Pick your Neighbor: Local Gauss-Southwell Rule for Fast Asynchronous Decentralized Optimization

Marina Costantini, Nikolaos Liakopoulos, Panayotis Mertikopoulos and Thrasyvoulos Spyropoulos

***Abstract*— In decentralized optimization environments, each agent i in a network of n optimization nodes possesses its individual component function f_i, and all nodes in the network communicate with their neighbors to cooperatively minimize the aggregate objective $\sum_{i=1}^n f_i$. In this setting, synchronizing the nodes' updates incurs significant communication overhead and computational costs, so much of the recent literature has focused on the analysis and design of *asynchronous* optimization algorithms where agents activate and communicate with their neighbors at arbitrary times, without being managed by a global synchronization enforcer. Nonetheless, in most of the work on the topic, active nodes select a neighbor to contact based on a fixed probability (e.g., uniformly at random), a choice that ignores the optimization landscape at the moment of activation. Instead, in this work we introduce an optimization-aware selection rule that chooses the neighbor with the highest *dual cost improvement* (a quantity related to a consensus-based dualization of the problem at hand). This scheme is related to the coordinate descent (CD) method with a Gauss-Southwell (GS) rule for coordinate updates; in our setting however, *only a subset of coordinates is accessible at each iteration* (because each node is constrained to communicate only with its direct neighbors), so the existing literature on GS methods does not apply. To overcome this difficulty, we develop a new analytical framework for smooth and strongly convex f_i that covers the class of *set-wise CD* algorithms – a class that directly applies to decentralized scenarios, but is not limited to them – and we show that the proposed set-wise GS rule achieves a speedup by a factor of up to the maximum degree in the network (which is of the order of $\Theta(n)$ in highly connected graphs). The speedup predicted by our theoretical analysis is subsequently validated in a series of numerical experiments with synthetic data.**

I. Introduction

A great number of timely applications require solving optimization problems over a network where nodes can only communicate with their direct neighbors. This may be due to the need of distributing storage and computation loads (e.g. training large machine learning models [1]), or to avoid transferring data that is naturally collected in a decentralized manner, either due to the communication costs or to privacy reasons (e.g. sensor networks [2], edge computing [3]).

Specifically, we consider a setting where the nodes want to solve the decentralized optimization problem

$$\underset{\theta\in\mathbb{R}^d}{\text{minimize}} \quad \sum_{i=1}^n f_i(\theta), \tag{1}$$

where each local function f_i is known only by node i and nodes can exchange optimization values (parameters, gradients) but *not* the local functions themselves. We represent the communication network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes (agents) and $E = |\mathcal{E}|$ edges, which are the links used by the nodes to communicate with their neighbors.

Problem (1) was formally introduced in [4] and widely studied ever since. A convenient reformulation often adopted in the literature assigns to each node a local variable θ_i and forces consensus between node pairs connected by an edge:

$$\underset{\theta_1,\dots,\theta_n\in R^d}{\text{minimize}} \quad \sum_{i=1}^n f_i(\theta_i) \tag{2a}$$

$$\text{subject to} \quad \theta_i = \theta_j \quad \forall\, (i,j) \equiv \ell \in \mathcal{E}, \tag{2b}$$

where $\ell \equiv (i,j)$ indicates that edge $\ell \in \mathcal{E}$ links nodes i and j. Decentralized algorithms to solve (2) allow all nodes to find the minimum value of (1) by just communicating with their neighbors and updating their local variables. This is in contrast with broadcast AllReduce algorithms [5] or parallel distributed architectures [6], which were recently shown to be slower than decentralized schemes in some scenarios [1].

Here we use reformulation (2) to propose an *asynchronous* decentralized algorithm where nodes activate at any time uniformly at random, and once activated they choose one of their neighbors to make an update. Methods with such minimal coordination requirements avoid incurring extra costs of synchronization that may also slow down convergence, which is the reason why many algorithms for this asynchronous setting have been proposed in the literature [7]–[12]. However, most of these works assume that when a node activates, it simply selects the neighbor to contact randomly, based on a predefined probability distribution. This approach overlooks the possibility of letting nodes *choose* the neighbor to contact taking into account the optimization landscape at the time of activation. Therefore, here we depart from the probabilistic choice and ask: *can nodes pick the neighbor smartly to make the optimization process converge faster?*

In this paper we give an affirmative answer and propose an algorithm that achieves this by solving the dual problem of (2). In the dual formulation, there is one dual variable $\lambda_\ell \in \mathbb{R}^d$ per constraint $\theta_i = \theta_j$, hence each dual variable can be associated with an edge ℓ in the graph. Our algorithm lets an activated node i contact a neighbor j so that together they update their shared variable λ_ℓ with a gradient step. In particular, we propose to select the neighbor j such that the updated λ_ℓ is the one *whose directional gradient for the dual function is the largest*, and thus the one that provides

M. Costantini and T. Spyropoulos are with EURECOM, Sophia Antipolis, France ({firstname}.{lastname}@eurecom.fr).

N. Liakopoulos is with Amazon, Luxembourg City, Luxembourg (nliako@amazon.lu).

P. Mertikopoulos is with Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG & Criteo AI Lab (panayotis.mertikopoulos@imag.fr).

the greatest cost improvement at that iteration. Such optimal choice for asynchronous decentralized optimization has not yet been considered in the literature.

Interestingly, the above protocol where a node activates and selects a λ_ℓ to update can be seen as applying the coordinate descent (CD) method [13] to solve the dual problem of (2), with the following key difference: unlike standard CD methods, now *only a small subset of coordinates are accessible at each step*, which are the coordinates associated with the edges connected to the node activated. Moreover, our proposal of updating the λ_ℓ with the largest gradient is similar to the Gauss-Southwell (GS) rule [14], but applied only to the parameters accessible by the activated node.

We name such protocols *set-wise CD* algorithms and we analyze, in particular, both random uniform sampling and the GS rule for the coordinate selection within the accessible set. To the best of our knowledge, convergence rates for set-wise CD schemes have not yet been explored; hence, it is not known what speedup the GS rule can provide compared to uniform sampling in this setting. Furthermore, there are three difficulties that complicate the analysis and constitute the base of our contributions, namely: (i) for arbitrary graphs, the dual problem of (2) has an objective function that is *not* strongly convex, even if the primal functions f_i *are* strongly convex, (ii) the fact that the GS rule is applied to a few coordinates only prevents the use of standard norms to obtain the linear rate, as commonly done for CD methods [13]–[15], and (iii) the fact that the coordinate sets are overlapping (i.e. non-disjoint) makes the problem even harder.

For this reason, we develop a methodology where we prove strong convexity in norms uniquely defined for each algorithm considered. In particular, for the set-wise GS rule this requires relating the norm that we originally define to an alternative norm that considers non-overlapping sets, for which the problem becomes easier and solvable analytically.

Finally, our results also apply to the parallel distributed setting where the parameter vector is stored at a single server and workers can update different subsets of its entries [6], [16], [17]. We show an example in our simulations.

