Decentralized Strongly-Convex Optimization with Affine Constraints: Primal and Dual Approaches[⋆]

Alexander Rogozin[1], Demyan Yarmoshik[1], Ksenia Kopylova[2], and Alexander Gasnikov[1,3,4]

[1] Moscow Institute of Physics and Technology, Moscow, Russia
[2] Saint Petersburg State University, Saint Petersburg, Russia
[3] Caucasus Mathematical Center, Adyghe State University, Maikop, Russia
[4] IITP RAS, Moscow, Russia

Abstract. Decentralized optimization is a common paradigm used in distributed signal processing and sensing as well as privacy-preserving and large-scale machine learning. It is assumed that several computational entities locally hold objective functions and are connected by a network. The agents aim to commonly minimize the sum of the local objectives subject by making gradient updates and exchanging information with their immediate neighbors. Theory of decentralized optimization is pretty well-developed in the literature. In particular, it includes lower bounds and optimal algorithms. In this paper, we assume that along with an objective, each node also holds affine constraints. We discuss several primal and dual approaches to decentralized optimization problem with affine constraints.

Keywords: distributed optimization, convex optimization, constrained optimization

1 Introduction

Many distributed systems such as distributed sensor networks, systems for power flow control and large-scale architectures for machine learning use decentralized optimization as a basic mathematical tool. Several applications such as power systems control [11,17] lead to problems where the agents locally hold optimization objectives and aim to cooperatively minimize the sum of the objectives. Moreover, every node locally holds affine constraints for its decision variable.

Decentralized optimization without affine constraints can be called a well-examined area of research. It is known that the performance of optimization algorithms executed over strongly-convex smooth objectives is lower bounded by

---

a multiple of the graph condition number and objective condition number (up to a logarithmic factor) [19]. Both primal [8] and dual [19] algorithms that reach the lower bounds have been proposed. The algorithms are based on reformulating network communication constraints as affine constraints via a communication matrix associated with the network (i.e. Laplacian matrix). Introduction of affine constraints at the nodes leads to new classes of algorithms that can be divided into two main types. The first type are consensus-based methods that can be either primal or dual [2,10,23,9,12,13,14]. The second type are ADMM-based methods [1,18,3,21]. Let us briefly review some of the closely related papers.

The paper [12] is dedicated to constrained distributed optimization and consider only separable objective functions (each agent has its own independent variable). Moreover, affine constraints are supposed to be network-compatiable (constraint matrix can have a non-zero element on position (i, j) only if there is an edge in communication graph between agents i and j). We do not impose such limitations: in our case each term in the objective functions depends on the same shared variable (formulation in [12] is obviously a special case of this) and matrix of constraints can have arbitrary structure.

In [14] the authors present various formulations of distributed optimization problems with different types of interconnections between constraints and objectives, including the case, when the objective (cost) cannot be represented as sum of cost functions of each agent. However, their algorithms for problems with coupled affine constraints require to solve a "master problem" on central node at each iteration and thus are not decentralized.

The authors of [20] consider multi-cluster distributed problem formulation which is a generalization of multi-agent approach. In multi-cluster case agents within one cluster have the same decision variable while different clusters corresponds to different decision variables. All variables are subject to a coupled affine constraint. By incorporating consensus constraints into dual problem with Lagrangian multipliers the author comes to solving a saddle point problem and prove asymptotic $O(1/N)$ ergodic convergence rate for their method. Dependency of convergence rate on problem parameters in saddle point approach was studied in [22].

Our paper studies the application of different techniques to decentralized problems with affine constraints. We obtain linear convergence rates with (explicitly specified) accelerated dependencies on function properties, constraint matrix spectrum and communication graph properties.

The paper outline is as follows. In Section 4 we discuss a primal approach, that is based on reformulation the initial distributed problem as a saddle-point problem and applying algorithm of paper [7] afterwards. In Section 5, we describe a method that allows to incorporate both affine and communication constraints to the dual function. We refer the approach in Section 5 as a globally dual approach. Finally, in Section 6 we describe a slightly different dual approach that firstly takes dual functions locally at the nodes and incorporates consensus constraints afterwards. We refer to the latter method as a locally dual approach.

2   Preliminaries

Let $\mathrm{col}(x_1, \ldots, x_m)$ define a column vector of $x_1, \ldots, x_m \in \mathbb{R}^d$, i.e. $\mathrm{col}(x_1, \ldots, x_m) = [x_1^\top \ldots x_m^\top]^\top$. For matrices P and Q, their Kronecker product is defined as $P \otimes Q$. Identity matrix of size $p \times p$ is denoted $\mathbf{I}_p$. Moreover, given a symmetric positive semi-definite matrix, we denote $\lambda_{\max}(\cdot)$, $\lambda_{\min}(\cdot)$, $\lambda_{\min}^+(\cdot)$ its maximal, minimal and minimal nonzero eigenvalues, respectively. We also let $\sigma_{\max}(\cdot)$, $\sigma_{\min}(\cdot)$ and $\sigma_{\min}^+(\cdot)$ be the maximal, minimal and minimal nonzero singular values of a matrix, respectively.

