A General Framework for Distributed Partitioned Optimization

Savelii Chezhegov [*] **Anton Novitskii** [*] **Alexander Rogozin** [**]
Sergei Parsegov [***] **Pavel Dvurechensky** [****]
Alexander Gasnikov [**,*]

[*] *ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia*
[**] *Moscow Institute of Physics and Technology, Dolgoprudny, Russia*
[***] *Skolkovo Institute of Science and Technology, Moscow, Russia*
[****] *Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany*

Abstract: Decentralized optimization is widely used in large scale and privacy preserving machine learning and various distributed control and sensing systems. It is assumed that every agent in the network possesses a local objective function, and the nodes interact via a communication network. In the standard scenario, which is mostly studied in the literature, the local functions are dependent on a common set of variables, and, therefore, have to send the whole variable set at each communication round. In this work, we study a different problem statement, where each of the local functions held by the nodes depends only on some subset of the variables. Given a network, we build a general algorithm-independent framework for decentralized partitioned optimization that allows to construct algorithms with reduced communication load using a generalization of Laplacian matrix. Moreover, our framework allows to obtain algorithms with non-asymptotic convergence rates with explicit dependence on the parameters of the network, including accelerated and optimal first-order methods. We illustrate the efficacy of our approach on a synthetic example.

Keywords: Large scale optimization problems, Convex optimization, Optimization and control of large-scale network systems, Decentralized and distributed control, Multiagent systems

1. INTRODUCTION

Distributed algorithms is a classical Borkar and Varaiya (1982); Tsitsiklis and Athans (1984); DeGroot (1974), yet actively developing research area with many applications including robotics, resource allocation, power system control, control of drone or satellite networks, distributed statistical inference and optimal transport, multiagent reinforcement learning Xiao and Boyd (2006); Rabbat and Nowak (2004); Ram et al. (2009); Kraska et al. (2013); Uribe et al. (2018); Kroshnin et al. (2019); Ivanova et al. (2020). Recent surge of interest to such problems in optimization and machine learning is motivated by large-scale learning problems with privacy constraints and other challenges such as data being produced or stored distributedly Bottou (2010); Boyd et al. (2011); Nedić et al. (2017). An important part of this research studies decentralized distributed optimization algorithms over arbitrary networks of computing agents, e.g. sensors or computers, which is represented by a connected graph in which two agents can communicate with each other if there is an edge between them. This imposes communication constraints and the goal of the whole system Nedić et al. (2009) is to cooperatively minimize a global objective using only local communications between agents, each of which has access only to a local piece of the global objective.

In this paper, we further exploit additional structure in such problems and consider distributed partitioned optimization problems also known as optimization with overlapping variables and distributed optimization of partially separable objective functions. Such problems arise in many modern big-data applications, e.g., distributed matrix completion, distributed estimation in power networks, network utility maximization, distributed resource allocation, cooperative localization in wireless networks, building maps by robotic networks Erseghe (2012); Kekatos and Giannakis (2012); Carli and Notarstefano (2013); Notarnicola et al. (2017); Cannelli et al. (2020). As in the standard formulations, the goal is to minimize a sum of m functions $\tilde{f}_i(x)$, $i = 1, ..., m$ with each $\tilde{f}_i(x)$ stored at a node of the computational network. Yet, unlike standard distributed problems, the space of decision variables is divided into n blocks, and each of $\tilde{f}_i(x)$ may depend only on a, possibly small, subset of blocks. Such sparse structure leads to inefficiency of the standard approaches Scaman et al. (2017); Kovalev et al. (2020) since they require each node to store and send the whole vector of variables instead of storing and exchanging with other nodes a small vector of variables which influences

[⋆] The research is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) №075-00337-20-03, project No. 0714-2020-0005.

[⋆⋆]S. Chezhegov and A. Novitskii contributed equally.

its local objective $\tilde{f}_i(x)$. This leads to inefficient usage of computational and communication resources.

Theory of algorithms exploiting such additional structure seems to be underdeveloped in the literature. Necoara and Clipici (2016) propose a parallel version of a randomized (block) coordinate descent method for minimizing the sum of a partially separable smooth convex function and a fully separable non-smooth convex function. Moreover, they explain how to implement their algorithms in a distributed setup and obtain convergence rate guarantees. Cannelli et al. (2020) consider convex and nonconvex constrained optimization with a partially separable objective function and propose an asynchronous algorithm with rate guarantees for this class of problems. Finally, Notarnicola et al. (2017) propose asynchronous dual decomposition algorithm for such problems and prove its asymptotic convergence.

Despite very advanced results and techniques, these works have two limitations. Firstly, their distributed algorithms assume that the number of functions m is the same as the number of variables blocks n and each node i stores not only the objective $\tilde{f}_i$, but also the i-th block of variables. Secondly, and more importantly, they assume that the computational graph is aligned with the dependence of $\tilde{f}_i$'s on the blocks of variables. The latter means that if $\tilde{f}_i$ depends on the block variable x^ℓ, then the nodes i and ℓ are connected by an edge of the computational network. In this paper, we do not make such assumptions and consider a more general setting. Moreover, we propose a general algorithm-independent framework that allows to reformulate distributed partitioned optimization problem using in a way suitable for application of many decentralized distributed optimization algorithms, see, e.g., Yang et al. (2019); Gorbunov et al. (2020); Dvinskikh and Gasnikov (2021). Our approach makes it possible to go beyond optimization problems and apply it to decentralized methods for saddle-point problems Rogozin et al. (2021) and variational inequalities Kovalev et al. (2022). At the core of our framework lie mixing matrices, e.g., Laplacian of the computational graph, which are widely used in decentralized optimization Gorbunov et al. (2020). Our approach allows to flexibly choose computational subgraph for each block of variables and the corresponding mixing matrices in order to make the storage, computational, and communication complexity smaller. This, in particular, allows obtaining algorithms with non-asymptotic convergence rates (unlike Notarnicola et al. (2017)) with explicit dependence on the parameters of the computational network, including accelerated and optimal algorithms (unlike Cannelli et al. (2020); Necoara and Clipici (2016)).