Our contributions can be summarized as follows:

- We introduce the class of *set-wise CD* algorithms and analyze two variants to pick the coordinate to update in the activated set: one that uses uniform sampling (SU-CD), and another that applies the GS rule (SGS-CD).
- We show that this class of algorithms can be used to solve (2) asynchronously, and we provide the linear convergence rates of the two variants considered when the primal functions f_i are smooth and strongly convex.
- To obtain these rates for SU-CD and SGS-CD, we prove strong convexity in uniquely-defined norms that, respectively (i) take into account the graph structure to show strong convexity in the linear subspace where the coordinate updates are applied, and (ii) account for both the random uniform node activation and the application of the GS rule to just a subset of the coordinates.
- We show that the speedup of SGS-CD with respect to SU-CD can be up to $N_{\max}$ (the size of the largest coordinate set), which is analogous to the that of the GS rule with respect to random uniform coordinate sampling in centralized CD [14].

II. Related work

A number of algorithms have been proposed to solve (1) in the asynchronous setup that we consider here. In [12], the activated node chooses a neighbor uniformly at random and both nodes average their primal local values. In [7] the authors adapted the ADMM algorithm to the decentralized setting, but it was the ADMM of [8] the first one shown to converge at the same rate as the centralized ADMM. The algorithm of [9] tracks the average gradients to converge to the exact optimum instead of just a neighborhood around it, as many algorithms back then. The algorithm of [10] can be used on top of directed graphs, which impose additional challenges. A key novelty of our scheme, compared to this line of work, is that we consider the possibility of letting the nodes choose the neighbor to contact in order to make convergence faster.

Work [18] is, to the best of our knowledge, the only work similarly considering smart neighbor selection. The authors propose Max-Gossip, a version of [4] where the activated node averages its local (primal) parameter with that of the neighbor with whom the parameter difference in the largest. They consider convex scalar functions f_i, and use Lyapunov analysis to prove convergence to an optimal value. In contrast, here we obtain linear convergence rates for smooth and strongly convex f_i using duality theory.

Moreover, our rate results for SU-CD and SGS-CD extend the results in [14], where the GS rule was shown to be up to d times faster than uniform sampling for $f : \mathbb{R}^d \to \mathbb{R}$, to the case where this choice is constrained to a subset of the coordinates only, sets have different sizes, each coordinate belongs to exactly two sets, and sets activate uniformly at random. This matches not only the decentralized case, but also parallel distributed settings such as [6], [16], [17].

III. Dual formulation

In this section, we define notation that will be used in the rest of the paper, obtain the dual problem of (2), and analyze the properties of the dual objective function. We will assume throughout that the functions f_i are M_i-smooth and μ_i-strongly convex:

$$f_i(y) \leq f_i(x) + \langle \nabla f(x), y - x \rangle + (M_i/2) \|y - x\|_2^2$$
$$f_i(y) \geq f_i(x) + \langle \nabla f(x), y - x \rangle + (\mu_i/2) \|y - x\|_2^2 .$$

We define the concatenated primal and dual variables $\theta = [\theta_1^T, \ldots, \theta_n^T]^T \in \mathbb{R}^{nd}$ and $\lambda = [\lambda_1^T, \ldots, \lambda_E^T]^T \in \mathbb{R}^{Ed}$, respectively. The graph's incidence matrix $A \in \mathbb{R}^{n \times E}$ has exactly one 1 and one -1 per column ℓ, in the rows corresponding to nodes $i, j : \ell \equiv (i, j)$, and zeros elsewhere (the choice of sign for each node is irrelevant). We call $u_i \in \mathbb{R}^n$ the vector that has 1 in entry i and 0 elsewhere; we define $e_\ell \in \mathbb{R}^E$ in the same way, with the only difference being the dimension. Vectors $\mathbf{1}$ and $\mathbf{0}$ are respectively the all-one and all-zero vectors, and I_d is the $d \times d$ identity matrix. Finally,

we define the block arrays $\Lambda = A \otimes I_d \in \mathbb{R}^{nd \times Ed}$ and $U_i = u_i \otimes I_d \in \mathbb{R}^{nd \times d}$, where $\otimes$ is the Kronecker product.

We can rewrite now (2b) as $\Lambda^T \theta = \mathbf{0}$, and the node variables as $\theta_i = U_i^T \theta$. The minimum value of (2) satisfies:

$$\begin{aligned}\inf_{\theta:\Lambda^T\theta=\mathbf{0}} \sum_{i=1}^n f_i(U_i^T\theta) &\overset{(a)}{=} \inf_\theta \sup_\lambda \left[\sum_{i=1}^n f_i(U_i^T\theta) - \lambda^T\Lambda^T\theta\right] \\ &\overset{(b)}{=} \sup_\lambda \inf_\theta \left[\sum_{i=1}^n f_i(U_i^T\theta) - \lambda^T\Lambda^T\theta\right] \\ &= -\inf_\lambda \sup_\theta \sum_{i=1}^n \left[(U_i^T\Lambda\lambda)^T U_i^T\theta - f_i(U_i^T\theta)\right] \\ &= -\inf_\lambda \sum_{i=1}^n f_i^*(U_i^T\Lambda\lambda) \triangleq -\inf_\lambda F(\lambda), \end{aligned} \tag{3}$$

where (a) holds due to Lagrange duality and (b) holds by strong duality (see e.g. Sec. 5.4 in [19]). Functions f_i^* are the Fenchel conjugates of the f_i, and are defined as

$$f_i^*(y) = \sup_{x \in \mathbb{R}^d} \left(y^T x - f_i(x)\right).$$

Our set-wise CD algorithms converge to the optimal solution of (2) by solving (3). In particular, they update a single dual variable $\lambda_\ell, \ell = 1, \dots, E$ at each iteration and converge to some minimum value λ^* of $F(\lambda)$. Since each λ_ℓ is associated with an edge of the network, the set-wise CD algorithms can run asynchronously.

We now state the convexity properties of F. Since the objective $\sum_{i=1}^n f_i(U_i^T\theta)$ in (2a) is $M_{\max}$-smooth and $\mu_{\min}$-strongly convex in θ, with $M_{\max} = \max_i M_i$ and $\mu_{\min} = \min_i \mu_i$, function F is L-smooth with $L = \frac{\gamma_{\max}}{\mu_{\min}}$, where $\gamma_{\max}$ is the largest eigenvalue of $\Lambda^+\Lambda$ (Sec. 4 in [20]). We also define $\gamma_{\min}^+$ as the smallest non-zero eigenvalue[1] of $\Lambda^+\Lambda$.

However, as shown next, function F is *not* strongly convex in the standard L2 norm, which is a property that facilitates the performance analysis of many linear rate optimization methods in the literature.

Proposition 1. F is not strongly convex in $\|\cdot\|_2$.

Proof. Since Λ does not have full column rank in the general case (i.e., unless the graph is a tree), there exist $w \in \mathbb{R}^{Ed}$ such that $w \neq \mathbf{0}$ and $F(\lambda) = F(\lambda + tw)\ \forall t \in \mathbb{R}$. □

Nevertheless, we can still show linear rates for the set-wise CD algorithms using the fact that $F(\lambda)$ *is* strongly convex in a *linear subspace* of $\mathbb{R}^{Ed}$, as stated next.

