2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Automated Worst-Case Performance Analysis of Decentralized Gradient Descent

Sebastien Colla and Julien M. Hendrickx

***Abstract*— We develop a methodology to automatically compute worst-case performance bounds for a class of decentralized algorithms that optimize the average of local functions distributed across a network. We extend the recently proposed PEP approach to decentralized optimization. This approach allows computing the exact worst-case performance and worst-case instance of centralized algorithms by solving an SDP. We obtain an exact formulation when the network matrix is given, and a relaxation when considering entire classes of network matrices characterized by their spectral range. We apply our methodology to the decentralized (sub)gradient method, obtain a nearly tight worst-case performance bound that significantly improves over the literature, and gain insights into the worst communication networks for a given spectral range.**

I. Introduction

The goal of this paper is to develop a methodology that automatically provides nearly tight performance bounds for primal first-order decentralized methods on convex functions.

Decentralized optimization has received an increasing attention due to its useful applications in large-scale machine learning and sensor networks, see for example references in this survey [1]. In decentralized methods for separable objective functions, we consider a set of agents $\{1,\dots,N\}$, working together to optimize this global objective:

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x)=\frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{1}$$

where f_i is the private function locally held by agent i. To achieve this goal, each agent i of the system holds its own version x_i of the decision variable $x\in\mathbb{R}^d$. Agents perform local computations and exchange local information with their neighbors to seek to reach an agreement on the minimizer x^* of the global function f. Exchanges of information often take the form of a multiplication by a given matrix $W\in\mathbb{R}^{N\times N}$, typically assumed symmetric and doubly stochastic.

A classical example of decentralized optimization method is the decentralized (sub)gradient descent (DGD) [2] where agents successively perform the consensus step (2) and the local gradient step (3). We have, for all $i\in\{1,\dots,N\}$,

$$y_i^k=\sum_{j=1}^{N} w_{ij}x_j^k, \tag{2}$$

$$x_i^{k+1}=y_i^k-\alpha\nabla f_i(x_i^k), \tag{3}$$

S. Colla and J. M. Hendrickx are with the ICTEAM institute, UCLouvain, Louvain-la-Neuve, Belgium. S. Colla is supported by the French Community of Belgium through a FRIA fellowship (F.R.S.-FNRS). J. M. Hendrickx is supported by the "RevealFlight" Concerted Research Action (ARC) of the Federation Wallonie-Bruxelles and by the Incentive Grant for Scientific Research (MIS) "Learning from Pairwise Comparisons" of the F.R.S.-FNRS. Email addresses: sebastien.colla@uclouvain.be, julien.hendrickx@uclouvain.be

where $\alpha>0$ is a constant step-size. Although the tools we develop in this paper are general, this simple algorithm is used as a case study in Section IV. This focus has been chosen for the simplicity of the algorithm and not for its performance. Indeed there exists many other decentralized algorithms that perform better, including EXTRA [3], DIGing [4], NIDS [5]. The analysis of these other algorithms will be the focus of future work.

The quality of an optimization method is often evaluated via a worst-case guarantee. Obtaining theoretical worst-case performance bounds for decentralized algorithms can often be a challenging task, requiring combining the impact of the optimization component and the interconnection network. This can result in performance bounds that are not very tight. For example, we will show in Section IV that the available performance bounds of DGD are significantly worse than the actual worst-cases.

In this work, we follow an alternative computational approach that finds a worst-case performance guarantee of an algorithm by solving an optimization problem. This is known as the performance estimation problem - PEP - and has been studied for centralized fixed-step first-order methods, see e.g. [6], [7]. PEP has never been applied to decentralized optimization methods. Particularly, the current PEP framework does not allow to represent matrix multiplications in the methods it analyzes, except if the matrix is explicitly given. This paper proposes possible solutions for this missing piece to the analysis of decentralized methods via PEP. Our contributions are the following:

We provide two formulations of the multiplication by a symmetric *generalized doubly stochastic* matrix W, which is defined as a doubly stochastic matrix but without the restriction of being non-negative. Those formulations allow to write and solve PEP for a large class of decentralized methods. The first formulation is specific to a communication matrix W given a priori, is exact, and is directly derived from the current PEP framework. It can be applied to any matrix W and leads to tight performance bounds that are specific to this given matrix. The second formulation is a relaxation that considers entire classes of possible matrices, based on their spectrum. It is our main methodological contribution. We demonstrate the usefulness of these new formulations of PEP by analyzing the worst-case performance of DGD. For DGD, the second relaxed formulation leads to tight spectral performance bounds significantly improving on the existing theoretical ones. Our numerical experiments show that our bounds are independent of the number of agents N and can be used to easily choose the optimal step-size of the method.

978-1-6654-3658-8/21/$31.00 ©2021 IEEE

II. General PEP approach

In principle, a tight performance bound on an algorithm could be obtained by running it on every single instance - function and initial condition - allowed by the setting considered and selecting the worst performance obtained. This would also directly provide an example of "worst" instance if it exists. The performance estimation problem (PEP) formulates this abstract idea as a real optimization problem that maximizes the error measure of the algorithm result, over all possible functions and initial conditions allowed. This optimization problem is inherently infinite-dimensional, as it contains a continuous function among its variables. Nevertheless, Taylor et al. have shown [6], [7] that PEP can be solved exactly for a wide class of centralized first-order algorithms on convex functions, using an SDP formulation.

