Random Block Coordinate Descent Methods for Linearly Constrained Optimization over Networks

Ion Necoara · Yurii Nesterov · François Glineur

Received: 18 April 2016 / Accepted: 30 December 2016 / Published online: 12 January 2017
© Springer Science+Business Media New York 2017

Abstract In this paper we develop random block coordinate descent methods for minimizing large-scale linearly constrained convex problems over networks. Since coupled constraints appear in the problem, we devise an algorithm that updates in parallel at each iteration at least two random components of the solution, chosen according to a given probability distribution. Those computations can be performed in a distributed fashion according to the structure of the network. Complexity per iteration of the proposed methods is usually cheaper than that of the full gradient method when the number of nodes in the network is much larger than the number of updated components. On smooth convex problems, we prove that these methods exhibit a sublinear worst-case convergence rate in the expected value of the objective function. Moreover, this convergence rate depends linearly on the number of components to be updated. On smooth strongly convex problems we prove that our methods converge linearly. We also focus on how to choose the probabilities to make our randomized algorithms converge as fast as possible, which leads us to solving a sparse semidefinite program. We then describe several applications that fit in our framework, in particular the convex feasibility problem. Finally, numerical experiments illustrate the behaviour.
of our methods, showing in particular that updating more than two components in parallel accelerates the method.

**Keywords** Convex optimization over networks · Linear coupled constraints · Random coordinate descent · Distributed computations · Convergence analysis

**Mathematics Subject Classification** 90C06 · 90C25 · 90C35

1 Introduction

The performance of a network composed of interconnected subsystems can be improved if the traditionally separated subsystems are optimized together. Recently, optimization methods have emerged as a powerful tool for solving some large-scale network problems, e.g., in resource allocation [1,2], coordination in multi-agent systems [1,3,4], image processing [5–7] and other areas [8–10]. The problems we consider in this paper have the following features: *the size of data is big* so that computation of the whole gradient required by standard methods becomes prohibitive. Moreover, *the incomplete structure of information* (e.g., the data are distributed over the nodes of the network, so that at a given time one can only work with the subset of the data that is available) may also be an obstacle for whole gradient computations. Finally, *their constraints are coupled*, meaning that problems cannot be solved individually at each node of the network, but that information must be shared between them in order to satisfy some global constraints. An appropriate way to approach these types of problems is through coordinate descent methods. These methods were among the first optimization methods studied in the literature, but had not received much attention until recently.

One of the main differences between variants of coordinate descent methods consists in the criterion used at each iteration to choose the coordinate over which the objective function is minimized (as well as the computational complexity of this choice). Two classical choices for these algorithms are the cyclic and the greedy coordinate search, which significantly differ by the amount of computation required to choose the appropriate index. For cyclic coordinate search, estimates for the rate of convergence were given recently in [11], while for the greedy coordinate search (e.g., Gauss-Southwell rule) the convergence rate is given in [12].

Another interesting approach is based on random choices of the coordinate. Recent complexity results on random coordinate descent methods for smooth convex objective functions were obtained in [1,13]. The extension to structured composite objective functions was given, e.g., in [14,15]. These methods are inherently serial. Recently, parallel and distributed implementations of coordinate descent methods were analyzed, e.g., in [8,10,16]. However, most of the aforementioned coordinate descent methods assume essentially unconstrained problems, which at best allow separable constraints. In contrast, in this paper we consider coordinate descent methods for optimization problems with linear coupled constraints. Greedy-type coordinate descent methods for general linearly constrained optimization problems are investigated in [12], while in [17] a two-coordinate greedy descent method is developed for minimizing a smooth
function subject to a single linear equality constraint and additional bound constraints on the decision variables. In both papers, detailed convergence analysis is provided for both the convex and non-convex settings. Random coordinate descent methods for optimization problems with linear coupled constraints have also been recently proposed in [1,14,18,19] (most of these papers being motivated by the work we present here, which was first made public in a technical report from 2011, see [20]). Another strand of this literature develops and analyses center-free gradient methods [2], augmented Lagrangian-based methods [21] or Newton methods [22].

Contributions: In this paper we develop random (block) coordinate descent methods with fixed probability distributions for choosing the (block of) coordinates, suited for large-scale optimization problems over networks, i.e., where information is distributed over nodes (corresponding to blocks of coordinates) and cannot be gather centrally. Moreover, this work is focused on optimization problems with linear coupled constraints, i.e., with a set of linear constraints that couples the different nodes. Due to this coupling we introduce a (block) variant of random coordinate descent method that updates at each iteration two or more (block) components of the solution vector (i.e., that involves two or more nodes of the network). This update involves the closed form solution of an optimization problem depending only on the chosen (blocks of) variables, while keeping all the other variables fixed. An important benefit of our algorithms is that they can be organized to exploit modern computational architectures, e.g., distributed and parallel computers.

Our approach allows us to analyze several methods within the same framework: full gradient (when all variables are updated at each iteration, implying full coordination of all nodes), serial random coordinate descent (where exactly two components are updated at each iteration) and any kind of parallel random coordinate descent method that updates a number of components in between those two extremes. For these methods we obtain a sublinear convergence rate for the expected value of a smooth objective function that depends linearly on the number of components to be updated in parallel. Thus, the theoretical speedup in terms of the number of iterations needed to approximately solve the problem, as compared to the basic serial method corresponding to two updated components, is an expression depending on the number of components to be updated (which can also correspond to nodes, processors or agents). For a complete network, the speedup is equal to the number of updated components. This result also shows that the speedup achieved by our methods on the class of separable problems with coupling constraints is the same as that for separable problems without coupling constraints.

For smooth strongly convex functions we prove that the new methods converge linearly. We also focus on how to choose the fixed probabilities used to select the (blocks of) coordinates to be updated at each iteration. This leads us to solving a sparse semidefinite programming problem (SDP) in order to compute probabilities making convergence of our randomized algorithms as fast as possible.

Contents: The paper is organized as follows. In Sect. 2 we introduce our optimization model and assumptions. In Sect. 3 we propose random block coordinate descent algorithms and derive in 4 their convergence rates in expectation. Section 5 studies the optimal choice of the probability distribution used to select coordinates. We describe
possible applications in Sect. 6 and conclude with a discussion of previous work and some preliminary numerical results in Sect. 7.

2 Problem Formulation

We work in the space $\mathbb{R}^N$ composed by column vectors. For $x, y \in \mathbb{R}^N$ denote the standard Euclidean inner product $\langle x, y \rangle = x^T y$ and the Euclidean norm $\|x\| = \langle x, x \rangle^{1/2}$. For symmetric matrices $X, Y$ we consider the inner product $\langle X, Y \rangle = \text{trace}(XY)$. We use the same notation $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ for spaces of different dimensions. We define the partition of the identity matrix as: $I_N = [e_1 \ldots e_N]$, where $e_i \in \mathbb{R}^N$. Then, for any $x \in \mathbb{R}^N$ we write $x = \sum_i x_i e_i$. Moreover, $D_x$ denotes the diagonal matrix with the entries $x$ on the diagonal and $x^p = [x_1^p \ldots x_N^p]^T$ for any integer $p$. We denote with $e \in \mathbb{R}^N$ the vector with all entries equal to one. For a positive semidefinite matrix $W \in \mathbb{R}^{N \times N}$ we consider the following order on its eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_N$ and define $\|x\|^2_W = x^T W x$ (which is not necessary a norm). The projection of a given vector $x \in \mathbb{R}^n$ onto a closed convex set $Q \subseteq \mathbb{R}^n$ is denoted by $[x]_Q$.

We consider large-scale network optimization problems where each agent in the network is associated with a local variable $x_j$. A linear coupling constraint states that their sum is fixed, and we minimize a separable convex objective:

$$f^* = \min_{x \in \mathbb{R}^N} f(x) = f_1(x_1) + \ldots + f_N(x_N) \; \text{s.t.} \; x_1 + \ldots + x_N = 0. \tag{1}$$

For convenience, we will focus on scalar convex functions $f_i : \mathbb{R} \rightarrow \mathbb{R}$, i.e., $x_i \in \mathbb{R}$. However, our results can be easily extended to the block case, when $x_i \in \mathbb{R}^n$, with $n \geq 1$, using the Kronecker product with the identity matrix $I_n$ in all our derivations. Moreover, constraints of the form $\alpha_1 x_1 + \ldots + \alpha_N x_N = b$, where $x_i \in \mathbb{R}^n$ and $\alpha_i \in \mathbb{R}$ can be easily handled in our framework by a change of coordinates. Finally, non-separable objective functions $f(x_1, \ldots, x_N)$ can be also treated in our framework, see [1,14] for more details. Optimization problems with linear coupled constraints (1) arise in many areas such as resource allocation [1,2], coordination in multi-agent systems [1,3,4], image processing [5–7,16] and other areas [8–10].

For the convex problem (1) we associate a network that can exchange information according to a communication graph $G = ([N], E)$, where we define the set of nodes $[N] = \{1, \ldots, N\}$ and the set of edges $E \subseteq [N] \times [N]$, i.e., $(i, j) \in E$ models that nodes $i$ and $j$ can exchange information. We assume that the graph $G$ is undirected and connected. For an integer $\tau \geq 2$, we also denote with $\mathcal{P}_\tau \subset 2^{[N]}$ the set of paths of $\tau$ vertices in the graph $G$. We consider a path in a graph to be a sequence of edges connecting a sequence of vertices, which are all distinct from one another. Note that we have at most $\binom{N}{\tau}$ paths of $\tau$ vertices in a graph. For $\tau = 2$, $\mathcal{P}_2$ represents the set of edges $E$ of the graph $G$. The local information structure imposed by the graph $G$ should be considered as part of the problem formulation (1).