Proposition 2 (*Appendix C of* [21]). F is σ_A-strongly convex in the semi-norm $\|\cdot\|_A$, with $\sigma_A = \frac{\gamma_{\min}^+}{M_{\max}}$.

In the definition of the semi-norm $\|x\|_A \triangleq (x^T\Lambda^+\Lambda x)^{\frac{1}{2}}$, Λ^+ denotes the pseudo-inverse of Λ. A key fact for the proofs in the next section is that matrix $\Lambda^+\Lambda$ is a projector onto range(Λ^T), the column space of Λ^T.

In order to make the definitions and notation simpler, in the next section we assume that $d = 1$, so that $\Lambda = A$, $U_i = u_i$, and the gradient $\nabla_\ell F(\lambda) = \frac{\partial F(\lambda)}{\partial \lambda_\ell}$ of $F(\lambda)$ in the direction of λ_ℓ is a scalar. After our theoretical analysis and presentation of the results, in Sec. IV-C we discuss the modifications needed to adapt them to the case $d > 1$.

[1]The "+" stresses that $\gamma_{\min}^+$ is the smallest *strictly positive* eigenvalue.

IV. Set-wise Coordinate Descent Algorithms

In this section we present the *set-wise CD* algorithms, which can solve generic convex problems such as (3) optimally and asynchronously. We propose two set-wise CD algorithms: (i) one where the coordinate to update is selected uniformly at random within the accessible set of coordinates (SU-CD), and (ii) one where we pick the coordinate applying the GS rule to the coordinates in the available set (SGS-CD).

If coordinate ℓ is updated at iteration k, under the simplification $d = 1$ the generic CD update applied to $F(\lambda)$ is:

$$\lambda^{k+1} = \lambda^k - \eta^k \nabla_\ell F(\lambda^k) e_\ell, \tag{4}$$

where η^k is the stepsize. Since $F(\lambda)$ is L-smooth, choosing $\eta^k = 1/L\ \forall k$ guarantees descent at each iteration [14]:

$$F(\lambda^{k+1}) \leq F(\lambda^k) - \frac{1}{2L}\left(\nabla_\ell F(\lambda^k)\right)^2. \tag{5}$$

Eq. (5) will be the departure point to prove the linear convergence rates of SU-CD and SGS-CD.

We now define formally the set-wise CD algorithms.

Definition 1 (*Set-wise CD algorithm*). In a set-wise CD algorithm, every coordinate $\ell = 1, \dots, E$ is assigned to (potentially multiple) sets $\mathcal{S}_i, i = 1, \dots, n$, such that all coordinates belong to at least one set. At any point in time, a set $\mathcal{S}_i$ might activate with uniform probability among the i. When a $\mathcal{S}_i$ activates, the set-wise CD algorithm chooses a single coordinate $\ell \in \mathcal{S}_i$ to update using (4).

The next remark shows how the decentralized problem (2) can be solved asynchronously with set-wise CD algorithms.

Remark 1. By letting (i) the E coordinates in Definition 1 be the dual variables λ_ℓ, and (ii) the $\mathcal{S}_i, i = 1, \dots, n$ be the sets of dual variables corresponding to the edges that are connected to each node i, nodes can run a set-wise CD algorithm to solve (3) (and thus, also (2)) asynchronously.

In light of Remark 1, in the following we illustrate the steps that should be performed by the nodes to run the set-CD algorithms to find a λ^*. We first note that the gradient of $F(\lambda)$ in the direction of λ_ℓ for $\ell \equiv (i,j)$ is

$$\nabla_\ell F(\lambda) = A_{i\ell} \nabla f_i^*(u_i^T A\lambda) + A_{j\ell} \nabla f_j^*(u_j^T A\lambda). \tag{6}$$

Nodes can use (4) and (6) to update the variables λ_ℓ that they have access to (i.e., those corresponding to the edges they are connected to) as follows: each node i keeps in memory the current values of $\lambda_\ell, \ell \in \mathcal{S}_i$, which are needed to compute $\nabla f_i^*(u_i^T A\lambda)$. Then, when edge $\ell \equiv (i,j)$ needs to be updated (either because node i activated and contacted j, or vice versa), both i and j compute their respective terms in the right hand side of (6) and exchange them through their link. Finally, both nodes compute (6) and update their local copy of λ_ℓ applying (4).

TABLE I: Set-related definitions

$\mathcal{S}_i$	Set of edges connected to node i
$\mathcal{N}_i$	Set of neighbors of node i
N_i	Degree of node i, i.e. $N_i = \lvert\mathcal{S}_i\rvert = \lvert\mathcal{N}_i\rvert$
$N_{\max}$	Maximum degree in the network, i.e. $\max_i N_i$
T_i	Selector matrix of set $\mathcal{S}_i$ (see Definition 2)
$\mathcal{S}'_i$	Subset $\mathcal{S}'_i \subseteq \mathcal{S}_i$ such that $\mathcal{S}'_i \cap \mathcal{S}'_j = \emptyset$ if $i \neq j$
T'_i	Selector matrix of set $\mathcal{S}_i$
$\overline{\mathcal{S}'_i}$	Complement set of $\mathcal{S}'_i$ such that $\overline{\mathcal{S}'_i} = \mathcal{S}_i \setminus \mathcal{S}'_i$
$\overline{T'_i}$	Selector matrix of set $\overline{\mathcal{S}'_i}$

Algorithms 1 and 2 below detail these steps for SU-CD and SGS-CD, respectively. In the algorithms we have used $\mathcal{N}_i$ to indicate the set of neighbors of node i (note that $\mathcal{S}_i = \{\ell : \ell \equiv (i,j), j \in \mathcal{N}_i\}$). Table I shows this and other set-related notation that will be frequently used in the sections that follow.

We now proceed to describe the SU-CD and SGS-CD algorithms in detail, and prove their linear convergence rates.

A. Set-wise Uniform CD (SU-CD)

In SU-CD, the activated node chooses the neighbor uniformly at random, as shown in Alg. 1. We can compute the per-iteration progress of SU-CD taking expectation in (5):

$$\begin{aligned}\mathbb{E}\Big[F(\lambda^{k+1}) \mid \lambda^k\Big] &\leq F(\lambda^k) - \frac{1}{2L}\mathbb{E}\Big[\big(\nabla_\ell F(\lambda^k)\big)^2 \mid \lambda^k\Big] \\ &= F(\lambda^k) - \frac{1}{2Ln}\sum_{i=1}^{n}\frac{1}{N_i}\sum_{\ell\in\mathcal{S}_i}\big(\nabla_\ell F(\lambda^k)\big)^2 \\ &\leq F(\lambda^k) - \frac{1}{LnN_{\max}}\left\|\nabla F(\lambda^k)\right\|_2^2 \qquad (7)\end{aligned}$$

where $N_i = |\mathcal{S}_i|$ and $N_{\max} = \max_i N_i$.