We illustrate this approach with a simple example on K steps of the centralized unconstrained subgradient descent. Let Perf denotes any performance measure for which we would like to find the worst-case, e.g. $f(x^K) - f(x^*)$. Perf can depend on the function f, its (sub)gradients, the minimizer x^* and any of the iterates x^k. PEP for this algorithm can be written as follows:

$$\begin{aligned} \sup_{f,x^0,\dots,x^K,x^*} \quad & \mathrm{Perf}\big(f, x^0, \dots, x^K, x^*\big) \qquad \text{(Subgradient-PEP)} \\ \text{s.t.} \quad & f \in \mathcal{F} \\ & x^* = \operatorname*{argmin}_x f(x), \\ & x^{k+1} = x^k - \alpha \nabla f(x^k), \quad \text{for } k = 0, \dots, K-1, \\ & x^0 \text{ satisfies some initial conditions.} \end{aligned}$$

where $\mathcal{F}$ denotes a class of functions and $\nabla f(x^k)$ denotes any subgradient of f at x^k.

To overcome the infinite dimension of variable f, we notice that the problem (Subgradient-PEP) only uses the values and subgradients of the function at specific points: the iterates $x^0, \dots, x^K$ and the minimizer x^*. This motivates the discretization of the problem: the decision variables are restricted to the iterations, subgradients, and function values associated with these specific points $\{(x^k, g^k, f^k)\}_{k \in I}$ and we add the constraint that there is a function of the class $\mathcal{F}$ which interpolates those data points $\{(x^k, g^k, f^k)\}_{k \in I}$. This can be done using necessary and sufficient interpolation constraints for the functional class under consideration. Such constraints are provided for many different classes of functions in [7, Section 3]. For example, for the class of convex functions with bounded subgradients $\mathcal{F}_B$, we can use interpolation constraints from the following particularization of [7, Theorem 3.5], initially formulated for smooth convex functions with bounded gradient.

Theorem 1 ([7, Theorem 3.5]): Let I be an index set. There exists a function $f \in \mathcal{F}_B$ such that $f^k = f(x^k)$ and $g^k = \nabla f(x^k)$ $(k \in I)$ if and only if for all pair $k \neq l \in I$

$$f^k \geq f^l + \langle g^l, x^k - x^l \rangle, \quad \text{and} \quad ||g^k||^2 \quad \leq B^2.$$

As it is the case for $\mathcal{F}_B$, the interpolation constraints are generally quadratics, potentially non-convex, involving scalar products and functional values. They can be reformulated using a Gram matrix G and a vector containing function values $f = [f_i]_{i \in I}$. The Gram matrix is a symmetric positive semidefinite matrix containing scalar products between iterates $x^k \in \mathbb{R}^d$ and subgradients $g^k \in \mathbb{R}^d$ for $k \in I$.

$$G = P^T P, \text{ with } P = \begin{bmatrix} g^0 \dots g^K g^* x^0 \dots x^K x^* \end{bmatrix}.$$

Quadratics interpolations constraints from Theorem 1 are linear in G and f. Linear equality constraints, such as the constraints for the iterates of the algorithm, can also be expressed linearly with G, by isolating all elements on the same side and squaring the equation. This leads to an equivalent positive semidefinite program (SDP) for PEP with the Gram matrix G and the vector of functional values f as variables. Besides the reformulation of interpolation, optimality, iterates, and initial constraints, we should also impose that $G \succeq 0$ and rank $G \leq d$. By relaxing this rank constraint, the formulation becomes independent of the dimension d of the iterates and provides the worst case in any dimension. This SDP formulation is convenient because it can be solved numerically to global optimality. See [7] for details about the SDP formulation of PEP, including ways of reducing the size of matrix G. However, the dimension of G always depends on the number of iterations K.

From a solution G, f of the SDP formulation, we can construct, using Cholesky decomposition for example, a solution for the discretized variables $\{(x^k, g^k, f^k)\}_{k \in I}$. Since these points satisfy sufficient interpolation constraints, we can also construct a function from $\mathcal{F}$ interpolating these points.

The following proposition states a sufficient condition under which a PEP can be formulated as an SDP, which can then be solved exactly.

Definition 1 (Gram-representable): Consider a Gram matrix G and a vector f, as defined previously. We say that a constraint or an objective is linearly (resp. LMI) Gram representable if it can be expressed using a finite set of linear (resp. LMI) constraints involving (part of) G and f.

Proposition 2 ([7, Proposition 2.6]): If the interpolation constraints of the class of functions $\mathcal{F}$, the satisfaction of the method $\mathcal{M}$, the performance measure Perf and the set of constraints $\mathcal{I}$, which includes the initial conditions, are linearly (or LMI) Gram representable, then, computing the worst-case for criterion Perf of method $\mathcal{M}$ after K iterations on objective functions in class $\mathcal{F}$ with constraints $\mathcal{I}$ can be formulated as an SDP, with G and f as variables.
This remains valid when the objective function is the sum of N sub-functions.

Remark: Proposition 2.6 in [7] was only formulated for linearly Gram-representable constraints, but its extension to LMI Gram-representable constraints is direct. Such constraints appear in the analysis of classes of network matrices.