Our goal is to devise a distributed algorithm that iteratively solves the convex problem (1) by passing estimates of the optimizer only between neighboring nodes along paths of $\tau$ vertices. There is great interest in designing such distributed and
parallel algorithms, since centralized algorithms scale poorly with the number of nodes and are less resilient to failure of the central node.

Let us define the extended subspace $S \subseteq \mathbb{R}^N$ and its orthogonal complement, denoted with $T \subseteq \mathbb{R}^N$, as:

$$S = \left\{ x \in \mathbb{R}^N : \sum_{i=1}^N x_i = 0 \right\} \quad \text{and} \quad T = \{ u \in \mathbb{R}^N : u_1 = \cdots = u_N \} = \text{span}\{ e \}.$$ 

The basic assumption considered in this paper is the following:

**Assumption 2.1** Each function $f_i$ is convex, differentiable and has Lipschitz continuous gradient with constant $L_i > 0$ on $\mathbb{R}$, i.e., the following holds:

$$\| \nabla f_i(x_i) - \nabla f_i(y_i) \| \leq L_i \| x_i - y_i \| \quad \forall x_i, y_i \in \mathbb{R}, \quad i \in [N]. \quad (2)$$

From this property of the gradient (2), the following inequality holds [23]:

$$f_i(x_i + s_i) \leq f_i(x_i) + \langle \nabla f_i(x_i), s_i \rangle + \frac{L_i}{2} \| s_i \|^2 \quad \forall x_i, s_i \in \mathbb{R}. \quad (3)$$

We denote with $X^*$ the set of optimal solutions for problem (1). Note that $x^*$ is an optimal solution for (1) if and only if it satisfies:

$$\sum_{i=1}^N x_i^* = 0, \quad \nabla f_i(x_i^*) = \nabla f_j(x_j^*) \quad \forall i \neq j \in [N].$$

### 3 Random Block Coordinate Descent Algorithms

In this section we devise randomized block coordinate feasible descent algorithms for solving the convex problem (1) and analyze their convergence. Since we have coupled constraints in the problem, the algorithm cannot update a single variable while keeping feasibility and has to update in parallel $\tau \geq 2$ components per iteration. Usually, the algorithm can be accelerated by parallelization, i.e., by using more than one pair of coordinates per iteration, as we will show in Sect.4.3. Our approach allows us to analyze in the same framework several methods: full gradient ($\tau = N$), serial random 2-coordinate descent ($\tau = 2$) and any parallel random coordinate descent method in between ($2 < \tau < N$). Let us fix $2 \leq \tau \leq N$ and denote with $\mathcal{N} \in \mathcal{P}_\tau$ a path of $\tau$ vertices in the connected undirected graph $\mathcal{G}$ ($\mathcal{N}$ is an edge when $\tau = 2$). We also assume availability of a probability distribution $p_\mathcal{N}$ over the set $\mathcal{P}_\tau$ of paths $\mathcal{N}$ of $\tau$ vertices in the graph $\mathcal{G}$. Then, we can derive a randomized $\tau$-coordinate descent algorithm where at each iteration we only update $\tau$ coordinates in the vector $x \in S$. Let us define $\mathcal{N} = (i_1, \ldots, i_\tau) \in \mathcal{P}_\tau$, with $i_j \in [N], s_{\mathcal{N}} = [s_{i_1} \cdots s_{i_\tau}]^T \in \mathbb{R}^\tau$, $L_{\mathcal{N}} = [L_{i_1} \cdots L_{i_\tau}]^T \in \mathbb{R}^\tau$ and $\nabla f_{\mathcal{N}}(x) = [\nabla f_{i_1}(x_{i_1}) \cdots \nabla f_{i_\tau}(x_{i_\tau})]^T \in \mathbb{R}^\tau$. With these notations, the following inequality holds from Assumption 2.1:
\[
    f(x + \sum_{i \in N} s_i e_i) \leq f(x) + \langle \nabla f_N(x), s_N \rangle + \frac{1}{2}\|s_N\|_{D_L N}^2. \tag{4}
\]

We can now devise a general randomized \(\tau\)-coordinate descent algorithm for problem (1) based on the inequality (4). Indeed, given an iterate \(x\) in the feasible set \(S\), we decide to randomly select the coordinate \(\tau\)-tuple \(N \in \mathcal{P}_\tau\) with probability \(p_N\). Let the next iterate be chosen as follows:

\[
x^+ = x + \sum_{i \in N} d_i e_i,
\]

i.e., we update \(\tau\) components in the vector \(x\), where the direction \(d_N\) is determined by requiring that the next iterate \(x^+\) to be also feasible for (1) and minimizing the right-hand side in (4), i.e.,:

\[
d_N = \arg \min_{s_N: \sum_{i \in N} s_i = 0} f(x) + \langle \nabla f_N(x), s_N \rangle + \frac{1}{2}\|s_N\|_{D_L N}^2
\]
or explicitly, in closed form:

\[
d_i = \frac{1}{L_i} \sum_{j \in N} \frac{1}{L_j} (\nabla f_j(x_j) - \nabla f_i(x_i)) \quad \forall i \in N.
\]

In conclusion, for a given probability distribution \((p_N)_{N \in \mathcal{P}_\tau}\) and a feasible starting point \(x^0 \in S\), we design the following random \(\tau\) (block) coordinate descent method, referred to in the sequel as RCD\(_\tau\):

---

**Algorithm RCD\(_\tau\) (2 \(\leq \tau \leq N\))**

For \(k = 0, 1, \ldots\), do:

1. choose \(\tau\)-tuple \(N_k = (i_1^k, \ldots, i_\tau^k)\) according to probabilities \((p_N)_{N \in \mathcal{P}_\tau}\)
2. set \(x_i^{k+1} = x_i^k \forall i \notin N_k\) and \(x_i^{k+1} = x_i^k + d_i^k \forall i \in N_k\).

Note that our scheme is of the block coordinate gradient descent type, since its updates are based on partial gradient information, but for simplicity we omit the term gradient as it is done in most of the literature [10, 13]. The random choice of coordinates makes algorithm RCD\(_\tau\) adequate for parallel and distributed implementations, and thus more flexible than greedy coordinate descent methods [12, 17]. Indeed, algorithm RCD\(_\tau\) is parallel and distributed, i.e., computations can be performed in parallel on the nodes of the graph and using local information, since only neighboring nodes along a path in the graph need to communicate with each other at each iteration. Further, only \(\tau\) components of \(x\) are updated at each iteration, so that our method has low complexity per iteration. More specifically, the complexity per iteration of RCD\(_\tau\) is \(O(\tau n_f + \tau)\), where \(n_f\) is the average cost of computing the gradient of each function.
fi, and τ is the cost of updating x+. Finally, observe that we maintain feasibility at each iteration of algorithm RCDτ, i.e., x_1^k + \ldots + x_N^k = 0 for all k ≥ 0.

Algorithm RCDτ belongs to the class of center-free methods, e.g., in [2] the term center-free refers to the absence of a coordinator in a distributed algorithm over a graph, which for appropriate weights w_{ij} ∈ \mathbb{R} and index sets N(i) ⊆ [N] perform the following general iteration:

\[ x_{i}^{k+1} = x_{i}^{k} + \sum_{j \in N(i)} w_{ij} \left( \nabla f_{j}(x_{j}^{k}) - \nabla f_{i}(x_{i}^{k}) \right) \quad \forall i \in [N]. \quad (5) \]

Indeed, in our case the weights are either \( w_{ij} = \frac{1}{L_{i} L_{j}} \) for \( i, j \in N_{k} \), or \( w_{ij} = 0 \) otherwise, and \( N(i) = N_{k} \). Moreover, for \( \tau = 2 \), our scheme chooses one pair of connected nodes, i.e., an edge of the graph, \((i_k, j_k) \in E\), according to some given probabilities \((p_{ij})_{(i,j) \in E}\) over the set \( E \) and performs the following basic (serial) iteration:

\[ x_{i}^{k+1} = x_{i}^{k} + \frac{1}{L_{i_k} + L_{j_k}} (e_{i_k} - e_{j_k}) \left( \nabla f_{j}(x_{j}^{k}) - \nabla f_{i}(x_{i}^{k}) \right). \]

Optimality conditions for the subproblem corresponding to \( d_{N} \) imply that there exists some real scalar \( \mu \) such that \( \nabla f_{N}(x) + D_{L_{N}} d_{N} + \mu e = 0 \) and \( \sum_{i \in N} d_i = 0 \). Based on that and inequality (4), the following decrease in the objective function values can be immediately derived for RCDτ:

\[
f(x^+) \leq f(x) + \langle \nabla f_{N}(x), d_{N} \rangle + \frac{1}{2} \|d_{N}\|_{D_{L_{N}}}^2 \]
\[
= f(x) - \langle D_{L_{N}} d_{N} + \mu e, d_{N} \rangle + \frac{1}{2} \|d_{N}\|_{D_{L_{N}}}^2 \]
\[
= f(x) - \sum_{i \in N} \frac{1}{2} \left( \nabla f_{i}(x_{i}) \right)^{2} \]
\[
= f(x) - \frac{1}{2} \nabla f(x)^{T} G_{N} \nabla f(x),
\]

where the matrix \( G_{N} \) is defined as follows:

\[
G_{N} = D_{L_{N}}^{-1} - \frac{1}{\sum_{i \in N} 1/L_{i}} L_{N}^{-1} (L_{N}^{-1})^{T}, \quad (6)
\]

where, with some abuse of notation, \( L_{N} \in \mathbb{R}^{N} \) denotes the vector with components zero outside the index set \( N \) and components \( L_{i} \) for \( i \in N \). Moreover, the symmetric matrix \( G_{N} \) is positive semidefinite, since according to its definition it satisfies

\[
x^{T} G_{N} x = \sum_{i \in N} \frac{\left( \sum_{j \in N} \frac{1}{L_{j}} (x_{j} - x_{i}) \right)^{2}}{2L_{i} \left( \sum_{j \in N} \frac{1}{L_{j}} \right)^{2}} \geq 0 \text{ for all } x \in \mathbb{R}^{N}.
\]

Therefore, given \( x \), taking
the conditional expectation of the function values over the random \( \tau \)-tuple \( N \in \mathcal{P}_\tau \), we obtain the following inequality:

\[
E[f(x^+ \mid x)] \leq f(x) - \frac{1}{2} \nabla f(x)^T G_\tau \nabla f(x),
\]

where \( G_\tau = \sum_{N \in \mathcal{P}_\tau} p_N G_N \) and can be interpreted as a weighted Laplacian for the graph \( G \). Since the matrix \( G_N \) is positive semidefinite, matrix \( G_\tau \) is also positive semidefinite; moreover, it has an eigenvalue \( \lambda_1(G_\tau) = 0 \) with the corresponding eigenvector \( e \in T \). Furthermore, since the graph \( G \) is connected, it also follows that the eigenvalue \( \lambda_1(G_\tau) = 0 \) is simple, i.e., \( \lambda_2(G_\tau) > 0 \).