The standard procedure to show the linear convergence of CD in the centralized case is to lower-bound $\|\nabla F(\lambda)\|_2^2$ using the strong convexity of the function [13], [14]. However, since F is *not* strongly convex (Prop. 1), we cannot apply this procedure to get the linear rate of SU-CD.

We can, however, use the strong convexity of F in $\|\cdot\|_A$ instead (Prop. 2). The next result gives the core of the proof.

Proposition 3. It holds that

$$\|\nabla F(\lambda)\|_2 = \|\nabla F(\lambda)\|_A = \|\nabla F(\lambda)\|_A^*\,,$$

where $\|\cdot\|_A^*$ is the dual norm of $\|\cdot\|_A$, defined as (see e.g. Sec. A.1.6 in [19])

$$\|z\|_A^* = \sup_{x\in\mathbb{R}^d}\left\{z^T x \,\middle|\, \|x\|_A \leq 1\right\}. \qquad (8)$$

Proof. Note that $\forall w \neq \mathbf{0}$ such that $F(\lambda + tw) = F(\lambda)\ \forall t$, it holds that $w^T\nabla F(\lambda) = 0$ and thus $\nabla F(\lambda) \in \text{range}(A^T)$. This means that $A^+A\nabla F(\lambda) = I_E\nabla F(\lambda)$, and therefore it holds that $\|\nabla F(\lambda)\|_A = \|\nabla F(\lambda)\|_2$. Finally, since the dual norm of the L2 norm is again the L2 norm, we have that also $\|\nabla F(\lambda)\|_A^* = \|\nabla F(\lambda)\|_2$, which gives the result. □

We now use Prop. 3 to prove the linear rate of SU-CD.

Algorithm 1 Set-wise Uniform CD (SU-CD)

1: **Input:** Functions f_i, step η, incidence matrix A, graph $\mathcal{G}$
2: Initialize $\theta_i^0, i = 1,\dots,n$ and $\lambda_\ell^0, \ell = 1,\dots,E$
3: **for** $k = 1, 2, \dots$ **do**
4: Sample activated node $i \in \{1,\dots,n\}$ uniformly
5: Node i picks neighbor $j \leftarrow \mathcal{U}\{h : h \in \mathcal{N}_i\}$
6: Node i computes $\nabla f_i^*(u_i^T A\lambda)$ and sends it to j
7: Node j computes $\nabla f_j^*(u_j^T A\lambda)$ and sends it to i
8: Nodes i, j: $(i,j) \equiv \ell$ use (6) to update their local copies of λ_ℓ by $\lambda_\ell^k \leftarrow \lambda_\ell^{k-1} - \eta\nabla_\ell F(\lambda)$
9: $\lambda_m^k \leftarrow \lambda_m^{k-1}\ \forall$ edges $m \neq \ell$

Proposition 4 (*Rate of SU-CD*). SU-CD converges as

$$\mathbb{E}\big[F(\lambda^{k+1}) \mid \lambda^k\big] - F(\lambda^*) \leq \left(1 - \frac{2\sigma_A}{LnN_{\max}}\right)\big[F(\lambda^k) - F(\lambda^*)\big]\,.$$

Proof. Since $F(\lambda)$ is strongly convex in $\|\cdot\|_A$ with strong convexity constant σ_A (Prop. 2), it holds

$$F(y) \geq F(x) + \langle\nabla F(x), y - x\rangle + \frac{\sigma_A}{2}\|y - x\|_A^2\,.$$

Minimizing both sides of the above equation respect to y as in Sec. 4 in [14] we get

$$F(x^*) \geq F(x) - \frac{1}{2\sigma_A}\left(\|\nabla F(x)\|_A^*\right)^2, \qquad (9)$$

and rearranging terms we can lower-bound $\left(\|\nabla F(x)\|_A^*\right)^2$.

Finally, we can use Prop. 3 to replace $\left(\|\nabla F(x)\|_2\right)^2$ with $\left(\|\nabla F(x)\|_A^*\right)^2$ in (7), and use the lower bound on $\left(\|\nabla F(x)\|_A^*\right)^2$ given by (9) to get the result. □

Note that vector λ has $\frac{1}{2}\sum_i N_i = E \leq \frac{nN_{\max}}{2}$ coordinates, where the inequality holds with equality for regular graphs. We make the following remark.

Remark 2. If $\mathcal{G}$ is regular, the linear convergence rate of SU-CD is $\frac{\sigma_A}{LE}$, which matches the rate of centralized uniform CD for strongly convex functions [13], [14], with the only difference that now the strong convexity constant σ_A is defined over norm $\|\cdot\|_A$.

In the next section we analyze SGS-CD and show that its convergence rate can be up to $N_{\max}$ times that of SU-CD.

B. Set-wise Gauss-Southwell CD (SGS-CD)

In SGS-CD, as shown in Alg. 2, the activated node i selects the neighbor j to contact applying the GS rule within the edges in $\mathcal{S}_i$:

$$\ell = \operatorname*{argmax}_{m\in\mathcal{S}_i}\left(\nabla_m F(\lambda)\right)^2,$$

and then j satisfies $\ell \equiv (i,j)$. In order to make this choice, all nodes $h \in \mathcal{N}_i$ must send their $\nabla f_h^*(u_h^T A\lambda)$ to node i (line 5 in Alg. 2). We discuss this additional communication step of SGS-CD with respect to SU-CD in Sec. VI.

Algorithm 2 Set-wise Gauss-Southwell CD (SGS-CD)

Input: Functions f_i, step η, incidence matrix A, graph $\mathcal{G}$
Initialize $\theta_i^0, i = 1, \dots, n$ and $\lambda_\ell^0, \ell = 1, \dots, E$
for $k = 1, 2, \dots$ **do**
  Sample activated node $i \in \{1, \dots, n\}$ uniformly
  All $h \in \mathcal{N}_i$ compute $\nabla f_h^*(u_h^T A\lambda)$ and send it to i
  Node i computes $\nabla f_i^*(u_i^T A\lambda)$
  Compute $\nabla_\ell F(\lambda)\ \forall \ell \in \mathcal{S}_i$ (equivalently, $\forall h \in \mathcal{N}_i$) with (6) using the received $\nabla f_h^*(u_h^T A\lambda)$
  Node i selects $j \leftarrow \max_{h\in\mathcal{N}_i} |\nabla_\ell F(\lambda)|, \ell \equiv (i,h)$
  Node i sends $\nabla f_i^*(u_i^T A\lambda)$ to j
  Nodes i, j: $(i,j) \equiv \ell$ use (6) to update their local copies of λ_ℓ by $\lambda_\ell^k \leftarrow \lambda_\ell^{k-1} - \eta \nabla_\ell F(\lambda)$
  $\lambda_m^k \leftarrow \lambda_m^{k-1}\ \forall$ edges $m \neq \ell$

To obtain the convergence rate of SGS-CD we will follow the steps taken for SU-CD in the proof of Prop. 4. As done for SU-CD, we start by computing the per-iteration progress taking expectation in (5) for SGS-CD:

$$\mathbb{E}\left[F(\lambda^{k+1}) \mid \lambda^k\right] \leq F(\lambda^k) - \frac{1}{2Ln} \sum_{i=1}^{n} \max_{\ell\in\mathcal{S}_i} \left(\nabla_\ell F(\lambda^k)\right)^2. \tag{10}$$

Given this per-iteration progress, to proceed as we did for SU-CD we need to show (i) that the sum on the right hand side of (10) defines a norm, and (ii) that strong convexity holds in its dual norm. We start by defining the selector matrices T_i, which will significantly simplify notation.