In the forthcoming analysis, we will need the following basic lemma concerning Kronecker product properties.

Lemma 1. *Given two matrices P and Q such that $\sigma_{\min}(P) = \sigma_{\min}(Q) = 0$, we have*

$$\sigma_{\max}(P \otimes \mathbf{I} + \mathbf{I} \otimes Q) = \sigma_{\max}(P) + \sigma_{\max}(Q),$$
$$\sigma_{\min}^+(P \otimes \mathbf{I} + \mathbf{I} \otimes Q) = \min\left\{\sigma_{\min}^+(P), \sigma_{\min}^+(Q)\right\}$$

Proof. Consider decompositions $P = U_P \Sigma_P V_P^\top$ and $Q = U_Q \Sigma_Q V_Q^\top$, where U_P, V_P, U_Q, V_Q are orthogonal matrices and Σ_P and Σ_Q are diagonal matrices with corresponding eigenvalues at the diagonal. We have

$$(U_P^\top \otimes U_Q^\top)(P \otimes \mathbf{I} + \mathbf{I} \otimes Q)(V_P \otimes V_Q) = \Sigma_P \otimes \mathbf{I} + \mathbf{I} \otimes \Sigma_Q.$$

Denote singular values of P as $\alpha_1, \ldots, \alpha_n$ and the singular values of Q as $\beta_1, \ldots, \beta_m$. Singular values of $P \otimes \mathbf{I} + \mathbf{I} \otimes Q$ have form

$$\lambda(\alpha_i, \beta_j) = \alpha_i + \beta_j, \ i = 1, \ldots, n, \ j = 1, \ldots, m.$$

Therefore, $\sigma_{\max}(P \otimes \mathbf{I} + \mathbf{I} \otimes Q) = \sigma_{\max}(P) + \sigma_{\max}(Q)$. For the minimal nonzero singular values we obtain

$$\sigma_{\min}^+(P \otimes \mathbf{I} + \mathbf{I} \otimes Q) = \min\left\{\sigma_{\min}^+(P), \sigma_{\min}^+(Q)\right\}.$$

3   Problem Statement

Consider minimization problem with affine constraints.

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^m f_i(x) \ \text{ s.t. } Bx = 0. \tag{1}$$

We assume that each f_i is held by a separate agent, and the agents can exchange information through some communication network. Each agent also locally holds affine optimization constraints $Bx = 0$, where $B \in \mathbb{R}^{p \times d}$. Further we assume that $\mathrm{Ker}\, B \neq \{0\}$, because otherwise the constraints $Bx = 0$ define a set consisting of only $\{0\}$, which is not an interesting case.

We make assumptions on the optimization objectives that are standard for optimization literature [16].

Assumption 1 *Each f_i $(i = 1, \ldots, m)$ is differentiable, μ-strongly convex and L-smooth, i.e.*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|_2^2,$$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|_2^2.$$

The communication network is represented by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The communication constraints are represented by a specific matrix W associated with the graph $\mathcal{G}$.

Assumption 2
1. *W is a symmetric positive semi-definite matrix.*
2. *(Network compatibility) For all $i, j = 1, \ldots, m$ it holds $[W]_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and $i \neq j$.*
3. *(Kernel property) For any $v = [v_1, \ldots, v_m]^\top \in \mathbb{R}^m$, $Wv = 0$ if and only if $v_1 = \ldots = v_m$, i.e. $\operatorname{Ker} W = \operatorname{span} \{\mathbf{1}\}$.*

An explicit example of a matrix that satisfies Assumption 2 is the Graph Laplacian $W \in \mathbb{R}^{m \times m}$:

$$[W]_{ij} \triangleq \begin{cases} -1, & \text{if } (i, j) \in E, \\ \deg(i), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let us introduce $\mathbf{x} = \operatorname{col}(x_1 \ldots x_m)$ and $\mathbf{W} = W \otimes \mathbf{I}$. According to Assumption 2, communication constraints $x_1 = \ldots = x_m$ can be equivalently rewritten as $\mathbf{W}\mathbf{x} = 0$. Also introduce $\mathbf{B} = \mathbf{I} \otimes B$ and $F(\mathbf{x}) = \sum_{i=1}^m f_i(x_i)$. That allows to rewrite problem (1) in the following way.

$$\min_{\mathbf{x} \in \mathbb{R}^{md}} F(\mathbf{x}) \tag{3}$$
$$\text{s.t. } \mathbf{W}\mathbf{x} = 0, \ \mathbf{B}\mathbf{x} = 0.$$

Reformulation 3 admits implementation of optimization methods for affinely constrained minimization. The iterations of such methods become automatically decentralized in the following sense. Let the optimization algorithm use primal or dual oracle calls of the objective function and use multiplications by the matrices representing affine constraints. In the case of problem (3) the gradient $\nabla F(\mathbf{x}) = \operatorname{col}[\nabla f_1(x_1) \ldots \nabla f_m(x_m)]$ is computed locally on the nodes and stored in a distributed manner across the network. Multiplication by $\mathbf{B}$ is also performed locally due to its definition (i.e. the i-th node computes Bx_i), and the multiplication by $\mathbf{W}$ is performed in a decentralized manner due to the network compatibility property of W (see Assumption 2).