We now define a norm that will be used subsequently for measuring distances in the extended subspace \( S \). We define the primal norm induced by the positive semidefinite matrix \( G_\tau \) as:

\[
\|u\|_{G_\tau} = \sqrt{u^T G_\tau u} \quad \forall u \in \mathbb{R}^N.
\]

Note that \( \|u\|_{G_\tau} = 0 \) for all \( u \in T \) and \( \|u\|_{G_\tau} > 0 \) for all \( u \in \mathbb{R}^N \setminus T \). On the subspace \( S \) we introduce the extended dual norm:

\[
\|x\|_{G_\tau}^* = \max_{u \in \mathbb{R}^N : \|u\|_{G_\tau} \leq 1} \langle x, u \rangle \quad \forall x \in S.
\]

Using the definition of conjugate norms, the Cauchy–Schwartz inequality holds:

\[
\langle u, x \rangle \leq \|u\|_{G_\tau} \cdot \|x\|_{G_\tau}^* \quad \forall x \in S, \ u \in \mathbb{R}^N.
\]

Let us define the average value: \( \hat{u} = \frac{1}{N} \sum_{i=1}^{N} u_i \). Then, the dual norm can be computed for any \( x \in S \) as follows:

\[
\|x\|_{G_\tau}^* = \max_{u \in \mathbb{R}^N : \langle G_\tau u, u \rangle \leq 1} \langle x, u \rangle = \max_{u : \langle G_\tau (u - \hat{u} e), u - \hat{u} e \rangle \leq 1} \langle x, u - \hat{u} e \rangle
\]

\[
= \max_{u : \langle G_\tau u, u \rangle \leq 1, \sum_{i=1}^{N} u_i = 0} \langle x, u \rangle = \max_{u : \langle G_\tau u, u \rangle \leq 1, e^T u = 0} \langle x, u \rangle
\]

\[
= \max_{u : \langle G_\tau u, u \rangle \leq 1} \langle x, u \rangle
\]

\[
= \min_{\nu, \mu \geq 0} \max_{u \in \mathbb{R}^N} [\langle x, u \rangle + \mu (1 - \langle G_\tau u, u \rangle) - \nu \langle e e^T u, u \rangle]
\]

\[
= \min_{\nu, \mu \geq 0} \mu + \langle (\mu G_\tau + \nu e e^T)^{-1} x, x \rangle
\]

\[
= \min_{\nu \geq 0} \min_{\mu \geq 0} \frac{1}{\mu} \langle (G_\tau + \nu e e^T)^{-1} x, x \rangle
\]

\[
= \min_{\xi \geq 0} \sqrt{\langle (G_\tau + \xi e e^T)^{-1} x, x \rangle}.
\]
We obtain an extended dual norm that is well defined on subspace $S$:

$$
\|x\|^*_G = \min_{\zeta \geq 0} \sqrt{\langle (G_\tau + \zeta ee^T)^{-1} x, x \rangle} \quad \forall x \in S.
$$

(8)

The eigenvalue decomposition of the positive semidefinite matrix $G_\tau$ can be written as $G_\tau = \Xi \operatorname{diag}(0, \lambda_2, \ldots, \lambda_N) \Xi^T$, where $\lambda_i$ are its positive eigenvalues, and $\Xi = [e \xi_2 \cdots \xi_N]$ satisfies $\langle e, \xi_i \rangle = 0$ for all $i$. Then, we have:

$$(G_\tau + \zeta ee^T)^{-1} = \Xi \operatorname{diag}(\zeta \|e\|^2, \lambda_2, \ldots, \lambda_N)^{-1} \Xi^T.$$  

From (8) it follows that our newly defined norm has the following closed form:

$$
\|x\|^*_G = \sqrt{x^T G_\tau^+ x} \quad \forall x \in S,
$$

(9)

where $G_\tau^+ = \Xi \operatorname{diag}(0, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_N}) \Xi^T$ denotes the pseudoinverse of matrix $G_\tau$.

We can also derive another closed form expression for the norm $\|\cdot\|^*_G$ on the subspace $S$. For example, if we define $G_\tau^{[N-1]}$ the leading matrix of dimension $N - 1$ of $G_\tau$ and $x^{[N-1]} = [x_1 \cdots x_{N-1}]^T \in \mathbb{R}^{N-1}$, then from the definition of the norm $\|\cdot\|^*_G$ on $S$ we have:

$$
\|x\|^*_G = \max_{u \in \mathbb{R}^N : \langle G_\tau u, u \rangle \leq 1} \langle x, u \rangle = \max_{u \in \mathbb{R}^N : \langle G_\tau (u-u_N e), u-u_N e \rangle \leq 1} \langle x, u-u_N e \rangle
$$

$$
= \max_{u \in \mathbb{R}^{N-1} : \langle G_\tau^{[N-1]} u, u \rangle \leq 1} \langle x^{[N-1]}, u \rangle.
$$

The optimality conditions in the previous maximization problem are given by:

$$
\exists \mu \geq 0 \text{ s.t. } \mu G_\tau^{[N-1]} u = x^{[N-1]} \quad \text{and} \quad \langle G_\tau^{[N-1]} u, u \rangle \leq 1.
$$

However, for a connected graph, the zero eigenvalue of $G_\tau$ is simple, and thus, it is easy to show that the leading matrix of dimension $N - 1$ of $G_\tau$, that is $G_\tau^{[N-1]}$, is always invertible [24]. Then, after simple manipulations of the previous optimality conditions, we also get the following closed form expression for the norm $\|\cdot\|^*_G$ on the subspace $S$:

$$
\|x\|^*_G = \sqrt{x_{[N-1]}^T (G_\tau^{[N-1]})^{-1} x_{[N-1]}} \quad \forall x \in S.
$$

(10)

4 Convergence Rates of Algorithm RCD$_\tau$

We consider separately the cases of a smooth convex objective function (satisfying Assumption 2.1) and of a smooth strongly convex objective function.
4.1 Convergence Rate: Smooth Case

In order to estimate the rate of convergence of our algorithm, we introduce the following distance that takes into account that our algorithm is a descent method:

$$\mathcal{R}(x^0) = \max_{\{x \in S : f(x) \leq f(x^0)\}} \min_{x^* \in X^*} \|x - x^*\|_{G,\tau},$$

which measures the size of the sublevel set of $f$ given by $x^0$. We assume that this distance is finite for the initial iterate $x^0$. After $k$ iterations of the algorithm, we generate a random output $(x^k, f(x^k))$, which depends on the observed implementation of random variable:

$$\eta^k = (N_0, \ldots, N_k).$$

Let us define the expected value of the objective function w.r.t. $\eta^k$:

$$\phi_k = E[f(x^k)].$$

We now prove the main result of this section, namely sublinear convergence in expected value of the objective function for the smooth convex case:

**Theorem 4.1** Let Assumption 2.1 holds for optimization problem (1) and let the sequence $(x^k)_{k \geq 0}$ be generated by algorithm RCD$\tau$. Then, we have the following sublinear convergence rate for expected value of the objective function:

$$\phi_k - f^* \leq \frac{2 \mathcal{R}^2(x^0)}{k}. \quad (11)$$

**Proof** Recall that all our iterates are feasible, i.e., $x^k \in S$. From convexity of $f$ and the definition of the norm $\| \cdot \|_{G,\tau}$ on the subspace $S$, we get:

$$f(x^l) - f^* \leq \langle \nabla f(x^l), x^l - x^* \rangle \leq \|x^l - x^*\|_{G,\tau}^* \|\nabla f(x^l)\|_{G,\tau} \leq \mathcal{R}(x^0) \cdot \|\nabla f(x^l)\|_{G,\tau} \quad \forall l \geq 0.$$

Combining this inequality with (7), we obtain:

$$f(x^l) - E[f(x^{l+1}) | x^l] \geq \frac{(f(x^l) - f^*)^2}{2 \mathcal{R}^2(x^0)},$$

or equivalently

$$E[f(x^{l+1}) | x^l] - f^* \leq f(x^l) - f^* - \frac{(f(x^l) - f^*)^2}{2 \mathcal{R}^2(x^0)}.$$
Taking the expectation of both sides of this inequality in $\eta_{l-1}$ and denoting $\Delta_l = \phi_l - f^*$ leads to:

$$\Delta_{l+1} \leq \Delta_l - \frac{\Delta_l^2}{2R^2(x^0)}.$$ 

Dividing both sides of this inequality with $\Delta_l \Delta_{l+1}$ and taking into account that $\Delta_{l+1} \leq \Delta_l$ (see 7), we obtain:

$$\frac{1}{\Delta_l} \leq \frac{1}{\Delta_{l+1}} - \frac{1}{2R^2(x^0)} \quad \forall l \geq 0.$$ 

Adding these inequalities from $l = 0, \ldots, k-1$ we get the following inequalities

$$0 \leq \frac{1}{\Delta_0} \leq \frac{1}{\Delta_k} - \frac{k}{2R^2(x^0)},$$

from which we obtain the desired statement (11). 