Definition 2 (***Selector matrices***)**.** The selector matrices $T_i \in \{0,1\}^{N_i \times E}, i = 1, \dots, n$ select the coordinates of a vector in $\mathbb{R}^E$ that belong to set $\mathcal{S}_i$. Note that any vertical stack of the unitary vectors $\left\{e_\ell^T\right\}_{\ell\in\mathcal{S}_i}$ gives a valid T_i.

We can now show that the sum in (10) is a (squared) norm. Since the operation involves applying $\max(\cdot)$ within each set $\mathcal{S}_i$, we will denote this norm $\|x\|_{\text{SM}}$, where the subscript SM stands for "Set-Max".

Proposition 5. The function $\|x\|_{\text{SM}} \triangleq \sqrt{\sum_{i=1}^n \|T_i x\|_\infty^2} = \sqrt{\sum_{i=1}^n \max_{j\in\mathcal{S}_i} x_j^2}$ is a norm in $\mathbb{R}^E$.

Proof. Using $\max_{j\in\mathcal{S}_i}\left(x_j^2+y_j^2\right) \leq \max_{j\in\mathcal{S}_i} x_j^2 + \max_{j\in\mathcal{S}_i} y_j^2$ and $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ we can show that $\|\cdot\|_{\text{SM}}$ satisfies the triangle inequality. It is straightforward to show that $\|\alpha x\|_{\text{SM}} = |\alpha| \, \|x\|_{\text{SM}}$ and $\|x\|_{\text{SM}} = 0$ iff $x = \mathbf{0}$. □

Following the proof of Prop. 4, we would like to show that F is strongly convex in the dual norm $\|\cdot\|_{\text{SM}}^*$. Furthermore, we would like to compare the strong convexity constant σ_{SM} with σ_A to quantify the speedup of SGS-CD with respect to SU-CD. It turns out, though, that computing $\|\cdot\|_{\text{SM}}^*$ is not easy at all; the main difficulty stems from the fact that sets $\mathcal{S}_i$ are overlapping (or non-disjoint), since each coordinate $\ell \equiv (i,j)$ belongs to both $\mathcal{S}_i$ and $\mathcal{S}_j$. The first scheme in Figure 1 illustrates this fact for the 3-node clique.

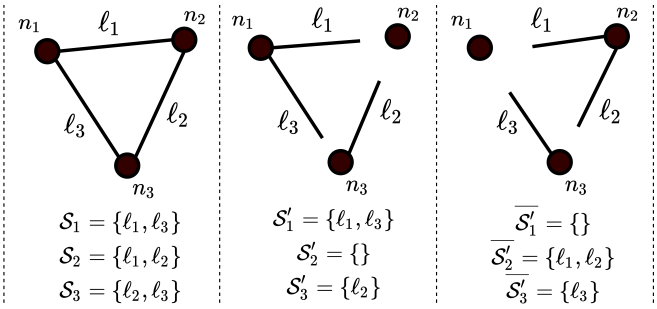

Fig. 1: Example of sets $\mathcal{S}_i$ and *one possibility* for $\mathcal{S}'_i$ and $\overline{\mathcal{S}'_i}$

To circumvent this issue, we define a new norm $\|\cdot\|_{\text{SMNO}}^*$ ("Set-Max Non-Overlapping") that we can directly relate to $\|\cdot\|_{\text{SM}}^*$ (Prop. 6) and whose value we can compute explicitly (Prop. 7), which will later allow us to relate its strong convexity constant σ_{SMNO} to σ_A (Prop. 8).

Definition 3 (***Norm*** $\|\cdot\|_{\textit{SMNO}}^*$)**.** We assume that each coordinate $\ell \equiv (i,j)$ is assigned to only one of the sets $\mathcal{S}'_i \subseteq \mathcal{S}_i$ or $\mathcal{S}'_j \subseteq \mathcal{S}_j$, such that the new sets $\{\mathcal{S}'_i\}_{i=1}^n$ are non-overlapping (some sets can be empty), and all coordinates ℓ belong to exactly one set in $\{\mathcal{S}'_i\}$. We name the selector matrices of these new sets T'_i, so that each possible choice of $\{\mathcal{S}'_i\}$ defines a different set $\{T'_i\}$. Then, we define

$$\|z\|_{\text{SMNO}}^* = \sup_x \left\{ z^T x \;\middle|\; \sqrt{\sum_{i=1}^n \|T_i^* x\|_\infty^2} \leq 1 \right\}, \tag{11}$$

with the choice of non-overlapping sets

$$\{T_i^*\} = \arg\max_{\{T'_i\}} \sum_{i=1}^n \|T'_i x\|_\infty^2. \tag{12}$$

Note that the maximizations in (11) and (12) are coupled. We denote the value of x that attains (11) as x_{SMNO}^*.

The definition of sets $\mathcal{S}'_i$ corresponds to assigning each edge ℓ to one of the two nodes at its endpoints, as illustrated in the second scheme of Figure 1. Therefore, for each possible pair $(\{\mathcal{S}'_h\}, \{T'_h\})$ we can define a complementary pair $(\{\overline{\mathcal{S}'_h}\}, \{\overline{T'_h}\})$ such that if $\ell \equiv (i,j)$ was assigned to $\mathcal{S}'_i$ in $\{\mathcal{S}'_h\}$, then it is assigned to $\overline{\mathcal{S}'_j}$ in $\{\overline{\mathcal{S}'_h}\}$. This corresponds to assigning ℓ to the opposite endpoint (node) to the one originally chosen, as shown in the third scheme of Figure 1. With these definitions, it holds (potentially with some permutation of the rows):

$$T_i = \begin{bmatrix} T'_i \\ \overline{T'_i} \end{bmatrix} = \begin{bmatrix} T'_i \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \overline{T'_i} \end{bmatrix}, \; i = 1, \dots, n.$$

We remark that the equality above holds for *any* $\{T'_i\}$ corresponding to a feasible assignment $\{\mathcal{S}'_i\}$, and in particular it hols for $(\{\mathcal{S}_i^*\}, \{T_i^*\})$. This fact is used in the proof of the following proposition, which relates norms $\|\cdot\|_{\text{SM}}^*$ and $\|\cdot\|_{\text{SMNO}}^*$. This will allow us to complete the analysis with $\|\cdot\|_{\text{SMNO}}^*$, which we can compute explicitly (Prop. 7).