4 Primal Approach

In this section, we discuss the solution of problem (3) by an algorithm APDG [7] that only uses primal oracle calls. The algorithm is designed for saddle-point problems, so we reformulate (3) as a saddle-point problem.

We add dual multipliers for the constraints and get a saddle-point problem

$$\min_{\mathbf{x}\in\mathbb{R}^{md}} \max_{\mathbf{u}\in\mathbb{R}^{mp},\mathbf{v}\in\mathbb{R}^{md}} F(\mathbf{x}) + \langle\mathbf{u},\mathbf{B}\mathbf{x}\rangle + \gamma\langle\mathbf{v},\mathbf{W}\mathbf{x}\rangle = F(\mathbf{x}) + \left\langle \begin{pmatrix}\mathbf{u}\\\mathbf{v}\end{pmatrix}, \begin{pmatrix}\mathbf{B}\\\gamma\mathbf{W}\end{pmatrix}\mathbf{x} \right\rangle. \tag{4}$$

Algorithm 1 APDG: Accelerated Primal-Dual Gradient Method

1: **Input:** $\mathbf{x}^0 \in \text{Range}\,\mathbf{A}^\top, \mathbf{y}^0 \in \text{Range}\,\mathbf{A},\ \eta_x,\eta_y,\alpha_x,\beta_x,\beta_y > 0,\ \tau_x,\tau_y,\sigma_x,\sigma_y \in (0,1]$,
$\quad\ \theta \in (0,1)$
2: $\quad \mathbf{x}_f^0 = \mathbf{x}^0$
3: $\quad \mathbf{y}_f^0 = \mathbf{y}^{-1} = \mathbf{y}^0$
4: **for** $k = 0,1,2,\ldots$ **do**
5: $\quad\quad \mathbf{y}_m^k = \mathbf{y}^k + \theta(\mathbf{y}^k - \mathbf{y}^{k-1})$
6: $\quad\quad \mathbf{x}_g^k = \tau_x\mathbf{x}^k + (1-\tau_x)\mathbf{x}_f^k$
7: $\quad\quad \mathbf{y}_g^k = \tau_y\mathbf{y}^k + (1-\tau_y)\mathbf{y}_f^k$
8: $\quad\quad \mathbf{x}^{k+1} = \mathbf{x}^k + \eta_x\alpha_x(\mathbf{x}_g^k - \mathbf{x}^k) - \eta_x\beta_x\mathbf{A}^\top\mathbf{A}\mathbf{x}^k - \eta_x\left(\nabla F(\mathbf{x}_g^k) + \mathbf{A}^\top\mathbf{y}_m^k\right)$
9: $\quad\quad \mathbf{y}^{k+1} = \mathbf{y}^k - \eta_y\beta_y\mathbf{A}(\mathbf{A}^\top\mathbf{y}^k + \nabla F(\mathbf{x}_g^k)) + \eta_y\mathbf{A}\mathbf{x}^{k+1}$
10: $\quad\quad \mathbf{x}_f^{k+1} = \mathbf{x}_g^k + \sigma_x(\mathbf{x}^{k+1} - \mathbf{x}^k)$
11: $\quad\quad \mathbf{y}_f^{k+1} = \mathbf{y}_g^k + \sigma_y(\mathbf{y}^{k+1} - \mathbf{y}^k)$
12: **end for**

Denote $\mathbf{A} = \begin{pmatrix}\mathbf{B}\\\gamma\mathbf{W}\end{pmatrix}$. In order to get complexity bounds for APDG applied to problem (4), we need to bound the spectrum of $\mathbf{A}$. Note that $\mathbf{A}^\top\mathbf{A} = \mathbf{B}^\top\mathbf{B} + \gamma^2\mathbf{W}^2 = \mathbf{I}_m \otimes (B^\top B) + \gamma^2 W^2 \otimes \mathbf{I}_d$. By Lemma 1 we have

$$\lambda_{\max}(\mathbf{A}^\top\mathbf{A}) = \lambda_{\max}(B^\top B) + \gamma^2\lambda_{\max}^2(W),$$
$$\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A}) = \min\left\{\lambda_{\min}^+(B^\top B), \gamma^2(\lambda_{\min}^+(W))^2\right\}.$$