PEP techniques allowed answering several important questions in optimization, see e.g. the list in [8], and to make important progress in the tuning of certain algorithms including the well-known centralized gradient-descent. Following a numerical exploration in [9], it was further exploited by Kim and Fessler to design the Optimized Gradient Method (OGM), the fastest possible first-order optimization method

for smooth convex functions [10]. It can also be used to deduce proofs about the performance of the algorithms [11]. It has been made widely accessible via a Matlab toolbox [12]. However, PEPs have never been used to study decentralized methods. The main challenge is that there is no representation of the interconnections between the agents that can be embedded in the formulation.

We also note an alternative approach with similar motivations for automated performance evaluation that is proposed in [13], and is inspired by dynamical systems concepts. Integral quadratic constraints (IQC), usually used to obtain stability guarantees on complex dynamical systems, are adapted in order to obtain sufficient conditions for the convergence of optimization algorithms. It provides iteration-independent linear rates of convergence, based on relatively small size problems, but it does not apply when the convergence is not geometric. Unlike PEP, it offers no a priori guarantee of tightness, though it turns out to be tight in certain situations. An application of the IQC methodology to decentralized optimization is presented in [14] and has already been used for designing a new algorithm that achieves a faster worst-case convergence rate. But this methodology cannot be directly applied to DGD, nor to smooth convex functions or any other situation that does not have a geometric convergence. Also, this IQC approach focuses on one iteration of the algorithm to derive the worst-case convergence rate, and hence, it cannot exploit situations where the communication matrix is identical at each iteration to improve it.

III. Representations of consensus steps for PEP

In this section, we present a way of representing the interactions between agents and thereby providing the missing block to PEP formulation for first-order decentralized optimization. We focus on the situation where the interactions take place via a weighted averaging and can thus be described as a consensus step of the following form:

$$y_i = \sum_{j=1}^{N} w_{ij} x_j, \qquad \text{for all } i \in \{1, \dots, N\}, \tag{4}$$

where x_j can represent any vector in $\mathbb{R}^d$ held by agent j, e.g. its local iterates in the case of DGD but it could be more evolved. Vector $y_i \in \mathbb{R}^d$ is an auxiliary variable that represents the result of the interaction and $W \in \mathbb{R}^{N\times N}$ is the weighted communication matrix. While we focus on DGD in section IV, this form of communication is used in many other decentralized methods such as EXTRA [3], DIGing [4], NIDS [5] and the results presented in this section can be exploited for all these methods too.

We suppose the communication matrix W symmetric, i.e. $w_{ij} = w_{ji}$ and *generalized doubly stochastic* [15], [16], i.e. $\sum_{i=1}^{N} w_{ij} = 1$, $\sum_{j=1}^{N} w_{ij} = 1$, for all i, j. This last assumption is a relaxation of the more usual double stochasticity since it does not require elements of W to be non-negative. However, many results from the literature on decentralized optimization are based on spectral information and do not use the non-negativity of W either, see e.g. [1], [3], [5]. We analyze the impact of this relaxation in the case of DGD in Section IV.

A. Communication matrix given a priori

When the communication matrix is given *a priori*, the consensus step (4) is a simple linear equality constraint that is linearly Gram-representable. In that case, the constraint can be used in the SDP formulation of a PEP, see Proposition 2. This allows writing PEPs that provide exact worst-case performances for the given decentralized method and the specific communication matrix given *a priori*. It can be applied to any matrix W, and not only for generalized doubly stochastic ones. This can be useful for trying different communication matrices and observing their impact on the worst-case performance of the algorithm. It will serve as an exact comparison baseline in the numerical experiments of Section IV. The next section presents a way of representing communications in PEP that allows obtaining more general performance guarantees, valid over entire classes of communication matrices and not only for a specific one.

B. Communication matrix as a variable

We now consider that the matrix W is not given *a priori*, but is *one of the decision variables* of the performance estimation problem with bounds on its possible eigenvalues. Hence the search space contains the matrix W in addition to the usual variables, and is restricted by the following constraints, for each agent $i \in \{1, \dots, N\}$ and each consensus steps $k \in \{1, \dots, K\}$,

$$W \in \mathcal{W}(\lambda^-, \lambda^+), \tag{5}$$

$$y_i^k = \sum_{j=1}^{N} w_{ij} x_j^k, \tag{6}$$

where $\mathcal{W}(\lambda^-, \lambda^+)$ is the set of real, symmetric and generalized doubly stochastic matrices that have their eigenvalues between λ^- and λ^+, except for $\lambda_1 = 1$:

$$\lambda^- \leq \lambda_N \leq \dots \leq \lambda_2 \leq \lambda^+ \quad \text{where } \lambda^-, \lambda^+ \in [-1, 1] \ .$$

The set of the different consensus steps represented by indices $k = 1, \dots, K$ can for example correspond to the set of the different iterations of the algorithm, but it could also be a subset of the iterations if the communication matrix changes, or other subsets of the consensus steps if different consensus (6) are applied to the same iteration.

We do not have a direct way for representing constraints (5) and (6) in an LMI Gram-representable manner, but we construct a relaxation that we will see in section IV is often close to tight. From constraints (5) and (6), we derive new necessary conditions involving only variables y_i^k and x_i^k allowing to eliminate W from the problem.

We first restrict ourselves to the simpler case when the local variables are one-dimensional: $x_i^k, y_i^k \in \mathbb{R}^d$ with $d = 1$.