Proposition 6. The value of the dual norm of $\|\cdot\|_{\text{SM}}$, denoted $\|\cdot\|_{\text{SM}}^*$, satisfies $\left(\|\cdot\|_{\text{SM}}^*\right)^2 \geq \frac{1}{2}\left(\|\cdot\|_{\text{SMNO}}^*\right)^2$.

Proof. By definition

$$\|z\|_{\text{SM}}^* = \sup_x \left\{ z^T x \;\middle|\; \sqrt{\sum_{i=1}^n \|T_i x\|_\infty^2} \leq 1 \right\}.$$

By inspection we can tell that the x that attains the supremum, denoted x_{SM}^*, will satisfy $\sum_{i=1}^n \|T_i x_{\text{SM}}^*\|_\infty^2 = 1$. We note now that

$$\sum_{i=1}^n \|T_i x\|_\infty^2 = \sum_{i=1}^n \left\| \begin{bmatrix} T_i' \\ \mathbf{0} \end{bmatrix} x + \begin{bmatrix} \mathbf{0} \\ \overline{T_i'} \end{bmatrix} x \right\|_\infty^2$$
$$\leq \sum_{i=1}^n \|T_i' x\|_\infty^2 + \sum_{i=1}^n \left\|\overline{T_i'} x\right\|_\infty^2 \leq 2 \sum_{i=1}^n \left\|\widehat{T_i'} x\right\|_\infty^2, \quad (13)$$

with

$$\{\widehat{T_i'}\} = \arg \max_{\{T_i'\},\{\overline{T_i'}\}} \left(\sum_{i=1}^n \|T_i' x\|_\infty^2, \sum_{i=1}^n \left\|\overline{T_i'} x\right\|_\infty^2 \right). \quad (14)$$

Note that if we evaluate (14) at x_{SMNO}^*, due to (12) we have $\{\widehat{T_i'}\} = \{T_i^*\}$. Also, again by inspection (now of problem (11)) we know that x_{SMNO}^* satisfies $\sum_{i=1}^n \|T_i^* x_{\text{SMNO}}^*\|_\infty^2 = 1$. Therefore, (13) says

$$\frac{1}{2} \sum_{i=1}^n \|T_i x_{\text{SMNO}}^*\|_\infty^2 = \sum_{i=1}^n \left\| T_i \frac{x_{\text{SMNO}}^*}{\sqrt{2}} \right\|_\infty^2 \leq 1,$$

from where we conclude that coordinate-wise it must hold $x_{\text{SM}}^* \succeq \frac{1}{\sqrt{2}} x_{\text{SMNO}}^*$, and thus $\|z\|_{\text{SM}}^* \geq \frac{1}{\sqrt{2}} \|z\|_{\text{SMNO}}^*$. □

The next proposition gives the value of $\|x\|_{\text{SMNO}}^*$ explicitly, which will be needed to compare the strong convexity constant σ_{SMNO} with σ_A.

Proposition 7. It holds that $\|x\|_{\text{SMNO}}^* = \sqrt{\sum_{i=1}^n \|T_i^* x\|_1^2}$.

Proof. Since the sets $\{\mathcal{S}_i^*\}$ are non-overlapping and in (11) norm $\|\cdot\|_\infty$ is applied per-set, the entries x_ℓ of x_{SMNO}^* will have $|x_\ell| = x^{(i)} \geq 0 \; \forall \; \ell \in \mathcal{S}_i^*$ and the sign will match that of the entries of z, i.e. $\text{sign}(x_\ell) = \text{sign}(z_\ell)$. The maximization of (11) then becomes

$$\begin{aligned} \underset{\{x^{(i)}\}}{\text{maximize}} \quad & \sum_{i=1}^n \sum_{\ell \in \mathcal{S}_i^*} \left(|z_\ell| \cdot x^{(i)} \right) \\ \text{subject to} \quad & \sqrt{\sum_{i=1}^n \left(x^{(i)}\right)^2} \leq 1. \end{aligned}$$

Factoring out $x^{(i)}$ in the objective and noting that $\sum_{\ell \in \mathcal{S}_i^*} |z_\ell| = \|T_i^* z\|_1$, we can define $w = [x^{(1)}, \ldots, x^{(n)}]^T$ and $y = [\|T_1^* z\|_1, \ldots, \|T_n^* z\|_1]^T$ so that (11) now reads

$$\|z\|_{\text{SMNO}}^* = \sup_w \left\{ y^T w \;\middle|\; \|w\|_2 \leq 1 \right\}.$$

The right hand side is the definition of the dual of the L2 norm evaluated at y. Since the dual of the L2 norm is again the L2, we have $\|z\|_{\text{SMNO}}^* = \|y\|_2 = \sqrt{\sum_{i=1}^n \|T_i' z\|_1^2}$. □

We can now prove the linear convergence rate of SGS-CD.

Proposition 8 (***Rate of SGS-CD***)**.** SGS-CD converges as

$$\mathbb{E}\left[F(\lambda^{k+1}) \mid \lambda^k\right] - F(\lambda^*) \leq \left(1 - \frac{2\sigma_{\text{SMNO}}}{Ln}\right) \left[F(\lambda^k) - F(\lambda^*)\right],$$

with

$$\frac{\sigma_A}{N_{\max}} \leq \sigma_{\text{SMNO}} \leq \sigma_A. \quad (15)$$

Proof. We start by proving (15) by showing that strong convexity in $\|\cdot\|_A$ implies strong convexity in $\|\cdot\|_{\text{SMNO}}^*$, which will give the inequalities as a by-product of the analysis. Below we assume that $x \in \text{range}(A^T)$; the results here can then be directly applied to the proofs above because $\|\cdot\|_A, \|\cdot\|_{\text{SM}}, \|\cdot\|_{\text{SMNO}}$ and their duals are applied to $\nabla F(\lambda)$, which is always in $\text{range}(A^T)$ (Prop 3).

For $x \in \text{range}(A^T)$ it holds that (Props. 3 and 7):

$$\|x\|_A^2 = \|x\|_2^2 = \sum_{i=1}^E x_i^2 = \sum_{i=1}^n \|T_i^* x\|_2^2$$
$$\left(\|x\|_{\text{SMNO}}^*\right)^2 = \sum_{i=1}^n \|T_i^* x\|_1^2.$$

We also note that, using the Cauchy-Schwarz inequality and denoting $[v]_i$ the i^{th} entry of vector v, it holds both that

$$\sum_{i=1}^n \|T_i^* x\|_2^2 \leq \sum_{i=1}^n \left(\sum_{j \in \mathcal{S}_i^*} |x_j| \right)^2 = \sum_{i=1}^n \|T_i^* x\|_1^2, \text{ and}$$
$$\sum_{i=1}^n \|T_i^* x\|_1^2 = \sum_{i=1}^n \left(\mathbf{1}^T \left[\left| [T_i^* x]_1 \right|, \ldots, \left| [T_i^* x]_{N_i^*} \right| \right]^T \right)^2$$
$$\overset{\text{C.S.}}{\leq} \sum_{i=1}^n N_i^* \|T_i^* x\|_2^2 \leq N_{\max} \sum_{i=1}^n \|T_i^* x\|_2^2,$$

where $N_i^* = |\mathcal{S}_i^*|$. We can summarize these relations as

$$\frac{1}{N_{\max}} \left(\|x\|_{\text{SMNO}}^*\right)^2 \leq \|x\|_A^2 \leq \left(\|x\|_{\text{SMNO}}^*\right)^2.$$