We can also compute the condition number of $\mathbf{A}^\top\mathbf{A}$:

$$\chi(\mathbf{A}^\top\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})} = \frac{\lambda_{\max}(B^\top B) + \gamma^2\lambda_{\max}^2(W)}{\min\left\{\lambda_{\min}^+(B^\top B), \gamma^2(\lambda_{\min}^+(W))^2\right\}}.$$

By accurately choosing factor γ, we can control the condition number $\chi(\mathbf{A}^\top\mathbf{A})$. The minimal value of $\chi(\mathbf{A}^\top\mathbf{A})$ is attained at $\gamma^2 = \frac{\lambda_{\min}^+(B^\top B)}{(\lambda_{\min}^+(W))^2}$ and equals $\chi(\mathbf{A}^\top\mathbf{A}) = \chi(B^\top B) + \chi^2(W)$. Therefore, if we apply APDG directly to problem (3), the

complexity would be

$$O\left(\max\left(\sqrt{\chi^2(W)+\chi(B^\top B)}\sqrt{\frac{L}{\mu}}, \chi^2(W)+\chi(B^\top B)\right)\log\frac{1}{\varepsilon_{\mathbf{x}}}\right)$$

calls of $\nabla f_i(\cdot)$ at each node and communication rounds, with $\varepsilon_{\mathbf{x}}$ being the desired distance to the solution: $\|\mathbf{x}^N - \mathbf{x}^*\| \le \varepsilon_{\mathbf{x}}$. In the smooth, strongly convex case it is also the complexity for satisfying $F(\mathbf{x}^N) - F(\mathbf{x}) \le \varepsilon_F$ or $\|\mathbf{A}\mathbf{x}^N\| \le \varepsilon_{\mathbf{A}}$ (up to logarithmic dependencies on the problem parameters). Indeed, from Lipschitz smoothness we have $F(\mathbf{x}^N) - F(\mathbf{x}) \le L\varepsilon_{\mathbf{x}}^2/2$ and $\|\mathbf{A}\mathbf{x}^N\| = \|\mathbf{A}\mathbf{x}^N - \mathbf{A}\mathbf{x}^*\| \le \sigma_{max}(\mathbf{A})\varepsilon_{\mathbf{x}}$. By that means, in the following inequalities ε can be replaced by any of $\varepsilon_{\mathbf{x}}$, ε_f, $\varepsilon_{\mathbf{A}}$.

The dependence on network parameters W and affine constraints parameters B can be enhanced by using Chebyshev acceleration [19]. Let us replace W by a Chebyshev polynomial $P_K(W)$ such that it has degree $K = O\left(\sqrt{\chi(W)}\right)$ and condition number $\chi(P_K(W)) = O(1)$. Multiplication by $P_K(W)$ is equivalent to making K communication rounds. Analogically, let us replace $B^\top B$ by a Chebyshev polynomial $P_M(B^\top B)$ with degree $M = O\left(\sqrt{\chi(B^\top B)}\right)$ and condition number $\chi\left(P_M(B^\top B)\right) = O(1)$. As a result, we obtain

$$N = O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right) \quad \text{oracle calls at each node,}$$

$$O\left(N\sqrt{\chi(W)}\right) \quad \text{communications,}$$

$$O\left(N\sqrt{\chi(B^\top B)}\right) \quad \text{multiplications by } B,\ B^\top \text{ at each node.}$$

5 Globally Dual Approach

In this section, we describe an approach to solving (3) that is based on passing to the dual problem. We call this approach "global" since both constraints, that is, affine constraints $\mathbf{B}\mathbf{x} = 0$ and communication constraints $\mathbf{W}\mathbf{x} = 0$ are used in the dual reformulation.

Let γ be a positive scalar and $\mathbf{A}^\top = [\mathbf{B}^\top\ \gamma\mathbf{W}]$ and introduce dual function

$$\Phi(\mathbf{y}) = \max_{\mathbf{x}\in\mathbb{R}^{md}}\left[-F(\mathbf{x}) + \langle\mathbf{y}, \mathbf{A}\mathbf{x}\rangle\right] = F^*(\mathbf{A}^\top\mathbf{y}).$$

We have $\nabla\Phi(\mathbf{y}) = \mathbf{A}\nabla F^*(\mathbf{A}^\top\mathbf{y}) = \mathbf{A}\cdot\arg\min_{\mathbf{x}\in\mathbb{R}^{md}}\left[-F(\mathbf{x}) + \langle\mathbf{y}, \mathbf{A}\mathbf{x}\rangle\right]$. Note that multiplication by $\mathbf{A}$ is performed in a distributed manner: indeed, it includes local multiplications by B and a consensus round, which is a multiplication by $\mathbf{W}$. Moreover, the $\arg\min$ operation is computed locally, which is standard for decentralized optimization [19]. Finally, dual function Φ is $\frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}{\mu}$-smooth