Algorithm RCD$_{\tau}$ features a low complexity per iteration (e.g., $O(\tau)$ basic operations provided that the cost of computing the gradient of each function $f_i$ is insignificant), and Theorem 4.1 shows that it has sublinear rate of convergence in expected value of the objective function when applied to the smooth convex problem (1). Note that the cost of randomly choosing a $\tau$-tuple of indices $N$ for a given probability distribution is negligible (e.g., for $\tau = 2$ the cost is $\ln N$ [13]). Moreover, the convergence rate (11) depends on the number of coordinates $\tau$ to be updated via the term $R(x^0)$, whose definition is based on matrix $G_{\tau}$. In Sect. 4.3 we show how the number of updated coordinates $\tau$ enters explicitly into the convergence rate (11).

4.2 Convergence Rate: Strongly Convex Case

In addition to the smoothness assumption (2.1), we now assume that the function $f$ is strongly convex with respect to the extended norm $\| \cdot \|_{G_{\tau}}^*$ with strong convexity parameter $\sigma_{G_{\tau}} > 0$ on the subspace $S$. More precisely, we assume that the objective function $f$ satisfies the following inequality:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \sigma_{G_{\tau}} \frac{1}{2} (\| x - y \|_{G_{\tau}}^*)^2 \quad \forall x, y \in S. \quad (12)$$

We now derive linear convergence estimates for algorithm RCD$_{\tau}$ under this additional strong convexity assumption:

**Theorem 4.2** Let Assumption 2.1 holds and assume $f$ to be a $\sigma_{G_{\tau}}$-strongly convex function with respect to norm $\| \cdot \|_{G_{\tau}}^*$ on $S$ (see (12)). Then, the sequence $(x^k)_{k \geq 0}$ generated by algorithm RCD$_{\tau}$ satisfies the following linear convergence rate for the expected value of the objective function:

$$\phi_k - f^* \leq (1 - \sigma_{G_{\tau}})^k \left( f(x^0) - f^* \right). \quad (13)$$
Proof. Given $x^k$, taking the conditional expectation in (7) over the random $\tau$-tuple $\mathcal{N}_k$ leads to the following inequality:

$$2 \left( f(x^k) - E \left[ f(x^{k+1}) \mid x^k \right] \right) \geq \| \nabla f(x^k) \|_{G^r}^2.$$ 

On the other hand, consider the minimization of the right-hand side in (12) over $x \in S$, and denote $x(y)$ its optimal solution. Using the definition of the dual norm $\| \cdot \|_{G^r}$ in the subspace $S$ as defined in (9), one can see that $x(y)$ satisfies the following optimality conditions:

$$\exists \mu \in \mathbb{R} \text{ s.t.: } \nabla f(y) + \sigma G^r (x(y) - y) + \mu e = 0 \text{ and } x(y) \in S.$$ 

Combining these optimality conditions with the well-known property of the pseudoinverse, that is $G^r G^r = G^r$, we get that the optimal value of this minimization problem has the following expression:

$$f(y) - \frac{1}{2\sigma} \| \nabla f(y) \|_{G^r}^2.$$ 

Therefore, minimizing both sides of inequality (12) over $x \in S$, we have:

$$\| \nabla f(y) \|_{G^r}^2 \geq 2\sigma(f(y) - f^*) \text{ \forall } y \in S$$

and for $y = x^k$ we get:

$$\| \nabla f(x^k) \|_{G^r}^2 \geq 2\sigma \left( f(x^k) - f^* \right).$$

Combining the first inequality with the last one, and taking expectation in $\eta_k^{-1}$ in both sides, we arrive at the statement of the theorem. 

If functions $f_i$ are strongly convex with respect to the Euclidean norm with convexity parameters $\sigma_i$, i.e.,

$$f_i(x_i) \geq f_i(y_i) + \langle \nabla f_i(y_i), x_i - y_i \rangle + \frac{\sigma_i}{2} |x_i - y_i|^2 \text{ \forall } x_i, y_i, \ i \in [N],$$

then the whole function $f = \sum_i f_i$ is also strongly convex w.r.t. the extended norm induced by the positive definite matrix $D_\sigma$, where $\sigma = [\sigma_1 \cdots \sigma_N]^T$, i.e.,

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} \| x - y \|_{D_\sigma}^2 \text{ \forall } x, y.$$ 

Note that in this extended norm $\| \cdot \|_{D_\sigma}$ the strong convexity parameter of the function $f$ is equal to 1. It follows immediately that the function $f$ is also strongly convex with
respect to the norm \( \| \cdot \|_{\tau}^* \), with a strong convexity parameter \( \sigma_{G_{\tau}} \) satisfying, for some \( \zeta \geq 0 \), the following inequality:

\[
\sigma_{G_{\tau}} D_{\sigma}^{-1} \leq G_{\tau} + \zeta e e^T \quad \Leftrightarrow \quad \sigma_{G_{\tau}} G_{\tau}^+ \leq D_{\sigma}.
\]

Thus, the strong convexity parameter \( \sigma_{G_{\tau}} \) satisfies the following linear matrix inequalities (LMIs) in \((\sigma_{G_{\tau}}, \zeta) \geq 0\):

\[
\sigma_{G_{\tau}} I_N \leq D_{\sigma}^{1/2} (G_{\tau} + \zeta e e^T) D_{\sigma}^{1/2} \quad \text{or} \quad \sigma_{G_{\tau}}^{-1} I_N \geq D_{\sigma}^{-1/2} G_{\tau}^+ D_{\sigma}^{-1/2}.
\]

(14)

Finally, we should notice that we could also easily derive results showing that the problem is approximately solved with high probability in both situations (with/without strong convexity), see, e.g., [1,13–15] for more details.

### 4.3 How Number of Updated Coordinates \( \tau \) Enters into Convergence Rates

Note that matrix \( G_{\tau} \) depends directly on the number of components \( \tau \) to be updated and, therefore, \( R(x^0) \) is also depending on \( \tau \). Moreover, the convergence rate can be explicitly expressed in terms of \( \tau \) for some specific choices of the probabilities and the graph \( G \). In particular, let us assume that \( G \) is a complete graph and that we know some constants \( R_i > 0 \) such that for any \( x \) satisfying \( f(x) \leq f(x^0) \) there exists an \( x^* \in X \) such that:

\[
|x_i - x^*_i| \leq R_i \quad \forall i \in [N],
\]

(15)

and recall that \( L = [L_1 \cdots L_N]^T \) and \( D_L \) is the diagonal matrix with entries on the diagonal given by the vector \( L \). Moreover, let us consider probabilities depending on the Lipschitz constants \( L_i \) for any path \( N^i \in P_{\tau} \) of \( \tau \) vertices in the complete graph \( G \), defined as:

\[
p_{N^i}^L = \frac{\sum_{i \in N^i} 1/L_i}{\sum_{N \in P_{\tau}} \sum_{i \in N} 1/L_i}.
\]

(16)

**Theorem 4.3** Under Assumption 2.1 and for the choice of probabilities (16) on a complete graph \( G \), algorithm RCD\( _{\tau} \) satisfies the following sublinear convergence rate in expected value of the objective function:

\[
\phi_k - f^* \leq \frac{N - 1}{\tau - 1} \cdot \frac{2 \sum_{i \in [N]} L_i R_i^2}{k}.
\]

(17)

**Proof** Using the definition of the indicator function \( 1_{N^i} \), we can see that:

\[
\Sigma_{\tau}^{-1} = \sum_{N \in P_{\tau}} \sum_{i \in N} \frac{1}{L_i} = \sum_{j=1}^{N} \sum_{i \in N_j} \frac{1}{L_i} = \sum_{j=1}^{N} \sum_{i=1}^{N_j} 1_{N_j}(i) \frac{1}{L_i}
\]

\( \Box \) Springer
\[ \sum_{i=1}^{N} \frac{1}{L_i} \left( \sum_{j=1}^{N} L_j^T \right) = \sum_{i=1}^{N} \frac{1}{L_i} \left( \sum_{j=1}^{\tau} L_j^T \right). \]

Thus, using \( G_{\tau} = \sum_{N \in \mathcal{N}} p_N G_N \) and the expression of \( G_N \) given in (6), we can derive that matrix \( G_{\tau} \) has the following expression:

\[
G_{\tau} = \frac{1}{\tau} \sum_{N \in \mathcal{P}_{\tau}} \left[ \sum_{i \in N} \frac{1}{L_i} D_N^{-1} - L_N^{-1} (L_N^{-1})^T \right]
\]

Using the expression of \( G_{\tau} \) in (10) and the matrix inversion lemma, we get:

\[
\left( G_{\tau}^{[N-1]} \right)^{-1} = \frac{N-1}{\tau-1} \left( D_L^{[N-1]} + L_N e_{[N-1]} e_{[N-1]}^T \right),
\]

where \( e_{[N-1]} \in \mathbb{R}^{N-1} \) is the vector with all entries 1. Thus, we obtain:

\[
\left( \|x\|_{G_{\tau}}^* \right)^2 = \frac{N-1}{\tau-1} \sum_{i \in [N]} L_i |x_i|^2 \quad \forall x \in S.
\]  

From the definition of \( R_i \) and the expression (18) for the norm on \( S \), we obtain:

\[
R^2(x^0) \leq \frac{N-1}{\tau-1} \sum_{i \in [N]} L_i R_i^2.
\]

Replacing this upper bound for \( R(x^0) \) in Theorem 4.1 we get the statement of the theorem.