Let X and Y be the following matrices from $\mathbb{R}^{N\times K}$:

$$X_{ik} = x_i^k \quad \text{and} \quad Y_{ik} = y_i^k \qquad \text{for} \quad \begin{matrix} i=1,\dots,N \\ k=1,\dots,K \end{matrix}$$

Each column corresponds thus to a different consensus step k, and each row to a different agent i. Using this notation, the consensus steps constraints (6) can simply be written as $Y = WX$. We decompose matrices X and Y in average and centered parts:

$$X = \mathbf{1}\overline{X}^T + X_\perp, \qquad Y = \mathbf{1}\overline{Y}^T + Y_\perp,$$

where $\overline{X}$ and $\overline{Y}$ are agents average vectors in $\mathbb{R}^K$, defined as $\overline{X}_k = \frac{1}{N}\sum_{i=1}^N x_i^k$, $\overline{Y}_k = \frac{1}{N}\sum_{i=1}^N y_i^k$ for $k = 1, \dots, K$ and $\mathbf{1} = [1 \dots 1]^T$. Using these notations, we state the new necessary conditions in the following theorem.

Theorem 3 (Consensus Constraints): If $Y = WX$ for a matrix $W \in \mathcal{W}(\lambda^-, \lambda^+)$, with $X, Y \in \mathbb{R}^{N\times K}$, then

(i) The matrices X^TY and $X_\perp^T Y_\perp$ are symmetric,

(ii) The following constraints are satisfied

$$\overline{X} = \overline{Y}, \tag{7}$$

$$\lambda^- X_\perp^T X_\perp \preceq X_\perp^T Y_\perp \preceq \lambda^+ X_\perp^T X_\perp, \tag{8}$$

$$(Y_\perp - \lambda^- X_\perp)^T (Y_\perp - \lambda^+ X_\perp) \preceq 0, \tag{9}$$

where the notations $\succeq$ and $\preceq$ denote respectively positive and negative semi-definiteness.

(iii) Constraints (7), (8), (9) are LMI Gram-representable.

Proof: First, we average elements from both sides of the assumption $Y = WX$ to obtain constraint (7):

$$\overline{Y} = \frac{\mathbf{1}^T Y}{N} = \frac{\mathbf{1}^T W X}{N} = \frac{\mathbf{1}^T X}{N} = \overline{X}.$$

where $\mathbf{1}^T W = \mathbf{1}^T$ follows from W being generalized doubly stochastic, i.e. its rows and columns sum to one.

The symmetry of the matrix X^TY is directly obtained using the assumption $Y = WX$, with W being symmetric. We can use the same argument for the symmetry of $X_\perp^T Y_\perp$, because having $Y = WX$ and $\overline{X} = \overline{Y}$ implies that $Y_\perp = WX_\perp$.

Since the communication matrix W is real and symmetric, we can take an orthonormal basis $\mathbf{v_1}, \dots, \mathbf{v_N}$ of eigenvectors, corresponding to real eigenvalues $\lambda_N \leq \dots \leq \lambda_2 \leq \lambda_1$. The largest eigenvalue is $\lambda_1 = 1$ and corresponds to the eigenvector $\mathbf{v_1} = \mathbf{1}$. Indeed W is doubly-stochastic and have all its other eigenvalues below 1 by assumption:

$$\lambda^- \leq \lambda_i \leq \lambda^+ \text{ for } i = 2, \dots, N, \text{ with } \lambda^-, \lambda^+ \in [-1, 1].$$

Let us now consider a combination $X_\perp z$ of the columns of the matrix $X_\perp$, for an arbitrary $z \in \mathbb{R}^K$. It can be decomposed in the eigenvectors basis of W, and used to express the combination $Y_\perp z$:

$$X_\perp z = 0\mathbf{v_1} + \sum_{i=2}^N \gamma_i \mathbf{v}_i, \text{ and } Y_\perp z = WX_\perp z = \sum_{i=2}^N \gamma_i \lambda_i \mathbf{v}_i, \tag{10}$$

where γ_i are real coefficients. The coefficient for $\mathbf{v_1} = \mathbf{1}$ is zero because $X_\perp z$ is orthogonal to this eigenvector since it is centered: $\mathbf{1}^T(X_\perp z) = 0$. Using this decomposition to compute the scalar product $z^T X_\perp Y_\perp z$ for any $z \in \mathbb{R}^K$ leads to the following scalar inequalities

$$\begin{aligned} z^T X_\perp^T Y_\perp z = \sum_{i=2}^N \gamma_i^2 \lambda_i &\geq \lambda^- z^T X_\perp^T X_\perp z, \\ &\leq \lambda^+ z^T X_\perp^T X_\perp z. \end{aligned}$$

Having these inequalities satisfied for all $z \in \mathbb{R}^K$, is equivalent to (8). In the same way, (9) is obtained by verifying that the following inequality hold for all $z \in \mathbb{R}^K$:

$$(Y_\perp z - \lambda^- X_\perp z)^T (Y_\perp z - \lambda^+ X_\perp z) \leq 0.$$

This can be done by substituting $X_\perp z$ and $Y_\perp z$ using equation (10), and by using the bounds on λ_i ($i = 2, \dots, N$).

Finally, constraints (7), (8) and (9) can all be formulated as LMIs involving submatrices of the Gram matrix G of scalar products. Therefore, they are LMI Gram-representable. ■

Using Theorem 3, we can relax constraints (5) and (6) and replace them by (7), (8) and (9), which are LMI Gram-representable. Then, Proposition 2 allows to write a relaxed SDP formulation of a PEP providing worst-case results valid for the entire spectral class of matrices $\mathcal{W}(\lambda^-, \lambda^+)$.