on $\mathbb{R}^{m(p+d)}$ and $L_\Phi = \frac{\lambda^+_{\min}(\mathbf{A}^\top\mathbf{A})}{L}$-strongly convex on $(\operatorname{Ker}\mathbf{A}^\top)^\perp$. Solving dual problem

$$\min_{\mathbf{y}\in\mathbb{R}^{m(p+d)}}\ \Phi(\mathbf{y})$$

by a fast gradient method (see i.e. accelerated Nesterov method in Section 2.2 of [16]) until accuracy $\Phi(\mathbf{y}^N)-\Phi(\mathbf{y})\le\varepsilon_\Phi$ requires $N = O\left(\sqrt{\frac{L}{\mu}}\sqrt{\chi(\mathbf{A}^\top\mathbf{A})}\log\frac{1}{\varepsilon_\Phi}\right)$ iterations.

Following the same arguments as in Section 4, we compute the condition number $\chi(\mathbf{A}^\top\mathbf{A})$:

$$\chi(\mathbf{A}^\top\mathbf{A}) = \frac{\lambda_{\max}(B^\top B) + \gamma^2\lambda^2_{\max}(W)}{\min\left\{\lambda^+_{\min}(B^\top B), \gamma^2(\lambda^+_{\min}(W)^2)\right\}}.$$

The minimal value of $\chi(\mathbf{A}^\top\mathbf{A})$ is attained at $\gamma^2 = \frac{\lambda^+_{\min}(B^\top B)}{(\lambda^+_{\min}(W))^2}$ and equals $\chi(\mathbf{A}^\top\mathbf{A}) = \chi(B^\top B) + \chi^2(W)$. Communication and computation complexities of fast dual method equal

$$O\left(\sqrt{\frac{L}{\mu}}\left(\chi(B^\top B) + \chi^2(W)\right)^{\frac{1}{2}}\log\frac{1}{\varepsilon_\Phi}\right).$$

To obtain desired complexity estimates for the algorithm to find the approximate solution $\mathbf{x}^N$ satisfying $F(\mathbf{x}^N) - F(\mathbf{x}) \le \varepsilon$ and $\|\mathbf{A}\mathbf{x}^N\| \le \varepsilon$, we refer to the following properties of dual function (see, e.g. Theorem 5.2 from [4]):

$$\|\nabla\Phi(\mathbf{y})\| \le \epsilon/R_\mathbf{y} \Rightarrow F(\mathbf{x}(\mathbf{y})) - F(\mathbf{x}^*) \le \epsilon,$$
$$\|\nabla\Phi(\mathbf{y})\| \le \epsilon \Rightarrow \|\mathbf{A}\mathbf{x}(\mathbf{y})\| \le \epsilon,$$

where $\|\mathbf{y}\| \le 2R_\mathbf{y}$, and $\mathbf{x}(\mathbf{y}) = \arg\min_{\mathbf{x}\in\mathbb{R}^{md}}\left[-F(\mathbf{x}) + \langle\mathbf{y}, \mathbf{A}\mathbf{x}\rangle\right]$. Combining it with $\Phi(\mathbf{y}^N)-\Phi(\mathbf{y}) \ge \|\Phi(\mathbf{y}^N)\|^2/2L_\Phi$, which is true for a smooth convex function, we justify substitution of ε_Φ by ε in the complexity estimate. This transition will only change the constant hidden by big-O notation (by the factor of two), and affect omitted logarithmic dependencies on the problem parameters.

To employ Chebyshev acceleration in this case we do substitution $\mathbf{A}^\top\mathbf{y} \to \mathbf{p}$. In this variables accelerated Nesterov method turns into Algorithm 2, where $\mathbf{x}(\mathbf{q}) = \nabla F^*(\mathbf{q}) = \arg\min\left[-F(\mathbf{x}) + \langle\mathbf{q}, x\rangle\right]$:

Algorithm 2 Globally Dual Method

1: **Input: $\mathbf{p}^0 \in \operatorname{Range}\mathbf{A}^\top$, $\eta > 0$, $\beta \in (0,1)$**
2: $\mathbf{p}^{-1} = \mathbf{p}^0$
3: **for** $k = 0, 1, 2, \ldots$ **do**
4:    $\mathbf{q} = \mathbf{p}^k + \beta\left(\mathbf{p}^k - \mathbf{p}^{k-1}\right)$
5:    $\mathbf{p}^{k+1} = \mathbf{q} - \eta\mathbf{A}^\top\mathbf{A}\mathbf{x}(\mathbf{q})$
6: **end for**

For the algorithm in this form we can replace $\mathbf{A}^\top \mathbf{A}$ with Chebyshev polynomial of it, as we did in Section 4, and obtain the same complexity estimates as for APDG:

$$N = O\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon}\right) \text{ oracle calls at each node,}$$

$$O\left(N\sqrt{\chi(W)}\right) \text{ communications,}$$

$$O\left(N\sqrt{\chi(B^\top B)}\right) \text{ multiplications by } B, B^\top \text{ at each node.}$$

6   Locally Dual approach

In Section 5 we discussed a dual reformulation of (3) where both constraints $\mathbf{Bx} = 0$ and $\mathbf{Wx} = 0$ are used simultaneously. This section describes a dual approach, as well, but the difference is that we firstly pass to dual functions locally at the nodes and impose the communication constraints only afterwards.