We illustrate the effectiveness of our reformulation-based framework by considering graphs of a certain structure. These graphs have a two-layer hierarchy: the first layer is represented by groups of nodes that communicate on their local variable blocks, while the upper level reflects the communications between the groups. Within the intergroup information exchange, the nodes from each group share the variables according to the links between the groups. The reasons to consider such topologies are as follows: first, such structures are scalable in terms of the number of groups and the number of agents within each group. Second, these graphs admit closed-form calculation of the Laplacian spectra, which influence the convergence rates of distributed algorithms. Moreover, such hierarchical graphs mimic the nature of the distributed estimators that consider local variables as private information and exchange the shared variables with certain neighbors only (e.g. in distributed power system state estimation, see Kekatos and Giannakis (2012); Notarnicola et al. (2017). Such graphs allow us to study the asymptotics of the condition number of the Laplacian matrix for both approaches: with a common state vector and with blocks of variables, and show that our approach leads to better convergence rates of distributed algorithms.

This paper is organized as follows. In Section 2, we describe our framework for partitioned optimization problems. Namely, we describe the generalization of Laplacian matrix in Section 2.1, give an example of building such a matrix in Section 2.2 and analyze its spectral characteristics in Section 2.3. After that, we illustrate how our approach works on a synthetic network example in Section 3. There we consider a hierarchical network that consists of n cliques of size k connected by a ring graph and show that using our approach decreases the condition number of the communication matrix by $\Theta(n^2k)$ times.

1.1 Notation

Throughout this paper, $\mathbf{L}(\mathcal{G})$ denotes the Laplacian matrix of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$[\mathbf{L}(\mathcal{G})]_{ij} = \begin{cases} \deg(i), & \text{if } i = j, \\ -1, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{else,} \end{cases}$$

where $\deg(i)$ denotes the number of nodes adjacent to node i. We also let $\mathbf{I}_p$ be the identity matrix of size $p \times p$, $\mathbf{1}_p$ be the all-ones vector of length p and $\mathbf{0_p}$ be a vector consisting of p zeros. We denote $\mathbf{e}_p^{(q)} = (0, \ldots, 1, \ldots, 0)^\top$ the q-th coordinate vector in $\mathbb{R}^p$. After that, $\lambda_{\max}(\cdot)$ and $\lambda^+_{\min}(\cdot)$ are the largest and smallest positive eigenvalues of matrix, respectively, and $\chi(\cdot) = \lambda_{\max}(\cdot)/\lambda^+_{\min}(\cdot)$ is the condition number. Finally, $\mathbf{\Lambda}(\cdot)$ denotes the set of *unique* eigenvalues of a matrix.

2. DISTRIBUTED PARTITIONED OPTIMIZATION

2.1 The proposed framework

Consider the following distributed partitioned optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x), \quad f(x) := \sum_{i=1}^m \tilde{f}_i\left(x^{[\mathcal{N}_i]}\right), \tag{1}$$

where $\mathcal{N}_i \subseteq \{1, \ldots, n\}$, i.e., each function $\tilde{f}_i$ depends on a subset $x^{[\mathcal{N}_i]}$ of variables[1] $x^\ell \in \mathbb{R}$, $\ell = 1, \ldots, n$ and these subsets may be of different size and even overlap. Further, we assume that there is a computational network represented by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, m\}$ is the set of nodes and $\mathcal{E}$ is the set of

[1] For simplicity, we consider variables $x^\ell \in \mathbb{R}$, but everything can be straightforwardly generalized for the case when $x^\ell \in \mathbb{R}^{n_\ell}$ are blocks of variables which do not intersect and the sum of all their dimensions is equal to n.

edges connecting the elements of $\mathcal{V}$, and that each $\tilde{f}_i$ is locally held by a separate computational node of $\mathcal{G}$. The goal of the network is to cooperatively solve problem (1) under communication constraints: two nodes may exchange information if and only if there is an edge in $\mathcal{E}$ connecting these nodes. Unlike previous works, it is allowed that two functions $\tilde{f}_i$ and $\tilde{f}_j$ depend on the same variable x^ℓ, but nodes i and j are not connected by an edge in $\mathcal{G}$.

To exploit the partitioned structure of the problem, for every variable x^ℓ, we define a set of nodes that hold functions dependent on x^ℓ: $\mathcal{V}^\ell \subseteq \mathcal{V}$ (i.e. $\mathcal{V}^\ell = \{i : \ \ell \in \mathcal{N}_i\}$) and consider an undirected and connected communication subnetwork $\tilde{\mathcal{G}}^\ell = (\mathcal{V}^\ell, \mathcal{E}^\ell)$ with $\mathcal{E}^\ell \subseteq \mathcal{E}$. By construction, an edge (i,j) lies in $\mathcal{E}^\ell$ if $\tilde{f}_i$ and $\tilde{f}_j$ depend on x^ℓ and $(i,j) \in \mathcal{E}$.