Constraint (7) is related to the stochasticity of the communication matrix and imposes that variable x has the same agents average as y, for each consensus step. Linear matrix inequality constraints (8) and (9) imply in particular equivalent scalar constraints for the diagonal elements. They corresponds to independent constraints for each consensus step, i.e. for each column $x_\perp$ and $y_\perp$ of matrices $X_\perp$, $Y_\perp$:

$$\lambda^- x_\perp^T x_\perp \leq x_\perp^T y_\perp \leq \lambda^+ x_\perp^T x_\perp,$$

$$(y_\perp - \lambda^- x_\perp)^T (y_\perp - \lambda^+ x_\perp) \leq 0.$$

These constraints imply in particular that

$$y_\perp^T y_\perp \leq \beta^2 x_\perp^T x_\perp, \quad \text{where } \beta = \max(|\lambda^-|, |\lambda^+|),$$

meaning that the disagreement between the agents, measured by $y_\perp^T y_\perp$ for y and $x_\perp^T x_\perp$ for x, is reduced by a factor β^2 after a consensus. But constraints (8) and (9) also allow linking different consensus steps to each other, via the impact of off-diagonal terms, in order to exploit the fact that these steps use the same communication matrix. Moreover, if different communication matrices are used for different sets of consensus steps, the constraints from Theorem 3 can be applied independently for each set of consensus steps.

Theorem 3 is derived for $x_i^k, y_i^k \in \mathbb{R}$ but the worst-case solution of PEP has no guarantee to be one-dimensional. When the local variables are multi-dimensional: $x_i^k, y_i^k \in \mathbb{R}^d$ with $d \geq 1$, a natural approach would be to impose the constraints independently for each dimension. This might introduce some conservatism in our relaxed formulation, as it would allow each dimension to use a different matrix. But independently of this issue, this approach cannot be directly implemented in PEP for constraints (8) and (9) because we cannot access the different dimensions of a variable in the SDP formulation which only allows using scalar products. One solution is to sum these constraints over all dimensions. This may lead to the constraints not being exactly met for a specific dimension and may thus also introduce conservatism in our relaxed formulation. The following corollary presents the resulting constraints after the sum on the dimensions. The matrices $X^j, Y^j \in \mathbb{R}^{N\times K}$ contains the element from dimension j of each x_i^k, y_i^k variables and the matrices $X, Y \in \mathbb{R}^{Nd\times K}$ stack each X^j and Y^j vertically.

Corollary 4 (Theorem 3 for $d \geq 1$): If $Y^j = WX^j$, for every $j = 1, \dots, d$ and for a same matrix $W \in \mathcal{W}(\lambda^-, \lambda^+)$, i.e. $Y = (I_d \otimes W)X$, where $\otimes$ is the Kronecker product, then

(i) The matrices X^TY and $X_\perp^T Y_\perp$ are symmetric,

(ii) Constraints (7), (8) and (9) are satisfied.

(iii) Constraints (7), (8), (9) are LMI Gram-representable.

The proof of this corollary will be provided in the journal version of this article.

IV. Decentralized (sub)gradient descent (DGD): A case study

Using results from previous sections, we can build two PEP formulations for analyzing the worst-case performance of the well-known decentralized gradient descent. We consider K iterations of DGD described by (2) and (3), with constant step-size, in order to solve problem (1), i.e. minimizing $f = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$, with x^* as minimizer of f. There are different studies about DGD; e.g. [17] shows that its iterates converge to a neighborhood of the optimal solution x^* when the step-size is constant. In the following analysis, we consider the results of a recent survey [1] providing a theoretical bound for the functional error at the average of all the iterates. This error tends to zero since it considers the average of all the iterates and not only the last one.

Theorem 5 (Performance of DGD [1, Theorem 8]): Let $f_i, \ldots, f_N$ be convex local functions with subgradients bounded by B. Let x^0 be an identical starting point of each agent such that $||x^0 - x^*||^2 \leq R^2$. And let W be a symmetric and doubly stochastic matrix with eigenvalues $\lambda_2, \ldots, \lambda_N \in [-\lambda, \lambda]$, for some $\lambda \in [0, 1)$.
If we run DGD for K steps with a constant step-size $\alpha = \frac{1}{\sqrt{K}}$, then there holds[1]

$$f(x_{\text{av}}) - f(x^*) \leq \frac{R^2 + B^2}{2\sqrt{K}} + \frac{2B^2}{\sqrt{K}(1-\lambda)}, \qquad (11)$$

where $x_{\text{av}} = \frac{1}{N(K+1)}\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k$ is the average over all the iterations and all the agents.

For comparison purposes, our PEP formulations of DGD use the same assumptions as Theorem 5. Our first formulation characterizes the same performance measure as Theorem 5: $f(x_{\text{av}}) - f(x^*)$, with appropriate constraints for the initialization: $||x^0 - x^*||^2 \leq R^2$, the set of functions (Theorem 1) and the algorithm iterates that are generated by DGD with a given matrix W (equations (2) and (3)). All these constraints are linearly Gram-representable and can then be used in the SDP formulation of PEP, according to Proposition 2. This formulation is referred to as the *exact formulation* because it finds the exact worst-case performance of the algorithm in the specific case where it is used with the given matrix W. The second formulation relaxes the consensus constraints imposed by equation (2) and replaces them with the constraints from Theorem 3, with $-\lambda^- = \lambda^+ = \lambda$. Those are LMI Gram-representable (see Corollary 4) and can then be used in the SDP formulation of PEP, according to Proposition 2. This formulation is referred to as the *spectral formulation* and provides *spectral worst-cases*, i.e. upper bounds on the exact worst-case performances of the algorithm, valid for any matrix $W \in \mathcal{W}(-\lambda, \lambda)$, i.e. for any symmetric generalized doubly stochastic matrix with a given range of eigenvalues. In particular, these spectral upper bounds also hold for non-negative matrices from $\mathcal{W}(-\lambda, \lambda)$ and can therefore be compared with the bound from Theorem 5.