6.1   Utilizing locality on u

One can note that in the above approaches optimization over $\mathbf{u}$ could be done locally at each node. This is equivalent to including affine constraints into the objective (as an indicator function) instead of handling them with Lagrangian multipliers. In settings there the "cost" of communication is limiting or comparable to that of local computations, we can find the solution faster by going this way. It may be the case when x has a small dimension and decentralization is desirable due to privacy constraints.

Dual problem in this approach will be

$$\max_{\mathbf{v}} \min_{\mathbf{Bx}=0} \{F(\mathbf{x}) + \langle \mathbf{v}, \mathbf{Wx} \rangle\} = -\min_{\mathbf{v}} F^*_{[\mathbf{Bx}=0]}(\mathbf{W}^\top \mathbf{v}),$$

where $F^*_{[\mathbf{Bx}=0]}(\mathbf{v}) = \max_{\mathbf{Bx}=0}\{\langle \mathbf{v}, \mathbf{x} \rangle - F(\mathbf{x})\}$ denotes a convex conjugate under affine constraints.

We can reduce the problem of computing the gradient of such a modified conjugate function to calling conventional dual oracle. Let E be a matrix, the rows of which constitute an orthogonal basis in the null space of B (matrix E can be computed at the preprocessing stage of an optimization algorithm). Then instead of working with functions $f_i(x)$ we can optimize the sum of functions $h_i(t) = f_i(Et)$.

Denote $\mathbf{t} = \text{col}(t_1, \ldots, t_m)$, $H(\mathbf{t}) = \sum_{i=1}^m h_i(t_i)$. Then problem (1) could be written in decentralized way as follows

$$\min_{\mathbf{t}} \sum_{i=1}^{m} h_i(t_i)$$
$$\text{s.t. } \mathbf{W_t t} = 0.$$

Its dual form is

$$\max_{\mathbf{t}}\{\langle \mathbf{z}, \mathbf{W_t t}\rangle - H(\mathbf{t})\} = -\min_{\mathbf{z}} H^*(\mathbf{W_t}^\top \mathbf{z}),$$

and the gradient of the objective can be computed using Demyanov–Danskin's theorem:

$$\nabla H^*(\mathbf{z}) = \arg\max_{\mathbf{t}}\{\langle \mathbf{z}, \mathbf{t}\rangle - F(\mathbf{Et})\}.$$

From smaller dimension of t comparing to x we can expect that computation of $\nabla H^*(\mathbf{z})$ is easier than calling conventional first-order dual oracle, the only drawback is the necessity of storing matrix E and performing multiplications by E.

Let μ_t and L_t be the constants of strong convexity and Lipschitz smoothness of h_i respectively for all $i = 1, \ldots, m$. Then, obviously, $\mu_t \geq \mu$ and $L_t \leq L$. For example, if $f_i(x)$ is twice continuously differentiable, then its smoothness constant can be computed as $L_{x,i} = \sup_{x \in \mathbb{R}^n} \lambda_{\max}(\nabla^2 f_i(x))$. The smoothness constant of $h_i(t)$ is given by $L_{t,i} = \sup_{t \in \mathbb{R}^{d_t}} \lambda_{\max}(E^\top \nabla^2 f_i(Et)E)$. Note that the dimension of t can be computed as $d_t = d - \text{rank}(B)$. In the latter variant the maximum is taken over a smaller set of points, and multiplication by E is likely to further reduce the smoothness constant (and increase strong convexity constant).

Since $H(\mathbf{t})$ is L_t-smooth and μ_t-strongly convex, we have that $F^*_{[\mathbf{Bx}=0]}(\mathbf{z}) = H^*(\mathbf{z})$ is $\frac{1}{\mu_t}$-smooth and $\frac{1}{L_t}$-strongly convex [6].