Note that for \( \tau = N \) we recover the convergence rate of the full gradient method, while for \( \tau = 2 \) we get the convergence rate of the basic random coordinate descent method. Thus, the theoretical speedup of the parallel algorithm RCD_{\tau}, in terms of the number of iterations needed to approximately solve the problem, as compared to the basic random coordinate descent method RCD_{2}, is equal in this case to \( \tau \)—the number of components to be updated (or number of processors available). This result also shows that the speedup achieved by our method on the class of separable problems.
with linear coupled constraints is the same as for separable problems without coupled constraints.

We can also derive explicit convergence rates for other types of graphs. For example, our algorithm RCD is adequate for networks whose associated graph is of star shape type. In this case, we can take \( \tau = 2 \) and consider all the paths of length 2, i.e., \( \mathcal{P}_2 = \{(1, N), \ldots, (N - 1, N)\} \). If we define the vector of probabilities \( p \) and the vector \( g_{pL} \) as follows:

\[
p = [p_1N \cdots p_{N-1}N]^T \quad \text{and} \quad g_{pL} = \begin{bmatrix} p_1N \quad \cdots \quad p_{N-1}N \\ L_1 + L_N \quad \cdots \quad L_{N-1} + L_N \end{bmatrix},
\]

then the weighted Laplacian \( G_2 \) for the star graph has the following expression:

\[
G_2 = \begin{bmatrix} D_{g_{pL}} & g_{pL}^T \\ g_{pL} & \sum_{i=1}^{N-1} p_{iN}L_i \end{bmatrix}.
\]

Using the previous expression for \( G_2 \) in (10) and that \( G_2^{[N-1]} = D_{g_{pL}} \), we get:

\[
(x^*_{G_2})^2 = x^T_{[N-1]}D_{g_{pL}}^{-1}x_{[N-1]} = \sum_{i=1}^{N-1} \frac{L_i + L_N}{p_{iN}}|x_i|^2 \quad \forall x \in S.
\]

Now choosing, for simplicity, the uniform distribution \( p_{iN} = 1/(N - 1) \), we obtain from the definition of \( R_i \) and the expression (19) for the norm on \( S \):

\[
R^2(x^0) \leq (N - 1) \left( \sum_{i=1}^{N-1} L_i R_i^2 + \sum_{i=1}^{N-1} L_N R_i^2 \right),
\]

Replacing this upper bound for \( R(x^0) \) in Theorem 4.1, we get for algorithm RCD on star graphs with a uniform probability distribution the following sublinear convergence rate in the expected value of the objective function:

\[
\phi_k - f^* \leq \frac{N - 1}{\tau - 1} \cdot \frac{2 \sum_{i=1}^{N-1} L_i R_i^2 + 2 \sum_{i=1}^{N-1} L_N R_i^2}{k} \bigg|_{\tau = 2},
\]

expressed in this way in order to show again the dependence on \( \tau = 2 \). Comparing the convergence rate for a complete graph (17) with that for a star graph (20), we observe that the first one is better provided that \( R_N^2 \leq \sum_{i=1}^{N-1} R_i^2 \) (see also the discussion from Sect. 4). Similar convergence rates can be also derived for, e.g., the cycle graph. Due to space limitations we omit these derivations here, which can be found in [20].

For the strongly convex case, we note that combining the Lipschitz inequality (3) with the strong convex inequality (12) we get:

\[
\sum_{i \in [N]} L_i |x_i - y_i|^2 \geq \sigma_{G_r} \left( \|x - y\|_{G_r}^* \right)^2 \quad \forall x, y \in S.
\]
Now, if we consider, e.g., a complete graph and the probabilities given in (16) and use the expression for the norm \( \| \cdot \|_{G_{\tau}} \) given in (18), we obtain \( \sigma_{G_{\tau}} \leq \frac{\tau - 1}{N - 1} \). Thus, in the strongly convex case the linear convergence rate in expectation (13) can be also expressed in terms of \( \tau \).

5 Design of Optimal Probabilities

Probabilities \((p_{N})_{N \in \mathcal{P}_{\tau}}\) corresponding to paths of \( \tau \) vertices in the connected graph \( G \), which the randomized coordinate descent algorithm RCD\(_{\tau}\) depends on, can be chosen in several ways. For example, we can choose probabilities dependent on the Lipschitz constants \( L_{i} \) and some scalar parameter \( \alpha \):

\[
p_{N}^{\alpha} = \frac{\sum_{i \in N} L_{i}^{\alpha}}{\Sigma_{\tau}^{\alpha}} , \quad \Sigma_{\tau}^{\alpha} = \sum_{N \in \mathcal{P}_{\tau}} \sum_{i \in N} L_{i}^{\alpha} , \quad \alpha \in \mathbb{R}.
\]

(21)

Note that for \( \alpha = 0 \) we recover the uniform probabilities. Let us now try to compute optimal probabilities based on the convergence rate of the method. From the definition of the constants \( R_{i} \) given in (15) it follows that:

\[
\mathcal{R}(x^{0}) \leq \mathcal{R}(R) := \max_{x \in S, |x_{i}| \leq R_{i}} \| x \|_{G_{\tau}}^{*} , \quad \text{with} \quad G_{\tau} = \sum_{N \in \mathcal{P}_{\tau}} p_{N} G_{N},
\]

where \( R = [R_{1} \cdots R_{N}]^{T} \). We have the freedom to choose the matrix \( G_{\tau} \) that depends linearly on the probabilities \( p_{N} \). For the probabilities \((p_{N})_{N \in \mathcal{P}_{\tau}}\) corresponding to paths of \( \tau \) vertices in the connected graph \( G \) we define the following set of matrices:

\[
\mathcal{M} = \left\{ G_{\tau} : G_{\tau} = \sum_{N \in \mathcal{P}_{\tau}} p_{N} G_{N}, \quad G_{N} = D_{L_{N}}^{-1} - \frac{1}{\sum_{i \in N} 1/L_{i}} L_{N}^{-1} (L_{N}^{-1})^{T} \right\}.
\]

Therefore, we search for the probabilities \( p_{N} \) that are the optimal solution of the following optimization problem:

\[
\mathcal{R}^{*}(x^{0}) = \min_{p_{N}} \mathcal{R}(x^{0}) \leq \min_{p_{N}} \mathcal{R}(R) = \min_{G_{\tau} \in \mathcal{M}} \max_{x \in S, |x_{i}| \leq R_{i}} \| x \|_{G_{\tau}}^{*}.
\]

Let us define \( R = [R_{1} \cdots R_{N}]^{T} \) and \( v = [v_{1} \cdots v_{N}]^{T} \). In the next theorem we derive an easy to compute upper bound on \( \mathcal{R}^{*}(x^{0}) \) and provide a way to suboptimally select the probabilities \( p_{N} \):

**Theorem 5.1** Let Assumption 2.1 holds. Then, a suboptimal choice of probabilities \((p_{N})_{N \in \mathcal{P}_{\tau}}\) can be obtained as a solution of the following SDP, whose optimal value is an upper bound on \( \mathcal{R}^{*}(x^{0}) \), i.e.,

\[
(\mathcal{R}^{*}(x^{0}))^{2} \leq \min_{G_{\tau} \in \mathcal{M}, \xi \geq 0, v \geq 0} \left\{ \langle v, R^{2} \rangle : \begin{bmatrix} G_{\tau} + \xi ee^{T} & I_{N} \\ I_{N} & D_{v} \end{bmatrix} \succeq 0 \right\}.
\]

(22)
Proof Using the definition of \( \mathcal{R}(R) \) and of the norm \( \| \cdot \|_G^p \), we get:

\[
\min_{p_N} (\mathcal{R}(R))^2 \leq \min_{G \in \mathcal{M}} \max_{x \in S, |x_i| \geq R_i} \left( \|x\|_G^p \right)^2
\]

\[
= \min_{G \in \mathcal{M}} \max_{x \in S, |x_i| \geq R_i} \min_{\zeta \geq 0} ((G_\tau + \zeta ee^T)^{-1} x, x)
\]

\[
= \min_{G \in \mathcal{M}, \zeta \geq 0, \leq x \in S, |x_i| \geq R_i} \max_{x \in S, |x_i| \leq R_i} ((G_\tau + \zeta ee^T)^{-1} x, x)
\]

\[
= \min_{G \in \mathcal{M}, \zeta \geq 0} \max_{x \in S, |x_i| \leq R_i} ((G_\tau + \zeta ee^T)^{-1} x, x)
\]

\[
= \min_{G, \zeta \geq 0} \max_{x \in X} \langle (G_\tau + \zeta ee^T)^{-1}, x \rangle,
\]

where \( X = \{X : X \succeq 0, \text{ rank } X = 1, \langle ee^T, X \rangle = 0, \langle X, E_{ii} \rangle \leq R_i^2 \ \forall i \in [N] \} \) and \( E_{ii} = e_i e_i^T \). Using the standard SDP relaxation, we have:

\[
\min_{p_N} (\mathcal{R}(R))^2 \leq \min_{G \in \mathcal{M}, \zeta \geq 0} \max_{X \in X_r} \langle (G_\tau + \zeta ee^T)^{-1}, X \rangle,
\]

where \( X_r = \{X : X \succeq 0, \langle ee^T, X \rangle = 0, \langle X, E_{ii} \rangle \leq R_i^2 \ \forall i \in [N] \} \), i.e., we have removed the rank-one constraint. Then, we reformulate the right-hand side of the previous optimization problem using Lagrange multipliers as follows:

\[
\min_{G \in \mathcal{M}, \zeta \geq 0} \max_{X \in X_r} \langle (G_\tau + \zeta ee^T)^{-1}, X \rangle
\]

\[
= \min_{G \in \mathcal{M}, \zeta, \leq v \geq 0, \leq Z \leq 0, \theta \in \mathbb{R}} \max_{X \in \mathbb{R}^{N \times N}} [\langle (G_\tau + \zeta ee^T)^{-1} + Z + \theta ee^T, X \rangle
\]

\[
+ \sum_{i=1}^{N} v_i (R_i^2 - \langle X, E_{ii} \rangle)].
\]

where \( v = [v_1 \cdots v_N]^T \). Rearranging the terms, we can further write:

\[
\min_{G \in \mathcal{M}, \theta \in \mathbb{R}, Z \geq 0, \zeta, v \geq 0} \left[ \sum_{i} v_i R_i^2 + \max_{X \in \mathbb{R}^{N \times N}} \langle (G_\tau + \zeta ee^T)^{-1} + Z + \theta ee^T - \sum_{i} v_i E_{ii}, X \rangle \right]
\]

\[
= \min_{(G, \zeta, \leq v, \theta) \in \mathcal{F}} \sum_{i} v_i R_i^2,
\]

where the feasible set is described as: \( \mathcal{F} = \{(G_\tau, Z, \zeta, v, \theta) : G_\tau \in \mathcal{M}, \theta \in \mathbb{R}, Z \succeq 0, \zeta, v \geq 0, (G_\tau + \zeta ee^T)^{-1} + Z + \theta ee^T - \sum_{i} v_i E_{ii} = 0 \} \). Moreover, since \( Z \succeq 0 \), the feasible set can be rewritten as:

\[
\{(G_\tau, \zeta, v, \theta) : G_\tau \in \mathcal{M}, \theta \in \mathbb{R}, \zeta, v \geq 0, \sum_{i} v_i E_{ii} - (G_\tau + \zeta ee^T)^{-1} - \theta ee^T \succeq 0 \}.
\]
We observe that we can take $\theta = 0$ and then get the feasible set:

$$\{(G_{\tau}, \zeta, \nu) : G_{\tau} \in \mathcal{M}, \zeta, \nu \geq 0, G_{\tau} + \zeta ee^T \succeq D_{\tau}^{-1}\}.$$ 

Hence, we obtain the following SDP:

$$\min_{p_N} (R(\mathcal{R}))^2 \leq \min_{G_{\tau} \in \mathcal{M}, \zeta, \nu \geq 0, G_{\tau} + \zeta ee^T \succeq D_{\tau}^{-1}} (\nu, R^2).$$

Finally, the SDP (22) in the theorem is obtained from Schur complement formula applied to the previous optimization problem. \qed

An immediate consequence of Theorem 5.1 is the following upper bound on $R^2(x_0)$:

$$R^2(x_0) \leq \min_{\zeta \geq 0, \nu \geq 0} \left\{ (\nu, R^2) : \begin{bmatrix} G_{\tau} + \zeta ee^T & I_N \\ I_N & D_{\tau}^{-1} \end{bmatrix} \succeq 0 \right\} \quad \forall G_{\tau} \in \mathcal{M}. \quad (23)$$

Since we assume a connected graph $G$, we have that $\lambda_1(G_{\tau}) = 0$ is simple with the corresponding eigenvector $e$, and consequently $\lambda_2(G_{\tau}) > 0$. Then, the following equivalence holds:

$$G_{\tau} + t \frac{ee^T}{\|e\|^2} \succeq t I_N \quad \text{if and only if} \quad t \leq \lambda_2(G_{\tau}), \quad (24)$$

since the spectrum of the matrix $G_{\tau} + \zeta ee^T$ is $\{\zeta \|e\|^2, \lambda_2(G_{\tau}), \ldots, \lambda_N(G_{\tau})\}$. It follows that $\zeta = \frac{t}{\|e\|^2}$, $\nu_i = \frac{1}{t}$ for all $i$, and $G_{\tau}$ such that $t \leq \lambda_2(G_{\tau})$ is feasible for the SDP problem (23) We conclude that:

$$\left( R^2(x_0) \right)^2 \leq \min_{G_{\tau} \in \mathcal{M}, \zeta, \nu \geq 0, G_{\tau} + \zeta ee^T \succeq D_{\tau}^{-1}} \left( \nu, R^2 \right) \leq \min_{G_{\tau} \in \mathcal{M}, t \leq \lambda_2(G_{\tau})} \sum_{i=1}^{N} \frac{R_i^2}{t} \leq \frac{\sum_{i=1}^{N} R_i^2}{\lambda_2(G_{\tau})} \quad \forall G_{\tau} \in \mathcal{M}. \quad (25)$$

Then, according to Theorem 4.1 and (23), we obtain the following upper bound on the rate of convergence for the expected value of the objective function in the smooth convex case:

$$\phi_k - f^* \leq \frac{2 \sum_{i=1}^{N} R_i^2}{\lambda_2(G_{\tau})} \cdot k \quad \forall G_{\tau} \in \mathcal{M}. \quad (26)$$

The convergence rate estimate (26) allows us to compare the behavior of algorithm $RCD_{\tau}$ for different types of graphs. For example, the Laplacian $G$ of a connected graph with $N$ nodes has the second eigenvalue $\lambda_2(G)$ of order $O(N)$, $O(1)$ and $O(1/N^2)$ for complete graphs, star graphs and cycle graphs, respectively [24]. Therefore, as we expected, the convergence behavior of algorithm $RCD_{\tau}$ is the best for complete graphs and the worst for cycle graphs.

\copyright Springer
From the convergence rate for algorithm RCD\(_{\tau}\) given in (26) it follows that we can also choose the probabilities such that the second eigenvalue of the weighted Laplacian \(G_{\tau}\) is maximized:

\[
\max_{G_{\tau} \in \mathcal{M}} \lambda_2(G_{\tau}).
\]

In conclusion, in order to find some suboptimal probabilities \((p_N)_{N \in \mathcal{P}_{\tau}}\), we can solve the following SDP problem, simpler than the one given in (22):

**Corollary 5.1** From (24) we get for the smooth case that a suboptimal choice of probabilities \((p_N)_{N \in \mathcal{P}_{\tau}}\) can be obtained as a solution of the following SDP:

\[
p^*_N = \arg \max_{t, G_{\tau} \in \mathcal{M}} \left\{ t : G_{\tau} \succeq t \left( I_N - \frac{ee^T}{\|e\|^2} \right) \right\}.
\]

(27)

Note that the matrices on both sides of the LMI from (27) have the common eigenvalue zero associated with the eigenvector \(e\), so that this LMI has empty interior which can cause problems for some classes of interior point methods. We can overcome this problem by replacing the LMI constraint in (27) with the following equivalent LMI:

\[
G_{\tau} + \frac{ee^T}{\|e\|^2} \succeq t \left( I_N - \frac{ee^T}{\|e\|^2} \right).
\]

Finally, when the functions \(f_i\) are \(\sigma_i\)-strongly convex, it follows from Theorem 4.2 and the LMI (14) that we need to search for \(\sigma_{G_{\tau}}\) as large as possible in order to get the best convergence rate. Therefore, we get the following result:

**Corollary 5.2** For the smooth and strongly convex case suboptimal probabilities can be chosen as the solution of the following SDP:

\[
p^*_N = \arg \max_{\sigma_{G_{\tau}}, \zeta \geq 0, G_{\tau} \in \mathcal{M}} \left\{ \sigma_{G_{\tau}} : \sigma_{G_{\tau}} I_N \preceq D_{\sigma}^{1/2} \left( G_{\tau} + \zeta ee^T \right) D_{\sigma}^{1/2} \right\}.
\]

(28)

### 6 Applications

Problem (1) arises in many real-world applications, such as image processing [5–7], resource allocation [1,2] and coordination in multi-agent systems [1,4]. For example, we can interpret (1) as \(N\) agents exchanging \(n\) goods to minimize a total cost, where the constraint \(\sum_i x_i = 0\) is the equilibrium or market clearing constraint. In this context \([x_i]_j \geq 0\) means that agent \(i\) receives \([x_i]_j\) of good \(j\) from the exchange and \([x_i]_j < 0\) means that agent \(i\) contributes \(|(x_i)_j|\) of good \(j\) to the exchange. It can be also viewed as the distributed dynamic energy management problem: \(N\) devices exchange power in time periods \(t = 1, \ldots, n\), where \(x_i \in \mathbb{R}^n\) is the power flow profile for device \(i\), and \(f_i(x_i)\) is the cost of profile \(x_i\) (and usually encodes constraints). In this application the constraint \(\sum_i x_i = 0\) represents the energy balance (in each time period).
Problem (1) can also be seen as the dual corresponding to the minimization of a sum of convex functions. Consider the following primal convex optimization problem that arises in many engineering applications:

\[ g^* = \min_{v \in \cap_{i=1}^N Q_i} g_1(v) + \cdots + g_N(v), \]  

(29)

where each \( g_i \) is a strongly convex function with constant \( \sigma_i > 0 \), and \( Q_i \subseteq \mathbb{R}^n \) are closed convex sets. Denote with \( v^* \) the unique optimal solution of problem (29), which we assume to exist. Then, this primal convex problem can be reformulated as:

\[ \min_{u_i \in Q_i, u_i = v \ \forall i \in [N]} g_1(u_1) + \cdots + g_N(u_N). \]