The standard approaches Gorbunov et al. (2020) to solve distributed optimization problems require each node to store a local approximation of the *whole* vector $x \in \mathbb{R}^n$ and communicate it to the neighbors. To reduce the storage requirements and the amount of communicated information, we assume that each node i holds an approximation x_i^ℓ *only* of the variables x^ℓ such that $\ell \in \mathcal{N}_i$, i.e. $\tilde{f}_i$ depends on x^ℓ. To obtain a problem equivalent to (1) we impose consensus constraints $x_{j_1}^\ell = x_{j_2}^\ell = \ldots = x_{j_{|\mathcal{V}^\ell|}}^\ell$, where $j_1, \ldots, j_{\mathcal{V}^\ell}$ are the nodes of $\mathcal{V}^\ell$. Since all the graphs $\tilde{\mathcal{G}}^\ell$ are connected and have vertices $\mathcal{V}^\ell$, we can equivalently rewrite (1) using the approximations x_i^ℓ as

$$\min_{\mathbf{x}\in\mathbb{R}^{mn}} \ F(\mathbf{x}) = \sum_{i=1}^{m} \tilde{f}_i(x_i^{[\mathcal{N}_i]}) \tag{2}$$
$$\text{s.t. } x_i^\ell = x_j^\ell \ \ \forall (i,j) \in \mathcal{E}^\ell, \ \ell = 1, \ldots, n.$$

The next reformulation step is based on stating the constraints of this problem as a system of linear equations using a communication matrix, which requires some notation. For each node i, we define a vector $x_i \in \mathbb{R}^n$ such that its ℓ-th component $x_i^\ell = 0$ if $\ell \notin \mathcal{N}_i$, i.e., $\tilde{f}_i$ does not depend on x^ℓ and x_i^ℓ is the above-defined approximation of x^ℓ if $\ell \in \mathcal{N}_i$. Also, we introduce the stacked vector $\mathbf{x} = [x_1^\top \ldots x_m^\top]^\top \in \mathbb{R}^{mn}$ and $\mathbf{x}^\ell = [x_1^\ell \ldots x_m^\ell]^\top \in \mathbb{R}^m$ to be the vector of approximations to the variable x^ℓ with the convention that $x_i^\ell = 0$ if $\ell \notin \mathcal{N}_i$. We further introduce graphs $\mathcal{G}^\ell = (\mathcal{V}, \mathcal{E}^\ell)$ for $\ell = 1, \ldots, m$, which are graphs $\tilde{\mathcal{G}}^\ell$ augmented with isolated vertices (if needed). The communication matrix $\mathbf{W}$ associated with the set of networks $\{\mathcal{G}^\ell\}_{\ell=1}^n$ is defined as [2]

$$\mathbf{W} = \sum_{\ell=1}^{n} \mathbf{L}(\mathcal{G}^\ell) \otimes \mathbf{e}_n^{(\ell)} \mathbf{e}_n^{(\ell)\top}. \tag{3}$$

According to the definition of $\mathbf{x}$, we have $(\mathbf{L}(\mathcal{G}^\ell) \otimes \mathbf{e}_n^{(\ell)} \mathbf{e}_n^{(\ell)\top})\mathbf{x} = (\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell) \otimes \mathbf{e}_n^{(\ell)}$ and $\mathbf{W}\mathbf{x} = \sum_{\ell=1}^m (\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell) \otimes \mathbf{e}_n^{(\ell)}$. Thus, we conclude that the linear constraint $\mathbf{W}\mathbf{x} = 0$ is equivalent to $\{\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell = 0\}_{\ell=1}^m$, which, in turn, by the definition of the graph Laplacian, are equivalent to the constraints in (2). Finally, we introduce $f_i(x) = \tilde{f}_i(x^{[\mathcal{N}_i]})$ for $i = 1, \ldots, m$. Functions f_i depend only on the variable subset $\mathcal{N}_i$, but formally take the whole variable vector as their argument. Combining everything together, we obtain the following equivalent reformulation of (1) and (2)

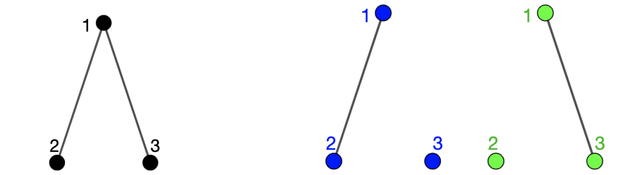
$$\min_{x\in\mathbb{R}^n} \ F(\mathbf{x}) := \sum_{i=1}^{m} f_i(x_i) \ \text{ s.t. } \mathbf{W}\mathbf{x} = 0. \tag{4}$$

Thus, our framework results in this reformulation which is standard for decentralized optimization and allows to apply a long list of algorithms to solve (1) by solving (4). Any decentralized consensus-based optimization algorithms Gorbunov et al. (2020), including the state-of-the-art primal (OPAPC Kovalev et al. (2020)) and dual (MSDA Scaman et al. (2017)) methods can be applied to our problem reformulation. The communication complexity of these algorithms explicitly depends on the parameters of the network through the condition number $\chi(\mathbf{W})$ of $\mathbf{W}$. Thus, in what follows we focus on studying the spectrum of $\mathbf{W}$.

Remark 1. When all f_i depend on the common set of variables, i.e. $\mathcal{N}_i = \{1, \ldots, n\}$ for $i = 1, \ldots, m$, we have $\mathbf{L}(\mathcal{G}^\ell) = W$ for $\ell = 1, \ldots, n$. In this case $\mathbf{W} = W \otimes \sum_{\ell=1}^n \mathbf{e}_n^{(\ell)} \mathbf{e}_n^{(\ell)\top} = W \otimes \mathbf{I}_n$, which is the standard communication matrix used in decentralized optimization.