Thus, the fast gradient method [15] applied to the dual problem requires

$$O\left(\sqrt{\frac{L_t}{\mu_t}}\chi(W) \log\frac{1}{\varepsilon}\right),$$

dual-oracle calls and communication rounds to ensure $F(\mathbf{x}^N) - F(\mathbf{x}) \leq \varepsilon$ and $\|\mathbf{Ax}^N\| \leq \varepsilon$ (see Section 5 for details). And using Chebyshev acceleration as described in Section 4 we can reduce the complexities to

$$N = O\left(\sqrt{\frac{L_t}{\mu_t}} \log\frac{1}{\varepsilon}\right) \text{ oracle calls at each node,}$$
$$O\left(N\sqrt{\chi(W)}\right) \text{ communications.}$$

7 Numerical Experiments

In the simulation we consider the following smooth, strongly convex objective function:

$$f_i(x) = \frac{1}{2}\|C_i x - d_i\|_2^2 + \frac{\theta}{2}\|x\|_2^2,$$

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{Cx} - \mathbf{d}\|_2^2 + \frac{\theta}{2}\|\mathbf{x}\|_2^2,$$

$$\mathbf{C} = \mathrm{diag}(C_1, \ldots, C_m), \ \mathbf{d} = \mathrm{col}\,(d_1, \ldots, d_m).$$
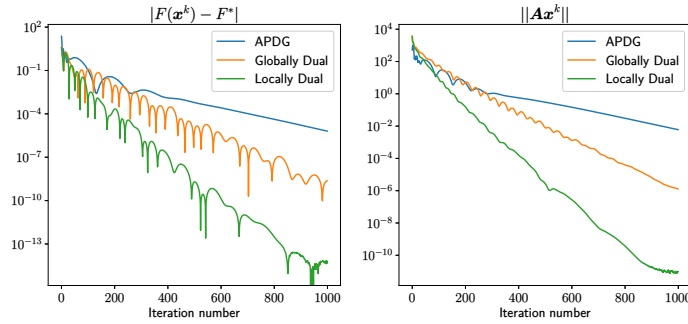
We consider different parameters of the problem such as the dimension of x, the rank of $B \in \mathbb{R}^{\dim(x) \times \dim(x)}$ and the number of nodes. For each case we plot function error and constraints violation norm at each iteration for all our algorithms: APDG, Locally and Globally Dual approaches. The Chebyshev acceleration is not applied in the experiments, so each iteration corresponds to one gradient computation (gradient of primal function in case of APDG, and gradient of dual function is case of dual approaches). We also provide tables with comparison of time and number of iterations required to achieve given accuracy. Time is measured with our Python/NumPy [5] implementation of the algorithms, which is available on GitHub[5].

1. For the first case we consider the ring network with $m = 5$ nodes, $x \in \mathbb{R}^{40}$ and rank $B = 1$. Typical convergence plot is shown on Fig. 1. One can see that all algorithms converge linearly, with the fastest one in terms of iterations number being Locally Dual, and the slowest one being APDG. However, computing the gradient of a dual function might be an arithmetically more expensive operation than computing primal gradient in the black-box scenario. In our implementation we compute the gradient of dual function by numerically solving the system of linear equations with its right-hand part being changed between iterations. It means that one iteration of the Dual methods is more time-consuming than one iteration of APDG. In the Table 1, we compare computational time and number of iterations required to achieve given accuracy. The results are averaged for 100 randomly generated problems.
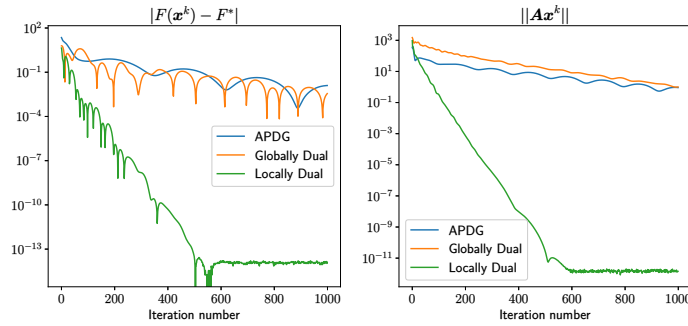
Table 1: Time and iterations for achieving $\|\mathbf{Ax}^k\| < 10^{-2}$. Averaged over 100 experiments. Problem parameters: 5 nodes, $\dim(x) = 40$, rank $B = 1$.

	APDG	Globally Dual	Locally Dual
Iterations	875.3	502.7	276.7
Time (s)	0.193	0.510	0.233

---

[5] Source code: https://github.com/niquepolice/decentr_constr_dual

Fig. 1: 5 nodes, $\dim(x) = 40$, rank $B = 1$.

2. Next we use the same number of nodes and the dimension of x, but increase the rank of B. Even for rank $B = 3$ the condition number of the locally dual problem usually is about two orders of magnitude smaller than the condition number of the globally dual problem, therefore the globally dual approach has a significant advantage in that case. Typical convergence plots are shown in Figure 2, averaged iteration and time complexities for satisfying stopping criteria are shown in Table 2.

Fig. 2: 5 nodes, $\dim(x) = 40$, rank $B = 3$.

3. In the case of higher dimension (10 nodes, $\dim(x) = 100$, rank $B = 1$) we used Erdős-Rényi random communication graphs with edge probability $= 0.3$. APDG seems to converge much faster by constraints violation norm at first iterations then other methods (Fig. 3), and its convergence rate is close to other methods. See also Table 3 for averaged results of multiple experiments.