Let us define \( u = [u_1^T \cdots u_N^T]^T \) and \( g(u) = g_1(u_1) + \cdots + g_N(u_N) \). Under standard constraint qualification conditions, using duality with the Lagrange multipliers \( x_i \) for the constraints \( u_i = v \), we obtain the equivalent convex problem (1), where \( f_i(x_i) = \tilde{g}_i^*(x_i) \) and \( \tilde{g}_i^* \) is the convex conjugate of the function \( \tilde{g}_i = g_i + 1_{Q_i} \), i.e.,:

\[ f_i(x_i) = \max_{u_i \in Q_i} \langle x_i, u_i \rangle - g_i(u_i) \ \forall i. \]  

(30)

Further we have \( f^* + g^* = 0 \). Note that if \( g_i \) is \( \sigma_i \)-strongly convex, then the convex conjugate \( f_i \) is well defined and has Lipschitz continuous gradient with constants \( L_i = \frac{1}{\sigma_i} \) (see [23]), so that Assumption 2.1 holds. A particular application is the problem of finding the projection of a point \( v_0 \) on the intersection of the closed convex sets \( \cap_{i=1}^N Q_i \subseteq \mathbb{R}^n \), denoted \( [v_0] \cap_{i=1}^N Q_i \). This convex feasibility problem can be written as an optimization problem in the form:

\[ \min_{v \in \cap_{i=1}^N Q_i} \pi_1 \|v - v_0\|^2 + \cdots + \pi_N \|v - v_0\|^2, \]  

(31)

where \( \pi_i > 0 \) such that \( \sum_i \pi_i = 1 \). This is a special case of the separable problem (29). Since the functions \( g_i(v) = \pi_i \|v - v_0\|^2 \) are strongly convex, functions \( f_i \) have Lipschitz continuous gradient with constants \( L_i = 1/\pi_i \) for all \( i \in [N] \). Note that the optimal solution for (31) is \( v^* = [v_0] \cap_{i=1}^N Q_i \).

We now show how we can recover an approximate primal solution for the primal problem (29) by solving the corresponding dual problem (1) with algorithm RCD_\( \tau \). Let us define for any dual variable \( x_i \) the primal variable:

\[ u_i(x_i) = \arg \min_{u_i \in Q_i} g_i(u_i) - \langle x_i, u_i \rangle \ \forall i \in [N]. \]

Let us define \( \hat{v}^* = e \otimes v^* \), where \( \otimes \) is the Kronecker product, and the norm \( \|u\|^2_{D_\sigma} = \sum_i \sigma_i \|u_i\|^2 \). Moreover, let \( \sigma_{\min} = \min_i \sigma_i \) and \( \lambda_N \) the largest eigenvalue of \( G_\tau \). Furthermore, for simplicity of the presentation, consider an initial starting point \( x^0 = 0 \). Then, we can derive the following convergence estimates on primal infeasibility and suboptimality for convex problem (29).

\( \odot \) Springer
**Theorem 6.1** Assume all functions \( g_i \) in convex optimization problem (29) are \( \sigma_i \)-strongly convex. Let \( x^k \) be the sequence generated by algorithm RCD\( \tau \) for solving the corresponding dual problem (1), and let the primal sequence be \( u^k = u(x^k) \). Then, we have the following convergence estimates in expected value on both primal infeasibility and suboptimality:

\[
E \left[ \left\| u^k - \hat{v}^* \right\|_{D_\sigma}^2 \right] \leq \frac{4R^2(x^0)}{k} \quad \text{and} \quad E \left[ |g(u^k) - g^*| \right] \leq \frac{4R^2(x^0)\lambda N}{\sigma_{\min} \sqrt{k}}.
\]

**Proof** Since all the functions \( g_i \) are \( \sigma_i \)-strongly convex, the objective function \( \sum_{i=1}^N g_i(u_i) - \langle x_i, u_i \rangle \) is also 1-strongly convex in the variable \( u \) w.r.t. the norm \( \|u\|_{D_\sigma}^2 = \sum_i \sigma_i \|u_i\|^2 \). Using this property and the expression of \( u_i(x_i) \), we obtain the following inequalities:

\[
\frac{1}{2} \|u(x) - \hat{v}^*\|_{D_\sigma}^2 = \sum_{i=1}^N \frac{\sigma_i}{2} \|u_i(x_i) - v^*\|^2 \\
\leq \left( \sum_{i=1}^N g_i(v^*) - \langle x_i, v^* \rangle \right) - \left( \sum_{i=1}^N g_i(u_i(x_i)) - \langle x_i, u_i(x_i) \rangle \right) \\
= (-f^*) - (-f(x)) = f(x) - f^* \quad \forall x \in S. \tag{32}
\]

Now, let us consider for \( x \) the sequence \( x^k \) generated by algorithm RCD\( \tau \) and let \( u^k = u(x^k) \). We note that \( u^{k+1}_i = u^k_i \) for all \( i \in [N] \setminus \mathcal{N}_k \) and that \( u^{k+1}_i = u_i(x^{k+1}_i) \) for all \( i \in \mathcal{N}_k \). Taking expectation over the entire history \( \eta^k \) and using Theorem 4.1, we get an estimate on primal infeasibility:

\[
E[\|u^k - \hat{v}^*\|_{D_\sigma}^2] \leq 2E[f(x^k) - f^*] = 2(\phi_k - f^*) \leq \frac{4R^2(x^0)}{k}.
\]

Furthermore, to derive estimates on primal suboptimality, first observe:

\[
\|u\|_{G_\tau} \leq \frac{\lambda N}{\sigma_{\min}} \|u\|_{D_\sigma} \quad \forall u,
\]

and combine with (32) to get:

\[
\|u(x) - \hat{v}^*\|_{G_\tau} \leq \frac{\lambda N \sqrt{2(f(x) - f^*)}}{\sigma_{\min}}. \tag{33}
\]

For the left-hand side suboptimality, we proceed as follows:

\[
f(x^*) = \langle x^*, \hat{v}^* \rangle - g(\hat{v}^*) = \max_{u_i \in Q_i} \langle x^*, u \rangle - g(u) \geq \langle x^*, u(x) \rangle - g(u(x)),
\]
which leads to the following relation:

\[
g(u(x)) - g^* \geq \langle x^*, u(x) - \hat{v}^* \rangle \geq -\|x^*\|^G_r \|\hat{v}^* - u(x)\|_G \tag{33}
\]

\[
\geq -\|x^*\|^G_r \frac{\lambda \sqrt{2(f(x) - f^*)}}{\sigma_{\min}} \quad \forall x \in S.
\]

(34)

Secondly, from the definition of the dual function, we have:

\[
f(x) = \langle x, u(x) \rangle - g(u(x)) \quad \forall x \in S.
\]

Subtracting \( f^* = f(x^*) \) from both sides and using the complementarity condition \( \langle x^*, u(x^*) \rangle = 0 \), where \( u(x^*) = \hat{v}^* \), we get the following relations:

\[
g(u(x)) - g^* = \langle x, u(x) \rangle - f(x) + f^*
\]

\[
= f(x^*) - f(x) + \langle x - x^*, u(x^*) \rangle + \langle x, u(x) - u(x^*) \rangle
\]

\[
= f(x^*) + \langle x - x^*, \nabla f(x^*) \rangle - f(x) + \langle x, u(x) - u(x^*) \rangle
\]

\[
\leq \langle x, u(x) - u(x^*) \rangle \leq \|x\|^G_r \|\hat{v}^* - u(x)\|_G
\]

\[
\leq (\|x - x^*\|^G_r + \|x^*\|^G_r) \|u(x) - \hat{v}^*\|_G
\]

\[
\leq \frac{\lambda \sqrt{2(f(x) - f^*)}}{\sigma_{\min}}, \tag{33}
\]

valid for all \( x \in S \) and \( x^* \in X^* \), where in the first inequality we used convexity of the function \( f \) and the relation \( \nabla f(x) = u(x) \), and in the second inequality the Cauchy–Schwarz inequality. Now, using the definition of \( R(x^0) \) and the fact that \( x^0 = 0 \), replacing \( x \) with the sequence \( \lambda^k \) in the previous derivations and taking the expectation over the entire history \( \eta^k \), we obtain a bound on primal suboptimality:

\[
E[\|g(u^k) - g^*\|] \leq E \left[ \left( \|x^k - x^*\|^G_r + \|x^*\|^G_r \right) \frac{\lambda \sqrt{2(f(x^k) - f^*)}}{\sigma_{\min}} \right]
\]

\[
\leq \frac{2R(x^0)\lambda_N}{\sigma_{\min}} \frac{\sqrt{2E[f(x^k) - f^*]} \leq \frac{4R^2(x^0)\lambda_N}{\sigma_{\min} \sqrt{k}},
\]

which gives us a convergence estimate for primal suboptimality for (29). \( \Box \)

For the convex feasibility problem (31), our algorithm \( \text{RCD}_r \) leads to a new random alternating projection method for finding a point in the intersection of general convex sets. In particular, for \( \tau = 2 \), given a starting point \( x^0 \) satisfying \( \sum_i x^0_i = 0 \) and a probability distribution \( (p_{ij})_{i,j \in [N]} \) over the set \([N] \times [N] \), we obtain a random alternating projection method that requires projections onto two sets chosen randomly and then perform simple updates as follows:
Algorithm 2-Random Alternating Projection (2-RAP)

1. choose convex sets $Q_{ik}$ & $Q_{jk}$ according to probabilities $(p_{ij})_{i,j\in[N]}$
2. compute $u_{ik}^k = \left[ v_0 - 1/\pi_{ik} x_i^k \right]_{Q_{ik}}$ and $u_{jk}^k = \left[ v_0 - 1/\pi_{jk} x_j^k \right]_{Q_{jk}}$
3. update $x_{ik}^{k+1} = x_i^k + \frac{u_{ik}^k - u_{jk}^k}{1/\pi_{ik} + 1/\pi_{jk}}$, $x_{jk}^{k+1} = x_j^k + \frac{u_{jk}^k - u_{ik}^k}{1/\pi_{ik} + 1/\pi_{jk}}$, $u_i^k = u_{i}^{k-1}$ and $x_{i}^{k+1} = x_i^k \forall i \neq i_k, j_k$.

where we recall that $[v]_{Q}$ denotes the projection of vector $v$ onto the closed convex set $Q$. Then, from Theorem 6.1, the expected value of the distance between the primal generated points $u_{ik}^k \in Q_i$ and the unique optimal point $v^* = \left[ v_0 \right] \cap \cap_{i=1}^N Q_i$ of (31), i.e., $E[\|u_{ik}^k - v^*\|]$, is less than $O\left(1/\sqrt{k}\right)$ for all $i \in [N]$. Note that the usual alternating projection methods for finding a point in the intersection of general closed convex sets $Q_i$ generate a sequence $v^k$ such that $\sum_{i=1}^N \|v^k - [v^k]_{Q_i}\|$ is less than $O\left(1/\sqrt{k}\right)$, see, e.g., [5,6]. We conclude that our estimate for the convergence rate of the 2-random alternating projection algorithm 2-RAP is stronger than some existing convergence results for other alternating projection methods from the literature.