2.2 *Example*

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, 3\}, \mathcal{E} = \{(1,2), (1,3)\}$, i.e. $\mathcal{G}$ and let us have two variables x^1, x^2. Let $\tilde{f}_1 = \tilde{f}_1(x^1, x^2)$, $\tilde{f}_2 = \tilde{f}_2(x^1)$ and $\tilde{f}_3 = \tilde{f}_3(x^2)$. We build corresponding Laplacians for the variables x^1 and x^2 as follows.



(a) Original graph $\mathcal{G}$ (b) Subgraphs $\mathcal{G}^1$ and $\mathcal{G}^2$

Fig. 1. Graph $\mathcal{G}$ of the computational network with three nodes and subgraphs $\mathcal{G}^1$ (in blue), $\mathcal{G}^2$ (in green)

$$\mathbf{L}(\mathcal{G}^1) = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}(\mathcal{G}^2) = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

According to (3) $\mathbf{W}$ has the following form:

$$\mathbf{W} = \mathbf{L}(\mathcal{G}^1) \otimes \mathbf{e}_1\mathbf{e}_1^\top + \mathbf{L}(\mathcal{G}^2) \otimes \mathbf{e}_2\mathbf{e}_2^\top$$
$$= \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

On the other hand, the Laplacian of the original network $\mathcal{G}$ (not taking into account the different variables) writes as

$$\mathbf{L}(\mathcal{G}) = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Note, that the condition number of $\mathbf{W}$ is better as compared to the one of $\mathbf{L}(\mathcal{G})$: we have $\chi(\mathbf{W}) = 1$ and $\chi(\mathbf{L}(\mathcal{G})) = 3$.

[2] Instead of $\mathbf{L}(\mathcal{G}^\ell)$ we can take a doubly stochastic mixing matrix.

2.3 Spectrum of $\mathbf{W}$

In this section, we analyze the spectrum of the novel communication matrix $\mathbf{W}$.

Lemma 2. For matrix $\mathbf{W}$ defined in (3) we have $\mathbf{\Lambda}(\mathbf{W}) = \bigcup_{\ell=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}^\ell))$.

Proof. Firstly, let $\mathbf{W}\mathbf{x} = \lambda\mathbf{x}$ for some $\mathbf{x} \neq 0$ and $\lambda \in \mathbb{C}$. By definition of $\mathbf{W}$, we have $\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell = \lambda\mathbf{x}^\ell$ for $\ell = 1, \dots, n$. Since $\mathbf{x} \neq 0$, there exists ℓ such that $\mathbf{x}^\ell \neq 0$, and therefore $\lambda \in \bigcup_{\ell=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}^\ell))$.

Secondly, let $\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell = \lambda\mathbf{x}^\ell$ for some $\ell = 1, \dots, n$. Setting $\mathbf{x} = \mathbf{x}^\ell \otimes \mathbf{e}_n^{(\ell)}$ we obtain

$$\mathbf{W}\mathbf{x} = \sum_{j=1}^{n} (\mathbf{L}(\mathcal{G}^\ell) \otimes \mathbf{e}_n^{(j)}\mathbf{e}_n^{(j)\top})(\mathbf{x}^\ell \otimes \mathbf{e}_n^{(\ell)})$$
$$= (\mathbf{L}(\mathcal{G}^\ell)\mathbf{x}^\ell) \otimes \mathbf{e}_n^{(\ell)} = \lambda(\mathbf{x}^\ell \otimes \mathbf{e}_n^{(\ell)}) = \lambda\mathbf{x}$$

i.e. $\lambda \in \mathbf{\Lambda}(\mathbf{W})$. As a result, $\mathbf{\Lambda}(\mathbf{W}) \subseteq \bigcup_{\ell=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}^\ell))$ and $\bigcup_{\ell=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}^\ell)) \subseteq \mathbf{\Lambda}(\mathbf{W})$, therefore, $\bigcup_{\ell=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}^\ell)) = \mathbf{\Lambda}(\mathbf{W})$.

It immediately follows from Lemma 2 that $\lambda_{\max}(\mathbf{W}) = \max_{1\leq\ell\leq n} \lambda_{\max}(\mathbf{L}(\mathcal{G}^\ell))$ and $\lambda_{\min}^{+}(\mathbf{W}) = \min_{1\leq\ell\leq n} \lambda_{\min}^{+}(\mathbf{L}(\mathcal{G}^\ell))$. Thus, we have

$$\chi(\mathbf{W}) = \frac{\max_{1\leq\ell\leq n} \lambda_{\max}(\mathbf{L}(\mathcal{G}^\ell))}{\min_{1\leq\ell\leq n} \lambda_{\min}^{+}(\mathbf{L}(\mathcal{G}^\ell))}. \tag{5}$$

Concerning the example in Figure 1, we have $\lambda_{\max}(\mathbf{L}(\mathcal{G}^1)) = \lambda_{\min}^{+}(\mathbf{L}(\mathcal{G}^1)) = 1$, $\lambda_{\max}(\mathbf{L}(\mathcal{G}^2)) = \lambda_{\min}^{+}(\mathbf{L}(\mathcal{G}^2)) = 1$ and therefore $\chi(\mathbf{W}) = 1$.

