\tilde{y}_{ji}^{(k)}$ from its head ;
  $\mu_{ji}^{(k+1)}:=\frac{1}{1+\tau_{4j}^2}\tilde{\mu}_{ji}^{(k)}+\frac{\tau_{4j}}{1+\tau_{4j}^2}(\tilde{y}_{ji}^{(k)}-y_j^{(k+1)})$;
end
for *agent* $i\in\mathcal{N}$ **do**
  For each $j\in\mathcal{N}_i^+$, receive $\mu_{ji}^{(k+1)}$ from its in-edge ;
  $y_{ji}^{(k+1)}:=\tilde{y}_{ji}^{(k)}-\tau_{4j}\mu_{ji}^{(k+1)}$;
end
Return: $\{\boldsymbol{y}_i^{(k+1)}\}$, $\{\mu_{ji}^{(k+1)}\}$.

IV. Commodity Distribution Problem

We consider a commodity distribution problem adapted from [22, Sec. 1.4.3], [16], where several branches of the same company produce a common homogeneous commodity. A transport network exists that has markets as its nodes and roads as its edges. Denote the node set of this network by $\mathcal{N}_T$ and the edge set $\mathcal{E}_T$, the cardinalities of which are N_T and E_T, respectively. These branches attempt to cooperatively optimize their total profit by deciding the production quantities at the factories and the distribution quantities over the markets.

Algorithm 2: Algorithm for Globally-Coupled Multi-Agent Optimization Problem

Initialize: $\{\tilde{x}_i^{(0)}\}, \{\bar{\sigma}_i^{(0)}\}, \{\tilde{\lambda}_{ji}^{(0)}\}, \{\bar{y}_i^{(0)}\}, \{\tilde{\mu}_{ji}^{(0)}\}$;
Iterate until convergence:
$(\{y_i^{(k+1)}\}, \{\mu_{ji}^{(k+1)}\}) := \mathrm{ACU}(\{\tilde{y}_i^{(k)}\}, \{\tilde{\mu}_{ji}^{(k)}\})$;
for *agent* $i \in \mathcal{N}$ **do**
  Receive $\tilde{\lambda}_{ji}^{(k)}$ and $\tilde{\lambda}_{ij}^{(k)}$ from its in- and out-edges ;
  Obtain $x_i^{(k+1)}$ and $\sigma_i^{(k+1)}$ by solving (9);
  R-R updates: $\hat{x}_i^{(k+1)}, \hat{\sigma}_i^{(k+1)}, \hat{y}_i^{(k+1)}$;
end
for *edge* $(j,i) \in \mathcal{E}$ **do**
  Receive $A_i\hat{x}_i^{(k+1)} + \hat{\sigma}_i^{(k+1)}$ and $A_i\hat{x}_j^{(k+1)} + \hat{\sigma}_j^{(k+1)}$;
  Obtain $\lambda_{ji}^{(k+1)}$ by (10); R-R updates: $\hat{\lambda}_{ji}^{(k+1)}$;
end
for *agent* $i \in \mathcal{N}$ **do**
  Receive $\hat{\lambda}_{ji}^{(k+1)}$ and $\hat{\lambda}_{ij}^{(k+1)}$ from its in- and out-edges ;
  Obtain $\bar{x}_i^{(k+1)}$, $\bar{\sigma}_i^{(k+1)}$ and $\bar{y}_i^{(k+1)}$ by solving (13);
end
for *edge* $(j,i) \in \mathcal{E}$ **do**
  Receive $A_i\bar{x}_i^{(k+1)} + \bar{\sigma}_i^{(k+1)}$ and $A_i\bar{x}_j^{(k+1)} + \bar{\sigma}_j^{(k+1)}$;
  Obtain $\bar{\lambda}_{ji}^{(k+1)}$ by (14);
end
K-M updates: $\tilde{\psi}^{(k+1)} = \tilde{\psi}^{(k)} + 2\gamma^{(k)}(\bar{\psi}^{(k+1)} - \psi^{(k+1)})$;
Return: $\{x_i^{(k)}\}$.