Using these inequalities in the strong convexity definitions, similarly to [14], we get both

$$\begin{aligned} F(y) &\geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_A}{2} \left(\|y - x\|_A\right)^2 \\ &\geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_A}{2N_{\max}} \left(\|y - x\|_{\text{SMNO}}^*\right)^2, \end{aligned} \quad (16)$$

and

$$\begin{aligned} F(y) &\geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_{\text{SMNO}}}{2} \left(\|y - x\|_{\text{SMNO}}^*\right)^2 \\ &\geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_{\text{SMNO}}}{2} \left(\|y - x\|_A\right)^2. \end{aligned} \quad (17)$$

Equation (16) says that F is at least $\frac{\sigma_A}{N_{\max}}$-strongly convex in $\|\cdot\|_{\text{SMNO}}^*$, and eq. (17) says that F is at least σ_{SMNO}-strongly convex in $\|\cdot\|_A$. Together they imply (15).

To get the rate of SGS-CD, and following the procedure of SU-CD, we need to lower-bound the per-iteration progress $\frac{1}{2Ln} \|\nabla F(\lambda)\|_{\text{SM}}^2$ in (10). For this we will use the strong convexity in $\|\cdot\|_{\text{SM}}^*$, which we can obtain from the strong convexity that we just proved for $\|\cdot\|_{\text{SMNO}}^*$, as shown next.

Stating that F is at least σ_{SM}-strongly convex in $\|\cdot\|^*_{\text{SM}}$ and using Prop. 6 we obtain:

$$\begin{aligned} F(y) &\geq F(x) + \langle \nabla F(x), y-x\rangle + \frac{\sigma_{\text{SM}}}{2}\left(\|y-x\|^*_{\text{SM}}\right)^2 \\ &\geq F(x) + \langle \nabla F(x), y-x\rangle + \frac{\sigma_{\text{SM}}}{2}\frac{1}{2}\left(\|y-x\|^*_{\text{SMNO}}\right)^2, \end{aligned} \tag{18}$$

from where we conclude that $\sigma_{\text{SM}} = 2\sigma_{\text{SMNO}}$.

Minimizing both sides of the first inequality in (18) respect to y we obtain

$$F(x^*) \geq F(x) - \frac{1}{2\sigma_{\text{SM}}}\left(\|\nabla F(x)\|_{\text{SM}}\right)^2, \tag{19}$$

which is analogous to (9), and rearranging terms gives a lower bound on $\|\nabla F(\lambda)\|^2_{\text{SM}}$. Using this lower bound in (10) and replacing $\sigma_{\text{SM}} = 2\sigma_{\text{SMNO}}$ gives the rate of SGS-CD. □

Proposition 8 states that SGS-CD can be up to $N_{\max}$ faster than SU-CD. This result is analogous to that of [14] for the GS rule respect to uniform sampling in centralized CD.

Although this is an upper bound and may not always be achievable, we can think of the following scenario where this gain is attained: let all sets have the same size $|\mathcal{S}_i| = N_{\max}\ \forall i$, exactly m out of the $N_{\max}$ coordinates in each set have $\nabla_m F(\lambda) = 0$, and only one ℓ have $\nabla_\ell F(\lambda) \neq 0$. In this case, on average *only* $\frac{1}{N_{\max}}$ *times will SU-CD choose the coordinate that gives some improvement*, while SGS-CD will do it at all iterations.

Note that this example requires the gradients of all coordinates to be independent, which is not verified in the decentralized optimization setting: according to eq. (6), for a $\nabla_m F$ to be zero, it must hold that $\nabla f^*_i = \nabla f^*_j, m \equiv (i,j)$. But unless this equality holds for *all* $(i,j) \in \mathcal{E}$ (i.e., unless the minimum has been attained), λ will continue changing, and the ∇f^*_i will differ. Thus, the gains of SGS-CD in this setting may not attain the upper bound.

Nevertheless, when it comes to parallel distributed setups, the coordinates are not necessarily coupled as in the decentralized case, and thus the $N_{\max}$ speedup of SGS-CD is still achievable, as shown in our simulations below.

C. Case $d > 1$

To extend the proofs above for $d > 1$, the block arrays Λ and U_i should be used instead of A and u_i, and the selector matrices T_i should be redefined in the same way. Then, all the operations that in the proofs above are applied *per entry* (scalar coordinate) of the vector λ, should now be applied to *the magnitude* of each vector coordinate $\lambda_\ell \in \mathbb{R}^d$ of $\lambda \in \mathbb{R}^{Ed}$. Also, since $\nabla_m F \in \mathbb{R}^d$, in this case the GS rule becomes $\text{argmax}_{m\in\mathcal{S}_i} \|\nabla_m F(\lambda)\|^2_2$.

V. Numerical Results

Figure 2 shows the remarkable speedup of SGS-CD with respect to SU-CD in both the decentralized (left plots) and the parallel distributed (right plots) settings.

For the decentralized setting we created two regular graphs of $n = 24$ nodes and degrees $N_{\max} = 8$ and 12, respectively. The local functions were $f_i(\theta) = \theta^T c I_d \theta$ with $d = 5$, and $c = 50$ if (i modulo $N_{\max}$) $= 0$ and $c = 1$ otherwise, where i is the index of each node. We chose these f_i so that each node would have (approximately) one neighbor out of the $N_{\max}$ with whom the coordinate gradient would have maximum disagreement, thus maximizing the chances of observing differences between SU-CD and SGS-CD.

For the parallel distributed setting, we created a problem that was separable per-coordinate, and we tried to recreate the conditions described in the previous section to approximate the $N_{\max}$ gain. We chose $F(x) = x^T D x$ with $x \in \mathbb{R}^d$ and $d = 48$. Matrix D was diagonal with its non-zero entries sampled from $\mathcal{N}(10,3)$. We then created n sets of $N_{\max}$ coordinates such that each coordinate belonged to exactly two sets, similarly to the parallel distributed scenario with parameter server where each worker has access to a subset of the coordinates only. We simulated two different distributions of the $d = 48$ coordinates: one with $n = 24$ sets of $N_{\max} = 4$ coordinates each, and another with $n = 12$ sets of $N_{\max} = 8$ coordinates each. Following the reasoning in the previous section, we set the initial value of $(N_{\max} - 1)$ coordinates in each set to $x^0_m = 1$ (close to the optimal value $x^*_m = 0$), and the one remaining to $x^0_\ell = 100$ (far away from $x^*_\ell = 0$).