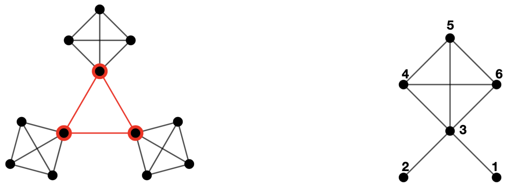
Remark 3. We can determine $\mathbf{L}(\mathcal{G})$ up to a positive multiplicative constant. Since that we can consider $\lambda_{\max}(\mathbf{L}(\mathcal{G}^\ell)) = 1$ for all $l = 1, ..., n$ without loss of generality. For that we need some preprocessing to estimate $\left\{\lambda_{\max}(\mathbf{L}(\mathcal{G}^\ell))\right\}_{l=1}^{n}$ by using Power method (see Golub and Van Loan (2013)) that could be done in a decentralized manner. We also note that $\chi(\mathbf{W})$ from (5) can be bounded from below by the largest diameter of graphs $\mathcal{G}^\ell$, $l = 1, ..., n$. Moreover, for many important classes of graphs this lower bound is tight up to a $\ln n$ factor Scaman et al. (2017).

3. ILLUSTRATIVE EXAMPLE: CYCLE OF CLIQUES

We illustrate the effect of using matrix $\mathbf{W}$ defined in (3) on a synthetic example. Our test case is a computing network with a two-level hierarchical structure. In the lower layer we have a clique with $k \geq 2$ nodes. Each node in a clique is supposed to share a local variable with its neighbors. In the upper level n cliques communicate through undirected cyclic topology: every clique has a “negotiator”, i.e. a node that has links with similar neighboring nodes (first node of every clique w.l.o.g.). We denote our graph $\mathcal{G}_{RC}$.

For every clique, we consider its union with two adjacent nodes and call such subgraph a ”crown”, see Figure 2b. Overall, we have n ”crowns” $\tilde{\mathcal{G}}^\ell$, $\ell = 1, \dots, n$, each corresponding to one of the n variables, i.e., each function $\tilde{f}_i$ held by a vertex i of $\tilde{\mathcal{G}}^\ell$ depends on x^ℓ. Further, if a node i lies in the intersection of two ”crowns”, its part $\tilde{f}_i$ of the objective depends also on the variables corresponding to each of the neighboring ”crowns”. All ”crowns” $\tilde{\mathcal{G}}^\ell$ are isomorphic and further denote the ”crown” graph as $\mathcal{G}_{cr}$.

Consider an example of graph $\mathcal{G}_{RC}$ with $n = 3$ given in Figure 2a and enumerate the vertices of the top ”crown” as shown in Figure 2b. Here, the lower lever of hierarchy are the cliques, depicted in black. The upper level of communication is an undirected ring (in red). The “crowns” can be treated as the cliques supplemented by two extra vertices and corresponding edges. Then the functions $\tilde{f}_4, \tilde{f}_5, \tilde{f}_6$ depend only on the variable x^1 since the corresponding nodes belong only to the top ”crown”. On the other hand, each of the nodes 1, 2, 3 lies in the intersection of all 3 crowns and therefore functions $\tilde{f}_1, \tilde{f}_2, \tilde{f}_3$ depend on x^1, x^2, x^3. In this section, we illustrate that matrix $\mathbf{W}$ defined in (3) has a better condition number than the Laplacian $\mathbf{L}(\mathcal{G}_{RC})$.



(a) Graph $\mathcal{G}_{RC}$ with $n = 3$ cliques of size $k = 4$

(b) Subgraph $\mathcal{G}_{cr}$ corresponding to variable x^1

Fig. 2. Hierarchical graph with cliques and its ”crown” subgraph

Theorem 4. For the hierarchical ring-clique graph it holds

$$\chi(\mathbf{W}) = \Theta(k), \ \ \chi(\mathbf{L}(\mathcal{G}_{RC})) = \Theta(n^2k^2).$$

We prove Theorem 4 in a sequence of Lemmas 7, 8, 9, 10 presented below in this section. Theorem 4 illustrates the flexibility and efficiency of our approach compared to standard approaches that do not take into account the partitioned structure of problem (1). Substituting matrix $\mathbf{W}$ instead of $\mathbf{L}(\mathcal{G}_{RC})$ allows to enhance the convergence rate of decentralized algorithms. In the following corollary, we illustrate this speedup on state-of-the-art primal and dual optimization methods.

Corollary 5. Let all the functions f_i, be L-smooth, μ strongly-convex and stored at the nodes of $\mathcal{G}_{RC}$. Consider two algorithms: dual MSDA Scaman et al. (2017) and primal OPAPC Kovalev et al. (2020). If we use $\mathbf{L}(\mathcal{G}_{RC})$, each of these methods has communication complexity $O\left(\sqrt{\frac{L}{\mu}}nk\ln\frac{1}{\varepsilon}\right)$; the communication complexity becomes $O\left(\sqrt{\frac{L}{\mu}}\sqrt{k}\ln\frac{1}{\varepsilon}\right)$ if we use $\mathbf{W}$, where ε is the accuracy.

Remark 6. Obviously, in general case there may be examples of graphs where our approach does not provide significant improvement in terms of condition numbers of the corresponding Laplacian matrices. However, the essential feature of partitioned representation of the state vector with sharing of the necessary states only makes this formulation of the distributed optimization problem attractive in terms of more sparse communication topology and reduced information exchange (as compared to (4)). It also corresponds to the preservation of the privacy of the data of interacting groups of nodes.