Each branch i delivers the commodity from the factories, denoted by $\mathcal{N}_{T_i}$, to different markets through the transport network. Let $N_{T_i} := |\mathcal{N}_{T_i}|$. Its decision vector $x_i \in \mathbb{R}^{n_i}$ with $n_i := E_T + N_{T_i}$ consists of two parts: $u_i \in \mathbb{R}_+^{E_T}$ represents the quantities of commodity transported through each road $e_T \in \mathcal{E}_T$; $v_i \in \mathbb{R}_+^{N_{T_i}}$ denotes the quantities of commodity produced by the factories owned by branch i. These two parts uniquely determine the distribution of commodity over the markets. Assuming the factories owned by branch i have maximum production capacities $b_i \in \mathbb{R}_{++}^{N_{T_i}}$, each entry of the vector $u_i \in \mathbb{R}^{E_T}$ is upper-bounded by $\|b_i\|_1$. Denote by $B_T \in \mathbb{R}^{N_T \times E_T}$ the incidence matrix of this transport network, and by $E_i \in \mathbb{R}^{N_T \times N_{T_i}}$ the indicator matrix which maps from each entry of v_i to the corresponding markets. Then, we have the local matrix $A_i := [B_T, E_i]$ and the local feasible set $\mathcal{X}_i := \{x_i \in \mathbb{R}^{E_T+N_{T_i}} | \mathbf{0} \le v_i \le b_i, \mathbf{0} \le u_i \le \|b_i\|_1 \otimes \mathbf{1}_{E_T}, A_i x_i \ge \mathbf{0}\}$.

The objective function of branch $i \in \mathcal{N}$ is given by $J_i(x_i, x_{-i}) = \frac{1}{2}x_i^T Q_i x_i + \alpha_i \|Ax\|_2^2 - (w - \Sigma Ax)^T A_i x_i$, where $A := [A_1, \ldots, A_N]$, $x := [x_1; \cdots; x_N]$, $Q_i \in \mathbb{S}_{++}^{n_i}$ is a diagonal matrix, and we let $w \in \mathbb{R}_{++}^{N_T}$ denote the initial unit price, and $\Sigma \in \mathbb{S}_{++}^{N_T}$ the decreasing rate of unit price. We further assume that there is a maximum capacity $c \in \mathbb{R}_{++}^{N_T}$ for the commodity sold at different markets, and the global constraints are accordingly defined as $\sum_{i\in\mathcal{N}} A_i x_i \le c$.

We use the transport network of the city of Oldenburg [26]: it consists of $N_T = 29$ nodes (markets) and $E_T = 2\times 34$ edges (roads). Five branches ($N = 5$) participate in this problem, each owning a single factory at the given location $(\{8, 14, 21, 10, 29\})$. Each factory has a maximum production capacity uniformly randomly chosen from $[10, 14]$. The normalized length η_{e_T} is defined as the ratio between the length of road e_T and the maximum road length. Each diagonal entry of matrix Q_i corresponding to u_i is set as $7\eta_{e_T}$, while the entries corresponding to v_i are fixed to be 2.8. Moreover, $w = 36 \cdot \mathbf{1}_{N_T}$, and Σ is a matrix with $[\Sigma]_{ii} = 0.23$, for all $i \in \mathcal{N}_T$, $[\Sigma]_{ji} = 0.069 \cdot (1 - \eta_{(j,i)})$, for all $(j,i) \in \mathcal{E}_T$, and otherwise zero. We also let $\alpha_i := 0.21$ and $c := 2 \cdot \mathbf{1}_{N_T}$. We can verify numerically that each J_i is jointly convex in x_i and s_i as required in Assumption 1. The communication network $\mathcal{G}$ contains a directed circle and $\lfloor \frac{N}{2} \rfloor = 2$ randomly selected directed edges.