The plots in Figure 2 show the steep rate gain of SGS-CD with respect to SU-CD as $N_{\max}$ increases. To quantify this gain we denoted $(1-\rho)$ the suboptimality reduction factor and we estimated ρ_U, ρ_G for SU-CD and SGS-CD, respectively, from the last third of the suboptimality curves. Proposition 8 says that $1 \leq \frac{\rho_G}{\rho_U} \leq N_{\max}$, and indeed, this is verified in both settings. In particular, for the decentralized setting $\frac{\rho_G}{\rho_U}$ is approximately in the middle of this range for both regular graphs. In the parallel distributed setting, however, the ratio is much closer to $N_{\max}$, as predicted.

VI. Discussion

We have presented the class of *set-wise CD* algorithms, where in a multi-agent system workers are allowed to modify only a subset of the total number of coordinates at each iteration. These algorithms are suitable for asynchronous decentralized optimization and distributed parallel optimization. We studied specifically two set-wise CD variants, SU-CD and SGS-CD, which required developing a new methodology that extends previous results for CD.

We obtained the convergence rates of SU-CD and SGS-CD for smooth and strongly convex functions f_i and showed that they are analogous, except for the network-related parameters, to those given in [14] for their centralized counterparts. More precisely, we showed that SGS-CD can be up to $N_{\max}$ (the size of the largest coordinate set) times faster than SU-CD; we further elaborated on the conditions under which such speedup may be attainable, and confirmed these predictions with numerical simulations.

A limitation of SGS-CD with respect to SU-CD is that all the neighbors of the activated node i must compute their ∇f^*_h and send it to i (line 5 in Alg. 2). This additional overhead with respect to SU-CD is analogous to that of the GS rule in centralized CD, which is the reason why GS makes sense only for problems with certain separability

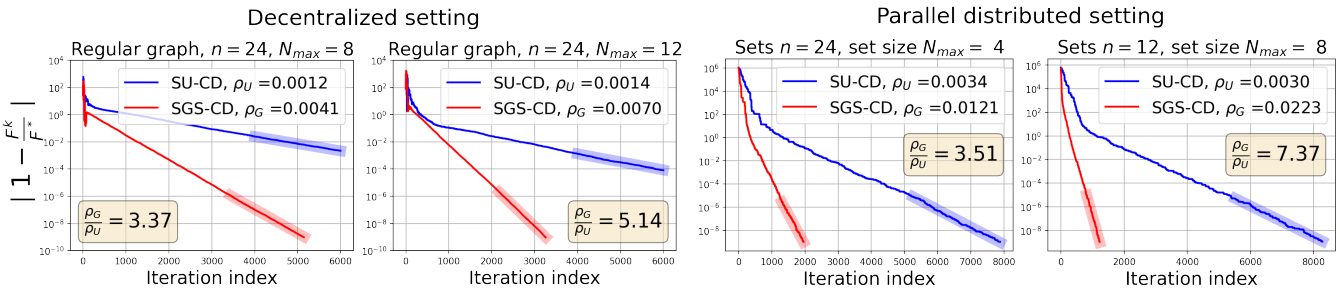

Fig. 2: Comparison of the convergence rates of SU-CD and SGS-CD in two settings: decentralized optimization over a network (left plots), and parallel distributed computation with parameter server (right plots). Given the linear suboptimality reduction $F(\lambda^k) - F(\lambda^*) \leq (1-\rho)^k [F(\lambda^0) - F(\lambda^*)]$, the thick transparent lines show the part of the curves used to estimate ρ_U and ρ_G for SU-CD and SGS-CD, respectively. The ratio $\frac{\rho_G}{\rho_U}$ increases notably with $N_{\max}$, in agreement with the theory.

and sparsity structures [14], [15]. Designing algorithms that approximate SGS-CD at the cost of SU-CD are a subject of future work.

A possibility that was not accounted for in this study is letting the nodes use different stepsizes to update each λ_ℓ [15]. Indeed, function F is coordinate-wise smooth with constant $L_\ell = \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)$ for $\ell \equiv (i,j)$; this could be used by each node to choose a different stepsize $\eta_\ell \geq \frac{1}{L}$ for each λ_ℓ, which would make convergence faster. Methods to estimate the per-coordinate smoothness when it is not known a priori were discussed in [13], [15].

Another way of obtaining faster convergence is to use Nesterov acceleration, as done in [21], now *on top* of the smart neighbor choosing rule. Although this would entail partially sacrificing the complete lack of coordination allowed by the set-wise CD algorithms presented here (because acceleration couples the coordinate updates), combined with per-coordinate specific stepsizes and well-designed neighbor sampling rules it would open the possibility of obtaining *the fastest* decentralized set-wise CD algorithm, similarly to recent results for accelerated centralized CD [22], [23].

REFERENCES

[1] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *2009 International Conference on Information Processing in Sensor Networks*, pp. 49–60, IEEE, 2009.

[3] M. Alrowaily and Z. Lu, "Secure edge computing in IoT systems: review and case studies," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 440–444, IEEE, 2018.

[4] A. Nedic and A. Ozdaglar, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," in *2007 46th IEEE Conference on Decision and Control*, pp. 4711–4716, IEEE, 2007.

[5] R. Rabenseifner, "Optimization of collective reduction operations," in *International Conference on Computational Science*, pp. 1–9, Springer, 2004.

[6] L. Xiao, A. W. Yu, Q. Lin, and W. Chen, "DSCOVR: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1634–1691, 2019.

[7] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *52nd IEEE Conference on Decision and Control (CDC)*, pp. 3671–3676, IEEE, 2013.

[8] E. Wei and A. Ozdaglar, "On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 551–554, IEEE, 2013.

[9] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 434–448, 2017.

[10] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push–pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, 2020.

[11] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[12] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3581–3586, IEEE, 2009.

[13] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[14] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, "Coordinate descent converges faster with the Gauss-Southwell rule than random selection," in *International Conference on Machine Learning*, pp. 1632–1641, PMLR, 2015.

[15] J. Nutini, I. Laradji, and M. Schmidt, "Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence," *arXiv preprint arXiv:1712.08859*, 2017.

[16] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[17] Z. Peng, Y. Xu, M. Yan, and W. Yin, "Arock: an algorithmic framework for asynchronous parallel coordinate updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[18] A. Verma, M. M. Vasconcelos, U. Mitra, and B. Touri, "Max-gossip subgradient method for distributed optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3130–3136, IEEE, 2021.

[19] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "A dual approach for optimal algorithms in distributed optimization over networks," in *2020 Information Theory and Applications Workshop (ITA)*, pp. 1–37, IEEE, 2020.

[21] H. Hendrikx, F. Bach, and L. Massoulié, "Accelerated decentralized optimization with local updates for smooth and strongly convex objectives," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 897–906, PMLR, 2019.

[22] Y. Nesterov and S. U. Stich, "Efficiency of the accelerated coordinate descent method on structured optimization problems," *SIAM Journal on Optimization*, vol. 27, no. 1, pp. 110–123, 2017.

[23] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan, "Even faster accelerated coordinate descent using non-uniform sampling," in *International Conference on Machine Learning*, pp. 1110–1119, PMLR, 2016.