We first estimate $\chi(\mathbf{W})$. From Lemma 2 it follows that $\mathbf{\Lambda}(\mathbf{W}) = \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_{cr}))$, where $\mathcal{G}_{cr}$ denotes the "crown" graph. To estimate the asymptotics of the condition number of "crown" graph, i.e., the condition number of its Laplacian, we use a technique described in Pozrikidis (2014).

Lemma 7. The condition number of crown-graph has asymptotics $\chi(\mathbf{L}(\mathcal{G}_{cr})) = \Theta(k)$, where k is the clique size.

Proof. Firstly, to find the eigenvalues of $\mathbf{L}(\mathcal{G}_{cr})$, we find the eigenvalues of $\mathbf{L}(\bar{\mathcal{G}}_{cr})$, where $\bar{\mathcal{G}}_{cr}$ is the complement of $\mathcal{G}_{cr}$.

The complement has one isolated vertex (node 3 in Figure 3). We denote the Laplacian of the connected part of $\bar{\mathcal{G}}_{cr}$ (i.e. graph in Figure 3 with vertices $\{1, 2, 4, 5, 6\}$) as $\mathbf{L}'(\bar{\mathcal{G}}_{cr})$.

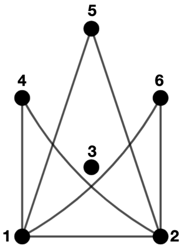


Fig. 3: Complement of $\mathcal{G}_{rc}$

$$\mathbf{L}'(\bar{\mathcal{G}}_{cr}) = \mathrm{diag}(k+2, k+2, 2, \dots, 2) \\ - \mathbf{1}_{k+1}(\mathbf{e}_{k+1}^{(1)} + \mathbf{e}_{k+1}^{(2)})^\top - (\mathbf{e}_{k+1}^{(1)} + \mathbf{e}_{k+1}^{(2)})\mathbf{1}_{k+1}^\top$$

Eigenvalues λ' of $\mathbf{L}'(\bar{\mathcal{G}}_{cr})$ are defined through the equation

$$\det\left(\mathbf{L}'\left(\bar{\mathcal{G}}_{cr}\right) - \lambda'\mathbf{I}_{k+2}\right) = 0$$

which, via linear conversions, leads us to the equation

$$\left[(k - \lambda' + 1)(2 - \lambda')^{k-2}\right] \cdot [\lambda'(\lambda' - k - 1)] = 0.$$

Thus, the eigenvalues of $\mathbf{L}'\left(\bar{\mathcal{G}}_{cr}\right)$ have the form $\lambda'_1 = 0,\ \lambda'_2 = \lambda'_3 = k+1,\ \lambda'_4 = \ldots = \lambda'_{k+2} = 2$. Using Eq. 2.2.22 from Pozrikidis (2014), we obtain the eigenvalues of $\mathbf{L}\left(\mathcal{G}_{cr}\right)$ to be $\lambda_1 = 0,\ \lambda_2 = k+2,\ \lambda_3 = \lambda_4 = 1,\ \lambda_5 = \ldots = \lambda_{k+2} = k$. This, Lemma 2 gives $\chi(\mathbf{W}) = \chi(\mathbf{L}(\mathcal{G}_{cr})) = (k+2)/1 = \Theta(k)$.

Our next goal is to estimate $\chi(\mathbf{L}(\mathcal{G}_{RC}))$ and show that it is worse than $\chi(\mathbf{W})$. To compute $\chi(\mathbf{L}(\mathcal{G}_{RC}))$, we decompose $\mathbf{L}(\mathcal{G}_{RC})$ into Laplacians of a k-clique $\mathcal{G}_C$ and a ring graph $\mathcal{G}_R$ that has n nodes. We also introduce matrix $\mathbf{B} = \mathbf{e}_k^{(1)}\mathbf{e}_k^{(1)\top}$ that allows us to write $\mathbf{L}(\mathcal{G}_{RC})$ in the following form.

$$\mathbf{L}(\mathcal{G}_{\mathbf{RC}}) = \mathbf{I_n} \otimes \mathbf{L}(\mathcal{G}_{\mathbf{C}}) + \mathbf{L}(\mathcal{G}_{\mathbf{R}}) \otimes \mathbf{B} \tag{6}$$

Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of $\mathcal{G}_R$. To obtain the eigenvalues of $\mathbf{L}(\mathcal{G}_{RC})$, we diagonalize it and decompose its spectrum. The result is formulated in the following Lemma.

Lemma 8. $\mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_{RC})) = \bigcup\limits_{i=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_C) + \lambda_i\mathbf{B})$.

Proof. Let $\mathbf{L}(\mathcal{G}_R) = \mathbf{S}^{-1}\Psi\mathbf{S}$, where $\Psi = \mathrm{diag}\left(\lambda_1, \lambda_2, \dots, \lambda_n\right)$ is a diagonal matrix with eigenvalues of $\mathbf{L}(\mathcal{G}_R)$ and $\mathbf{S}\mathbf{S}^\top = \mathbf{I}_n$. We define

$$\widehat{\mathbf{L}} = (\mathbf{S} \otimes \mathbf{I}_k)^{-1}\,\mathbf{L}(\mathcal{G}_{RC})\,(\mathbf{S} \otimes \mathbf{I}_k) = \mathbf{I}_n \otimes \mathbf{L}(\mathcal{G}_C) + \Psi \otimes \mathbf{B}.$$

Whence, $\mathbf{\Lambda}(\widehat{\mathbf{L}}) = \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_{RC}))$. Indeed, let $\mathbf{x}$ be an eigenvector of $\mathbf{L}(\mathcal{G}_{RC})$ such that $\mathbf{L}(\mathcal{G}_{RC})\mathbf{x} = \theta\mathbf{x}$. Then

$$\widehat{\mathbf{L}} \cdot ((\mathbf{S} \otimes \mathbf{I}_k)^{-1}\mathbf{x}) = \theta \cdot ((\mathbf{S} \otimes \mathbf{I}_k)^{-1}\mathbf{x}),$$

i.e., $(\mathbf{S} \otimes \mathbf{I}_k)\mathbf{x}$ is an eigenvector of $\widehat{\mathbf{L}}$.