Table 2: Time and iterations for achieving $\|\mathbf{A}\mathbf{x}^k\| < 10^{-1}$. Averaged over 100 experiments. Problem parameters: 5 nodes, $\dim(x) = 40$, rank $B = 3$.

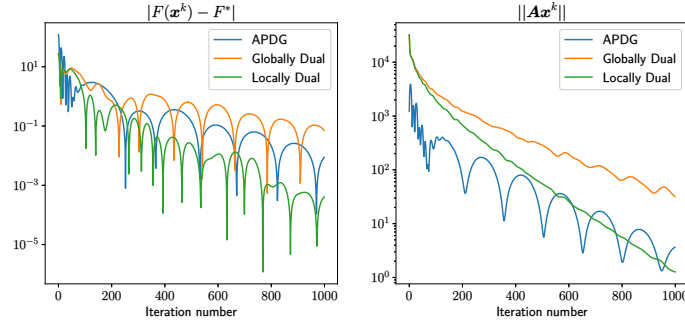|	APDG	Globally Dual	Locally Dual
Iterations	1555.5	1551.7	123.1
Time (s)	0.337	1.577	0.127

Fig. 3: Erdős-Rényi graph on 10 nodes, average degree = 3.6. $\dim(x) = 100$, rank $B = 1$.

Table 3: Time and iterations for achieving accuracy $\|\mathbf{A}\mathbf{x}^k\| < 10^1$. Averaged over 10 experiments. Problem parameters: 10 nodes, edge probability = 0.3, $\dim(x) = 100$, rank $B = 1$.

	APDG	Globally Dual	Locally Dual
Iterations	404.3	2227.9	1425.5
Time (s)	2.561	54.024	16.544

References

1. T. Erseghe. Distributed optimal power flow using admm. *IEEE Transactions on Power Systems*, 29(5):2370–2380, Sep. 2014.
2. A. Falsone, K. Margellos, S. Garatti, and M. Prandini. Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, 84:149–158, 2017.
3. A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini. Tracking-admm for distributed constraint-coupled optimization. *Automatica*, 117:1–13, 202.
4. E. Gorbunov, D. Dvinskikh, and A. Gasnikov. Optimal decentralized distributed algorithms for stochastic convex optimization. *arXiv:1911.07363*, 2019.
5. C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
6. S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. On the duality of strong convexity and strong smoothness: learning applications and matrix regularization. Technical report, Toyota Technological Institute, 2009.
7. D. Kovalev, A. Gasnikov, and P. Richtárik. Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling. *arXiv preprint arXiv:2112.15199*, 2021.
8. D. Kovalev, A. Salim, and P. Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33:18342–18352, 2020.
9. S. Liang, L. Y. Wang, and G. Yin. Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, 65:347–353, Jan 2020.
10. S. Liang, X. Zheng, and Y.Hong. Distributed ninsmooth optimization with coupled inequality constraints via modified lagrangian function. *IEEE Transaction on Automatic Control*, 63:1753–1759, 2018.
11. D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, Nov 2017.
12. I. Necoara and V. Nedelcu. Distributed dual gradient methods and error bound conditions. *arXiv:1401.4398*, 2014.
13. I. Necoara and V. Nedelcu. On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica*, 55:209–216, 2015.
14. I. Necoara, V. Nedelcu, and I. Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *Journal of Process Control*, 21(5):756–766, 2011. Special Issue on Hierarchical and Distributed Model Predictive Control.
15. Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
16. Y. Nesterov. *Introductory Lectures on Convex Optimization: a basic course.* Kluwer Academic Publishers, Massachusetts, 2004.

17. N. Patari, V. Venkataramanan, A. Srivastava, D. K. Molzahn, N. Li, and A. Annaswamy. Distributed optimization in distribution systems: Use cases, limitations, and research needs. *IEEE Transactions on Power Systems*, pages 1–1, 2021.

18. V. Rostampour, O. t. Haar, and T. Keviczky. Distributed stochastic reserve scheduling in ac power systems with uncertain generation. *IEEE Transactions on Power Systems*, 34(2):1005–1020, 2019.

19. K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3027–3036, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

20. J. Wang and G. Hu. Distributed optimization with coupling constraints in multi-cluster networks based on dual proximal gradient method. *arXiv preprint arXiv:2203.00956*, 2022.

21. Z. Wnag and C. J. Ong. Distributed model predictive control of linear descrete-times systems with local and global cosntraints. *Automatica*, 81:184–195, 2017.

22. D. Yarmoshik, A. Rogozin, O. Khamisov, P. Dvurechensky, A. Gasnikov, et al. Decentralized convex optimization under affine constraints for power systems control. *arXiv preprint arXiv:2203.16686*, 2022.

23. D. Yuan, D. W. C. Ho, and G. P. Jiang. An adaptive primal-dual subgradient algorithm for online distributed constrained optimization. *IEEE Transactions on Cybernetics*, 48:3045–3055, 2018.