[1]Note the factor 2 in the second term of the bound (11) was missing in [1] but its presence was confirmed by the authors of [1].

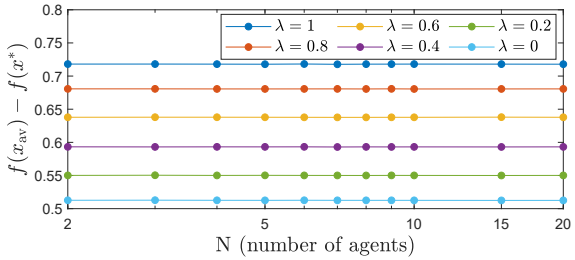

Fig. 1: Independence of N for the spectral worst-case performance of $K = 5$ iterations of DGD in the setting of Theorem 5.

In our experiments, we focus on the situation where $R = 1$ and $B = 1$, but the results obtained can be scaled up to general values of R and B using appropriate changes of variables, as explained in the appendix.

Impact of the number of agents N: In Fig. 1, we observe that the results of the spectral formulation are *independent* of the number of agents $N \geq 2$ there are in the problem. This is shown for K = 5 iterations and for different spectral ranges. This observation has been confirmed for other values of K (10, 15, and 20). Moreover, the theoretical performance bound from Theorem 5 is also independent of N. Therefore, in the sequel, we analyze the spectral formulation for $N = 3$, which is the smallest value of N that still allows non-trivial communication networks to be considered.

Comparison with Theorem 5: We compare the spectral bound with the theoretical bound from Theorem 5 for different values of λ. Fig. 2 shows the evolution of both bounds with λ for $K = 10$ iterations of DGD with $N = 3$ agents. We observe that the spectral worst-case performance bound (in blue) largely improves on the theoretical one (in red), especially when λ approaches 1, in which case the theoretical bound grows unbounded. Moreover, having large values for λ is frequent for communication matrices of large networks of agents [18]. Hence, the improvements of the bounds when λ is close to 1 are even more valuable. For example, for a 5 by 5 grid of agents with Metropolis weights [1], the resulting communication matrix has eigenvalues $\lambda_2, \ldots, \lambda_N \in [-0.92, 0.92]$. In that case, after $K = 10$ iterations, our spectral bound guarantees that the performance measure is below 0.85, compared to 8.3 for the theoretical bound from Theorem 5. This accuracy of 0.83 would only be guaranteed using Theorem 5 with $K = 950$.

Worst communication matrix and tightness analysis: If the solution variables X and Y from the spectral PEP formulation can be linked by a unique matrix, then it can be found using the pseudo-inverse $\hat{W} = YX^+$. In practice, the matrix may not exist or may not be unique but $\hat{W}$ still provides an approximation of the worst-case matrix. When considering the spectral formulation with a symmetric spectral range $-\lambda^- = \lambda^+ = \lambda$ and K large enough, we observe that it recovers, within numerical errors, matrices of the following form

$$[W^{(1)}]_{ij} = \begin{cases} \frac{1+\lambda}{N} & \text{if } i \neq j, \\ 1 - (N-1)\frac{1+\lambda}{N} & \text{if } i = j \end{cases} \qquad (12)$$

Matrix $W^{(1)}$ is symmetric and is generalized doubly stochastic, leading 1 to be one of its eigenvalues. All its other eigenvalues are equal to $-\lambda$. The bound obtained using the exact PEP formulation with this specific matrix $W^{(1)}$ for $K = 10$ is plotted in green in Fig. 2 and *exactly* reaches the spectral bound in blue, within numerical errors. In that case, the spectral formulation, even though it is a relaxation, provides a *tight performance bound* for DGD with symmetric generalized doubly stochastic matrices. This observation has been confirmed for other values of K (5, 15, and 20).

Doubly stochastic versus generalized doubly stochastic: Since every doubly stochastic matrix is also generalized doubly stochastic, the spectral bound also provides an upper bound on the performance of DGD with symmetric doubly stochastic matrices. This bound remains tight for $\lambda \leq \frac{1}{N-1}$ because the worst-case matrix $W^{(1)}$ (12) we have obtained is non-negative and is therefore doubly stochastic. For $\lambda > \frac{1}{N-1}$, this is no longer the case and the analysis is performed by empirically looking for symmetric stochastic communication matrices leading to the worst performance. In Fig. 2, for $N = 3$ and $\lambda > 0.5$, we have generated more than 6000 random symmetric doubly stochastic 3 by 3 matrices. We have analyzed their associated DGD performance using the exact PEP formulation and have only kept those leading to the worst performances. The resulting pink curve deviates no more than 20% below the spectral bound. In that case, the spectral bound is thus no longer tight for DGD with doubly stochastic matrices but remains very relevant. This observation has been confirmed for other values of K and N ($N = 3, 5, 7$, and $K = 10, 15$).