Further, for any eigenvalue θ of $\widehat{\mathbf{L}}$, we have

$$\widehat{\mathbf{L}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} (\mathbf{L}(\mathcal{G}_C) + \lambda_1\mathbf{B})\,x_1 \\ \vdots \\ (\mathbf{L}(\mathcal{G}_C) + \lambda_n\mathbf{B})\,x_n \end{bmatrix} = \theta \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

i.e., $(\mathbf{L}(\mathcal{G}_C) + \lambda_i\mathbf{B})x_i = \theta x_i$ for $i = 1, \dots, n$. Consequently, $\mathbf{\Lambda}(\widehat{\mathbf{L}}) = \bigcup\limits_{i=1}^{n} \mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_C) + \lambda_i\mathbf{B})$, which concludes the proof.

Due to Lemma 8, we only have to find the spectrums $\mathbf{\Lambda}(\mathbf{L}(\mathcal{G}_C) + \lambda_i\mathbf{B})$ for $i = 1, \dots, n$. We do this by applying the matrix determinant lemma.

Lemma 9. The matrix $\mathbf{L}(\mathcal{G}_C) + \lambda\mathbf{B}$ has the following eigenvalues.

$$\theta_1(\lambda) = \frac{k + \lambda + \sqrt{\lambda^2 + 2(k-2)\lambda + k^2}}{2}, \tag{7a}$$

$$\theta_2(\lambda) = \frac{k + \lambda - \sqrt{\lambda^2 + 2(k-2)\lambda + k^2}}{2}, \tag{7b}$$

$$\theta_3(\lambda) = \ldots = \theta_n(\lambda) = k.$$

Proof. Let θ denote some eigenvalue of $\mathbf{L}(\mathcal{G}_C) + \lambda\mathbf{B}$. Then, $\mathbf{L}(\mathcal{G}_C) + \lambda\mathbf{B} - \theta\mathbf{I}_k = \mathbf{A} - \mathbf{1}_k\mathbf{1}_k^\top$, where $\mathbf{A} = \mathrm{diag}\left(k + \lambda - \theta, k - \theta, \dots, k - \theta\right)$.

Firstly, note that $\theta = k$ is an eigenvalue of $\mathbf{L}(\mathcal{G}_C) + \lambda\mathbf{B}$. Indeed, substituting $\theta = k$, we obtain

$$\mathbf{A} - \mathbf{1}_k\mathbf{1}_k^\top = \begin{bmatrix} \lambda\text{-}1 & \text{-}1 & \cdots & \text{-}1 \\ \text{-}1 & \text{-}1 & \cdots & \text{-}1 \\ \vdots & \vdots & \ddots & \vdots \\ \text{-}1 & \text{-}1 & \cdots & \text{-}1 \end{bmatrix},$$

which has determinant equal to 0.

Further, we assume that $\theta \neq k$. According to the matrix determinant lemma we have

$$\begin{aligned} \det\left(\mathbf{A} - \mathbf{1}_k\mathbf{1}_k^\top\right) &= \left(1 - \mathbf{1}^\top \cdot \mathbf{A}^{-1} \cdot \mathbf{1}\right) \cdot \det\mathbf{A} \\ &= \left(1 - \frac{1}{k + \lambda - \theta} - \frac{k-1}{k - \theta}\right) \cdot (k + \lambda - \theta)(k - \theta)^{k-1} \\ &= (k - \theta)^{k-2}(\theta - \theta_1)(\theta - \theta_2) \end{aligned}$$

where θ_1 and θ_2 are defined in (7).

Equation (7) gives explicit formulas for eigenvalues of $\mathbf{L}(\mathcal{G}_{RC})$, and it remains to estimate $\lambda_{\max}(\mathbf{L}(\mathcal{G}_{RC}))$ and $\lambda^+_{\min}(\mathbf{L}(\mathcal{G}_{RC}))$. The eigenvalues of the ring graph $\mathcal{G}_R$ have the form $\lambda_i = 2 - 2\cos\frac{2\pi i}{n}$. Note that $0 \le \lambda_i \le 4$.

For the largest eigenvalue of $\mathbf{L}(\mathcal{G}_{RC})$, we have

$$\lambda_{\max}(\mathbf{L}(\mathcal{G}_{RC})) \le \frac{k + 4 + \sqrt{4 + 2(k-2)\cdot 2 + k^2}}{2} = \Theta(k)$$

Estimating $\lambda^+_{\min}(\mathbf{L}(\mathcal{G}_{RC}))$ is less straightforward and we need to compute the minimal θ_2 defined in (7b) over $\lambda \in \{\lambda_1, \dots, \lambda_n\}$.

Lemma 10. It holds that $\lambda^+_{\min}(\mathbf{L}(\mathcal{G}_{RC})) = \Theta(\frac{1}{n^2 k})$.