The package CVXPY 1.0.31 [27], [28] is used to solve the constrained optimization problem for each agent. We illustrate the performances of Algorithm 2 in Fig. 1. Fig. 1(a)(b) describe the relative updating step sizes at each iteration for x_i and y_i. Fig. 1(c)(d)(e) illustrate how the feasibility conditions are satisfied at each iteration. Fig. 1(e) shows the

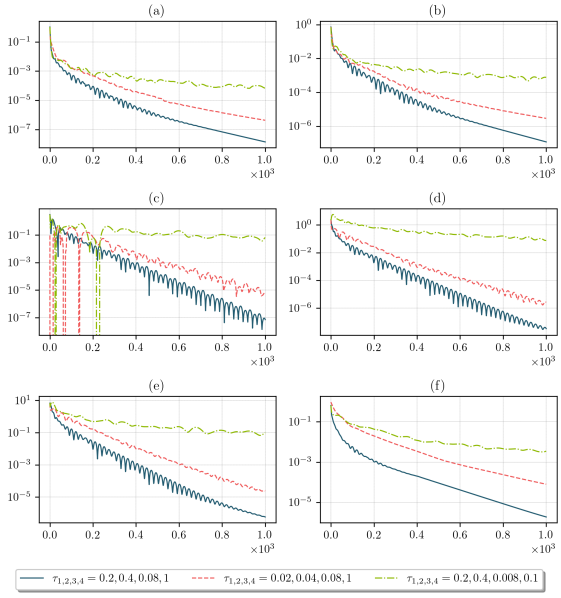

Fig. 1: Performances of Algorithm 2 (a) $\sum_{i\in\mathcal{N}}\|x_i^{(k+1)} - x_i^{(k)}\|/Nx_i^{(k)}$, (b) $\sum_{i\in\mathcal{N}}\|y_i^{(k+1)} - y_i^{(k)}\|/N\|y_i^{(k)}\|$, (c) $\|\max\{0, \sum_{i\in\mathcal{N}} A_i x_i^{(k)} - c\}\|$, (d) $\sum_{(j,i)\in\mathcal{E}}\|y_{ji}^{(k+1)} - y_j^{(k)}\|/E$, (e) $\sum_{i\in\mathcal{N}}\|\sigma_i^{(k)} - \sum_{j\in\mathcal{N}_{-i}} A_j x_j^{(k)}\|/N$, (f) $\sum_{i\in\mathcal{N}}\|x_i^{(k+1)} - x_i^*\|/N\|x_i^*\|$.

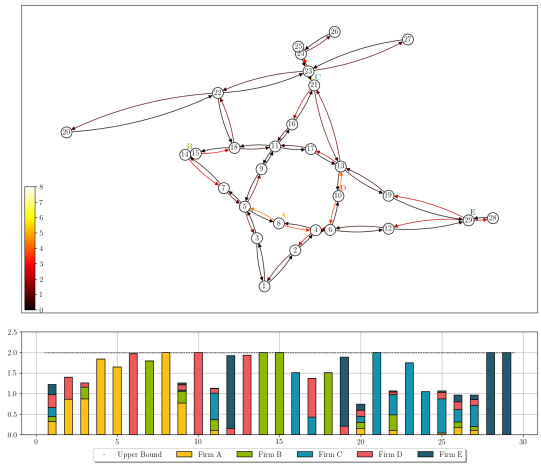

Fig. 2: The Commodity Distribution over the Transport Net

normalized distance between the decision variables obtained by the proposed algorithm and the unique minimizer. Fig.2 visualizes the solution computed by Algorithm 2, which is reflected by the commodity distribution over the markets and the quantity of commodity transported through each road. These numerical results verify the validity of Algorithm 2, and show a linear convergence rate towards a solution.

V. Conclusions and Future Directions

We propose a distributed solution for multi-agent optimization problems that are globally coupled by aggregates. Although we only discuss the cases with homogeneous aggregates in objectives and constraints, with the introduction of a new set of auxiliary variables and its associated local estimates, the proposed algorithm can solve problems with heterogeneous aggregates. Moreover, the results can be directly extended to handle the problems with convex constraints. One of our future directions is to extend the primal decomposition method to the non-cooperative setting. Even though the analysis in this paper can be applied to non-cooperative aggregative games with minor modifications, yet to ensure the convergence of the algorithm candidate, we need to postulate that the invoked extended pseudogradient operators are maximally monotone, which holds if and only if the partial derivatives of local objectives w.r.t. local decisions do not depend on others' decisions [13], [29], [30]. This condition dramatically restricts the applicability of the proposed method. In addition, we note that even under the cooperative setting, the solvable multi-agent optimization problems should have their local objectives jointly convex in the local decisions and the aggregates. Another future direction would lie in relaxing this assumption and enable the proposed algorithm to handle a wider range of problems.