Evolution with the total number of iterations K: Fig. 3 shows the evolution of the spectral worst-case performance for DGD multiplied by $\sqrt{K}$, for different values of λ. Except when $\lambda = 1$, all lines tend to a constant value, meaning that the spectral bound behaves in $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$, as the theoretical bound (11), but with a much smaller hidden constant. When $\lambda = 1$, the line grows linearly and never reaches a constant value. In that case, the worst communication matrices lead to counterproductive interactions, preventing DGD from working in the worst case.

Tuning the step-size α: The PEP methodology allows us to easily tune the parameters of a method. For example, Fig. 4 shows the evolution of the spectral worst-case performance of DGD with the constant step-size it uses, in the setting of Theorem 5 with $N = 3$, $K = 10$ and $\lambda = 0.8$. In that case, we observe that the value $\alpha = \frac{1}{\sqrt{K}}$ used in Theorem 5 for deriving the theoretical performance bound is not the best possible choice for α and should be divided by two to improve the performance guarantees by 30%. The optimal value for α, regarding our spectral bound, is the one that provides the best worst-case guarantee, whatever the communication matrix from $\mathcal{W}(-\lambda, \lambda)$ is used.

The impact of the step-size on the other experiments and observations can be studied by setting $\alpha = \frac{h}{\sqrt{K}}$, for some $h > 0$. We focused on $h = 1$ for comparison with the theoretical bound from Theorem 5. Nevertheless, all our other observations have been confirmed for $h = 0.1, 0.5, 2, 10$.

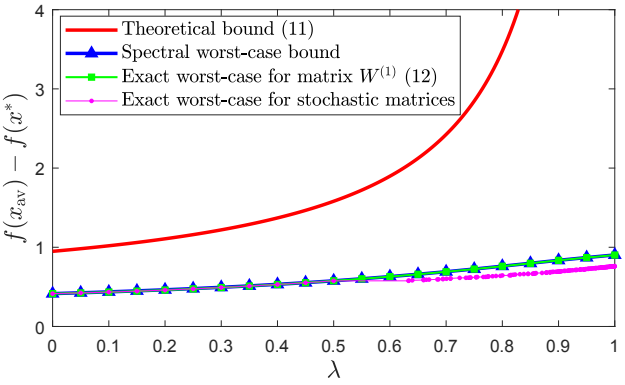

Fig. 2: Evolution with λ of the worst-case performance of $K = 10$ iterations of DGD in the setting of Theorem 5 with $N = 3$ agents. The plot shows (i) the theoretical bound from equation (11) (in red), largely above (ii) the spectral worst-case performance (in blue), (iii) the exact worst-case performance for the symmetric generalized doubly stochastic matrix $W^{(1)}$ from equation (12) (in green) and (iv) the exact worst-case performance for symmetric doubly stochastic matrices found based on an exhaustive exploration of such matrices used in the exact PEP formulation (in pink). This indicates the tightness of the spectral formulation of PEP for DGD with symmetric generalized doubly stochastic matrices, within numerical errors.

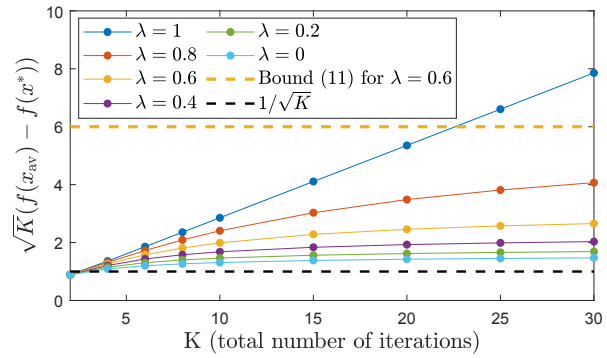

Fig. 3: Evolution with K of the *normalized* spectral worst-case performance of K iterations of DGD in the setting of Theorem 5 with $N = 3$. The shown spectral worst-cases are normalized by $\frac{1}{\sqrt{K}}$ to show that they evolve at this rate.

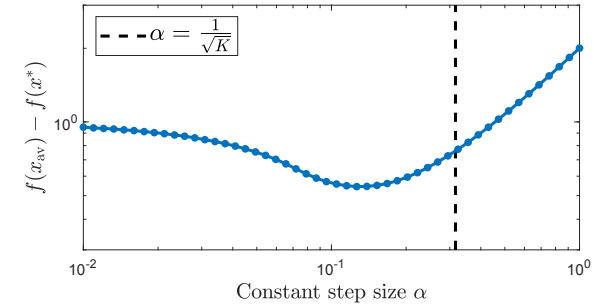

Fig. 4: Evolution with α of the spectral worst-case performance of $K = 10$ iterations of DGD in the setting of Theorem 5 with $N = 3$ agents and $\lambda = 0.8$ (except for α).

Code and Toolbox: PEP problems are solved using the PESTO Matlab toolbox [12], with Mosek solver, within 200 seconds. For example, for $N = 3$, the time needed for a regular laptop to solve the spectral formulation is about 3, 12, 48, and 192 seconds respectively for $K = 5, 10, 15, 20$. The PESTO toolbox is available on GITHUB (repository AdrienTaylor/Performance-Estimation-Toolbox). We have updated PESTO to allow easy and intuitive PEP formulation of gradient-type decentralized optimization methods, and we have added a code example for DGD.