Proof. Firstly, let us show that $\theta_2(\lambda)$ defined in (7b) is monotonically increasing by considering its derivative:

$$\frac{d\theta_2(\lambda)}{d\lambda} = \frac{-k - \lambda + 2}{2\sqrt{k^2 + \lambda^2 + 2(k-2)\lambda}} + \frac{1}{2}$$

Since $k \geq 2$, $k + \lambda - 2 < \sqrt{k^2 + \lambda^2 + 2(k-2)\lambda}$, and we have $\frac{d\theta_2(\lambda)}{d\lambda} > 0$. Since $\theta_2(0) = 0$, the minimal positive eigenvalue of $\mathbf{L}(\mathcal{G}_{RC})$ is reached at minimal positive λ, that is, $\lambda^+_{\min}(\mathcal{G}_{RC}) = \theta_2(2 - 2\cos\frac{2\pi}{n})$.

Secondly, we approximate $\theta_2(2 - 2\cos\frac{2\pi}{n})$ using the Taylor series at $n \to \infty$ and get

$$\theta_2\left(2 - 2\cos\frac{2\pi}{n}\right) = \frac{4\pi^2}{n^2 k} + o\left(\frac{1}{n^3}\right).$$

It follows that $\lambda^+_{\min}(\mathcal{G}_{RC}) = \Theta(\frac{1}{n^2 k})$.

REFERENCES

Borkar, V. and Varaiya, P.P. (1982). Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, 27(3), 650–655.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.

Cannelli, L., Facchinei, F., Scutari, G., and Kungurtsev, V. (2020). Asynchronous optimization over graphs: Linear convergence under error bound conditions. *IEEE Transactions on Automatic Control*, 66(10), 4604–4619.

Carli, R. and Notarstefano, G. (2013). Distributed partition-based optimization via dual decomposition. In *52nd IEEE Conference on Decision and Control*, 2979–2984. IEEE.

DeGroot, M.H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, 69(345), 118–121.

Dvinskikh, D. and Gasnikov, A. (2021). Decentralized and parallel primal and dual accelerated methods for stochastic convex programming problems. *Journal of Inverse and Ill-posed Problems.*

Erseghe, T. (2012). A distributed and scalable processing method based upon admm. *IEEE Signal Processing Letters*, 19(9), 563–566.

Golub, G.H. and Van Loan, C.F. (2013). *Matrix computations.* JHU press.

Gorbunov, E., Rogozin, A., Beznosikov, A., Dvinskikh, D., and Gasnikov, A. (2020). Recent theoretical advances in decentralized distributed convex optimization. *arXiv preprint arXiv:2011.13259.*

Ivanova, A., Dvurechensky, P., Gasnikov, A., and Kamzolov, D. (2020). Composite optimization for the resource allocation problem. *Optimization Methods and Software*, 0(0), 1–35. doi:10.1080/10556788.2020.1712599.

Kekatos, V. and Giannakis, G.B. (2012). Distributed robust power system state estimation. *IEEE Transactions on Power Systems*, 28(2), 1617–1626.

Kovalev, D., Beznosikov, A., Sadiev, A., Persiianov, M., Richtárik, P., and Gasnikov, A. (2022). Optimal algorithms for decentralized stochastic variational inequalities. *arXiv preprint arXiv:2202.02771.*

Kovalev, D., Salim, A., and Richtárik, P. (2020). Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33.

Kraska, T., Talwalkar, A., Duchi, J.C., Griffith, R., Franklin, M.J., and Jordan, M.I. (2013). Mlbase: A distributed machine-learning system. In *CIDR*, volume 1, 2–1.

Kroshnin, A., Tupitsa, N., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., and Uribe, C. (2019). On the complexity of approximating Wasserstein barycenters. In K. Chaudhuri and R. Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *PMLR*, 3530–3540. PMLR, Long Beach, California, USA.

Necoara, I. and Clipici, D. (2016). Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds. *SIAM Journal on Optimization*, 26(1), 197–226.

Nedić, A., Olshevsky, A., and Uribe, C.A. (2017). Fast convergence rates for distributed non-Bayesian learning. *IEEE Trans. on Autom. Contr.*, 62(11), 5538–5553.

Nedić, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J.N. (2009). On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11), 2506–2517.

Notarnicola, I., Carli, R., and Notarstefano, G. (2017). Distributed partitioned big-data optimization via asynchronous dual decomposition. *IEEE Transactions on Control of Network Systems*, 5(4), 1910–1919.

Pozrikidis, C. (2014). *An introduction to grids, graphs, and networks.* Oxford University Press.

Rabbat, M. and Nowak, R. (2004). Decentralized source localization and tracking wireless sensor networks. In *Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, 921–924.

Ram, S.S., Veeravalli, V.V., and Nedic, A. (2009). Distributed non-autonomous power control through distributed convex optimization. In *IEEE INFOCOM 2009*, 3001–3005. IEEE.

Rogozin, A., Beznosikov, A., Dvinskikh, D., Kovalev, D., Dvurechensky, P., and Gasnikov, A. (2021). Decentralized distributed optimization for saddle point problems. *arXiv preprint arXiv:2102.07758.*

Scaman, K., Bach, F., Bubeck, S., Lee, Y.T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In D. Precup and Y.W. Teh (eds.), *ICML*, volume 70 of *PMLR*, 3027–3036. PMLR.

Tsitsiklis, J.N. and Athans, M. (1984). Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1), 42–50.

Uribe, C.A., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., and Nedić, A. (2018). Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE Conference on Decision and Control (CDC)*, 6544–6549.

Xiao, L. and Boyd, S. (2006). Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3), 469–488.

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