Acknowledgements

The authors thank the reviewers for their suggestions and comments, in particular for pointing out the applicability issue of one of our previous assumptions. To address this comment, we modify the problem setting from non-cooperative games to cooperative cases. Extension to the non-cooperative setting is one of our future directions.

References

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

[2] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[3] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3649–3656.

[4] J. Hu, Y. Xiao, and J. Liu, "Distributed algorithms for solving locally coupled optimization problems on agent networks," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 2420–2425.

[5] M. K. Jensen, "Aggregative games and best-reply potentials," *Economic theory*, vol. 43, no. 1, pp. 45–66, 2010.

[6] J. Barrera and A. Garcia, "Dynamic incentives for congestion control," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 299–310, 2014.

[7] X. Wei and A. Anastasopoulos, "Mechanism design for demand management in energy communities," *Games*, vol. 12, no. 3, p. 61, 2021.

[8] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on control systems technology*, vol. 21, no. 1, pp. 67–78, 2011.

[9] A. Camisa, F. Farina, I. Notarnicola, and G. Notarstefano, "Distributed constraint-coupled optimization via primal decomposition over random time-varying graphs," *Automatica*, vol. 131, p. 109739, 2021.

[10] 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.

[11] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for network resource allocation problems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 621–636, 2019.

[12] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[13] G. Belgioioso and S. Grammatico, "A Douglas-Rachford splitting for semi-decentralized equilibrium seeking in generalized aggregative games," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3541–3546.

[14] P. Yi and L. Pavel, "An operator splitting approach for distributed generalized Nash equilibria computation," *Automatica*, vol. 102, pp. 111–121, 2019.

[15] L. Pavel, "Distributed GNE seeking under partial-decision information over networks via a doubly-augmented operator splitting approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1584–1597, 2019.

[16] F. Parise, B. Gentile, and J. Lygeros, "A distributed algorithm for almost-Nash equilibria of average aggregative games with coupling constraints," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 770–782, 2019.

[17] F. Parise, S. Grammatico, B. Gentile, and J. Lygeros, "Distributed convergence to Nash equilibria in network and average aggregative games," *Automatica*, vol. 117, p. 108959, 2020.

[18] S. Liang, P. Yi, and Y. Hong, "Distributed Nash equilibrium seeking for aggregative games with coupled constraints," *Automatica*, vol. 85, pp. 179–185, 2017.

[19] D. Gadjov and L. Pavel, "Single-timescale distributed GNE seeking for aggregative games over networks via forward-backward operator splitting," *IEEE Transactions on Automatic Control*, 2020.

[20] M. Bianchi, G. Belgioioso, and S. Grammatico, "Fast generalized Nash equilibrium seeking under partial-decision information," *arXiv preprint arXiv:2003.09335*, 2020.

[21] G. Belgioioso, A. Nedić, and S. Grammatico, "Distributed generalized Nash equilibrium seeking in aggregative games on time-varying networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2061–2075, 2020.

[22] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[23] H. H. Bauschke, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed., ser. CMS Books in Mathematics, Ouvrages de mathématiques de la SMC, 2017.

[24] Y. Huang and J. Hu, "A primal decomposition approach to globally coupled aggregative optimization over networks," *arXiv preprint arXiv:2109.12297*, 2021.

[25] H. E. Bell, "Gershgorin's theorem and the zeros of polynomials," *The American Mathematical Monthly*, vol. 72, no. 3, pp. 292–295, 1965.

[26] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.

[27] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[28] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[29] M. Ye and G. Hu, "Distributed Nash equilibrium seeking by a consensus based approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4811–4818, 2017.

[30] W. Shi and L. Pavel, "LANA: an ADMM-like Nash equilibrium seeking algorithm in decentralized environment," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 285–290.