V. CONCLUSION

We have developed two representations of consensus steps that can be embedded in the performance estimation problem (PEP) in order to automatically compute the worst-case performance of decentralized optimization methods in which such consensuses appear. Our first formulation uses a given communication matrix to directly incorporate the updates of the chosen method as constraints over the iterates. It provides the exact worst-case performance of the method for this specific matrix. The second formulation provides upper bounds on the worst-case performance that are valid for an entire spectral class of matrices. It relaxes the specific consensus constraints and adds new necessary constraints for the given spectral class of matrices. Although the second formulation is a relaxation, the performance guarantees it provides for DGD largely improve on the theoretical existing ones and are numerically tight for the class of symmetric generalized doubly stochastic communication matrices. Moreover, these resulting spectral bounds for DGD are independent of the number of agents in the problem and help for better tuning of the step-size. Further works could exploit this new spectral formulation to analyze, develop our understanding, and improve many other decentralized methods. Finally, this work could also help in the creation of new decentralized methods.

ACKNOWLEDGMENTS

The authors wish to thank Adrien Taylor for his helpful advice concerning the PESTO toolbox.

REFERENCES

[1] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.
[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, pp. 48 – 61, 02 2009.
[3] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, 04 2014.
[4] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, 07 2016.
[5] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Transactions on Signal Processing*, vol. PP, 04 2017.
[6] A. B. Taylor, J. M. Hendrickx, and F. Glineur, "Smooth strongly convex interpolation and exact worst-case performance of first-order methods," *Mathematical Programming*, vol. 161, 02 2015.
[7] ——, "Exact worst-case performance of first-order methods for composite convex optimization," *SIAM Journal on Optimization*, vol. 27, 12 2015.
[8] A. Taylor, "Convex interpolation and performance estimation of first-order methods for convex optimization," Ph.D. dissertation, UCLouvain, 2017, advisors : François Glineur and Julien M. Hendrickx.
[9] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: A novel approach," *Mathematical Programming*, vol. 145, 06 2012.
[10] D. Kim and J. Fessler, "Optimized first-order methods for smooth convex minimization," *Mathematical Programming*, vol. 159, 06 2014.
[11] A. B. Taylor, J. M. Hendrickx, and F. Glineur, "Exact worst-case convergence rates of the proximal gradient method for composite convex minimization," *Journal of Optimization Theory and Applications*, vol. 178, 08 2018.
[12] ——, "Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods," in *56th IEEE Conference on Decision and Control (CDC)*, 2017, pp. 1278–1283.
[13] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, 08 2014.
[14] A. Sundararajan, B. V. Scoy, and L. Lessard, "Analysis and design of first-order distributed optimization algorithms over time-varying graphs," *IEEE Transactions on Control of Network Systems*, vol. 7, pp. 1597–1608, 2020.
[15] Hanley and C.-K. Li, "Generalized doubly stochastic matrices and linear preservers," *Linear and Multilinear Algebra*, vol. 53, pp. 1–11, 01 2005.
[16] R. Sinkhorn, "Linear transformations under which the doubly stochastic matrices are invariant," vol. 27, 1971, pp. 213–221.
[17] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, 10 2013.
[18] P. Bhunia, S. Bag, and K. Paul, "Bounds for eigenvalues of the adjacency matrix of a graph," *Journal of Interdisciplinary Mathematics*, vol. 22, 06 2019.

APPENDIX
NOTE ON SCALING

We consider general positive values for parameters $R > 0$ and $B > 0$ and we parametrize the step-size by $\alpha = \frac{Rh}{B\sqrt{K}}$, for some $h > 0$. To pass from this general problem to the specific case where $R = 1$ and $B = 1$, we consider the following changes of variables:

$$\tilde{x} = \frac{x}{R}, \quad \tilde{f}(\tilde{x}) = \frac{1}{RB} f(x) \quad \text{and} \quad \tilde{\alpha} = \frac{B\alpha}{R}.$$

These changes of variables do not modify the nature of the problem and allow expressing the worst-case guarantee obtained for $f(x_{\text{av}}) - f(x^*)$ with general values of R, B, and h, denoted $w(R, B, h)$, in terms of the worst-case guarantee obtained for $\tilde{f}(\tilde{x}_{\text{av}}) - \tilde{f}(\tilde{x}^*)$ with $R = B = 1$:

$$w(R, B, h) = RB \; \tilde{w}(1, 1, h). \tag{13}$$

The same kind of scaling can be applied to Theorem 5. The theorem is valid for general values of R and B but is specific to $\alpha = \frac{1}{\sqrt{K}}$, which is equivalent to picking $h = \frac{B}{R}$. After the scaling, we obtain the following bound, valid for $R = 1$, $B = 1$ and any value of $\alpha = \frac{h}{\sqrt{K}}$ with $h > 0$:

$$\tilde{f}(\tilde{x}_{\text{av}}) - \tilde{f}(\tilde{x}^*) \leq \frac{h^{-1} + h}{2\sqrt{K}} + \frac{2h}{\sqrt{K}(1 - \lambda)}. \tag{14}$$

This scaled theoretical bound with $h = 1$ is equivalent to the bound from Theorem 5 with $R = B = 1$, which was the focus of the numerical analysis in Section IV.
This bound (14) can be extended to any value of $R > 0$ and $B > 0$, using the relation from equation (13):

$$f(x_{\text{av}}) - f(x^*) \leq RB \left(\frac{h^{-1} + h}{2\sqrt{K}} + \frac{2h}{\sqrt{K}(1 - \lambda)} \right).$$