7 Numerical Experiments

In this section we briefly present some existing methods for solving optimization problem (1) and provide preliminary numerical results comparing some of these methods with algorithm RCD$_\tau$.

7.1 Previous Work

We first assume that we measure distances in $\mathbb{R}^N$ with the standard Euclidean norm and consider the corresponding centralized gradient method. Based on Assumption 2.1 we can derive:

$$f(x+s) \leq f(x) + \langle \nabla f(x), s \rangle + \frac{L_{\text{max}}}{2} \|s\|^2 \quad \forall x, s \in \mathbb{R}^N,$$

where $L_{\text{max}} = \max_{i\in[N]} L_i$. Then, the updates in centralized gradient method are:

$$x_i^+ = x_i + \frac{1}{NL_{\text{max}}} \sum_{j=1}^N (\nabla f_j(x_j) - \nabla f_i(x_i)) \quad \forall i \in [N].$$

Therefore, if we consider the Euclidean norm in $\mathbb{R}^N$, the convergence rate of the centralized gradient method endowed with this norm is [23]:

$$\square$$ Springer
\[ f(x^k) - f^* \leq \frac{2 \sum_{i=1}^{N} L_{\max} R_i^2}{k}. \] (35)

On the other hand, if we measure distances in the norm defined in this paper, i.e., \( \| \cdot \|_{G^\tau} \), then we get for \( \tau = N \) and a complete graph a distributed gradient method RCD\(_N\) whose updates are given by:

\[
x_i^+ = x_i + \frac{1}{L_i} \sum_{j=1}^{N} \frac{1}{L_j} \left( \nabla f_j(x_j) - \nabla f_i(x_i) \right) \quad \forall i \in [N].
\]

In these settings the matrix \( G_N \) and the induced norm have the expressions:

\[
G_N = \text{diag}(L^{-1}) - \frac{1}{\sum_i 1/L_i} L^{-1}(L^{-1})^T \quad \text{and} \quad \|x\|_{G_N}^* = \sum_{i=1}^{N} L_i \|x_i\|^2 \quad \forall x \in S,
\]

where we recall that \( L = [L_1 \cdots L_N]^T \). Note that \( G_N \) is a weighted Laplacian matrix for the complete graph. We conclude from Theorem 4.1 that the convergence rate for the distributed gradient method endowed with the previous norm, i.e., for algorithm RCD\(_N\), is:

\[
f(x^k) - f^* \leq \frac{2 \sum_{i=1}^{N} L_i R_i^2}{k}, \] (36)

which is definitely better than (35) corresponding to the centralized version endowed with the Euclidean norm.

Center-free full gradient methods were derived in [2] for solving distributed optimization problems (1) over a graph, featuring a smooth and strongly convex objective function. More precisely, the following updates based on full gradient computation are proposed in [2]:

\[
x_i^+ = x_i + \sum_{j \in \mathcal{N}(i)} w_{ij} (\nabla f_j(x_j) - \nabla f_i(x_i)) \quad \forall i \in [N],
\]

where \( \mathcal{N}(i) \) denotes the neighbors of node \( i \) in the graph, and the weights must satisfy \( \sum_{j \in \mathcal{N}(i)} w_{ij} = 0 \). A linear rate of convergence is obtained for the center-free algorithm of [2], similar to that of Theorem 4.2, under the Lipschitz gradient and strong convexity assumption on \( f \), where the weights are designed based on the graph structure (e.g., Metropolis weights) or based on solving an SDP problem. Randomized algorithm RCD\(_\tau\) also belongs to this class of methods, and we recover for \( \tau = N \) a similar center-free method as in [2]; however, in our case, weights are designed based on the Lipschitz constants of the gradients of the objective function, and the probabilities can be designed via an SDP. Moreover, our convergence analysis also covers the smooth case, i.e., without requiring the strong convexity assumption on \( f \).

Furthermore, when \( \tau = 2 \), algorithm RCD\(_2\) is related to the methods in [1,14,17,18]. For example, in [17] ([14,18]) 2-coordinate greedy (random) descent...
methods are developed for minimizing a smooth function subject to a single linear equality constraint and additional bound constraints (general regularization terms) on the decision variables. Although the algorithm in [17] is able to take into account bound constraints, it is not adequate for distributed computations over a network, since it is based on a greedy coordinate search. The random choice of coordinates we use renders our algorithm \( \text{RCD}_T \) more flexible and adequate for parallel and distributed implementations. Moreover, the convergence rate analysis in this paper is different from the one in [1], which is based on geometric tools.

Note that Algorithm \( \text{RCD}_T \) can be also extended to non-separable objective functions \( f(x_1, \ldots, x_N) \), see, e.g., [1] for a detailed exposition. Finally, our approach is very general, allowing to analyze in the same framework several methods: full gradient as in [2] \((\tau = N)\), serial 2-coordinate descent as in [1,14,17] \((\tau = 2)\), but also any parallel random coordinate descent method in between \((2 < \tau < N)\), which can be very important, especially when dealing with incomplete structure of information in networks.

### 7.2 Numerical Tests

In this section we report some preliminary numerical results on solving optimization problem (1) where functions \( f_i \) are taken as in [2]:

\[
f_i(x_i) = \frac{1}{2} a_i (x_i - c_i)^2 + \log(1 + \exp(b_i(x_i - d_i))) \quad \forall i \in [N],
\]

with coefficients \( b_i, c_i \) and \( d_i \) randomly generated with uniform distributions on \([-15, 15]\), and \( a_i \geq 0 \) randomly generated with uniform distribution on \([0, 15]\).

The second derivatives of these functions are:

\[
f_i''(x_i) = a_i + \frac{b_i^2 \exp(b_i(x_i - d_i))}{(1 + \exp(b_i(x_i - d_i)))^2},
\]

which have the following lower and upper bounds: \( \sigma_i = a_i \) and \( L_i = a_i + \frac{1}{4} b_i^2 \). We assume that our problem forces all variables to sum to zero, i.e., we impose \( \sum_{i=1}^{N} x_i = 0 \). In all our numerical tests we consider a complete graph and choose Lipschitz-dependent probabilities as in (16) (unless specified otherwise).

In the first set of experiments, we solve a single randomly generated problem with \( N = 10^4 \) nodes for \( \tau = 2, 4 \) and 7 cores in parallel using MPI. Figure 1 displays the evolution of \( f(x^k) - f^* \) along normalized iterations \( k/N \) of algorithm \( \text{RCD}_T \). From the plot we can observe that increasing the number of cores reduces \( \tau \) substantially reduces the number of full iterations \( k/N \).

Then, we tested algorithm \( \text{RCD}_2 \) on a randomly generated problem of dimension \( N = 300 \) and with \( a_i = 0 \) for all \( i \in [N] \). Thus, at each iteration, we choose a pair of nodes in the graph \( (i, j) \in E \) with probability \( p_{ij} \) and then update only the components \( i \) and \( j \) of \( x \) as follows:
\[ x_i^+ = x_i + \frac{1}{L_i + L_j} \left( \nabla f_j (x_j) - \nabla f_i (x_i) \right), \]
\[ x_j^+ = x_j + \frac{1}{L_i + L_j} \left( \nabla f_i (x_i) - \nabla f_j (x_j) \right). \]

We consider in Fig. 2 three choices for the probability distribution of algorithm RCD\(_2\): uniform probability, probabilities depending on Lipschitz constants as given in (16) and optimal probabilities obtained from solving the SDP (27). As expected, the method based on optimal probabilities converges the fastest.

Finally, on a randomly generated problem of dimension \( N = 10^4 \), we compare our algorithm RCD\(_2\) for two choices of probabilities \( p_{ij} \) (uniform and Lipschitz-dependent probabilities (16)) with the centralized full gradient method (\( L_{\text{max}} = \max_i L_i \)) and the center-free gradient method with metropolis weights proposed in [2]. We plot in Fig. 3 the evolution of \( f(x^k) - f^* \) along normalized iterations \( k/N \) for those four
methods. We clearly see that fastest convergence is achieved by the RCD_2 algorithm with Lipschitz-dependent probabilities.

8 Conclusions

In this paper we have derived parallel random (block) coordinate descent methods for minimizing large-scale linearly constrained convex problems over networks. Since our problem features coupled constraints, the algorithms we have devised update in parallel at each iteration \( \tau \geq 2 \) random (block) components of the solution. We have also provided a SDP formulation enabling us to choose the probabilities of those coordinate updates in an optimal fashion. We have proved that our method reaches an \( \epsilon \)-accurate solution in expectation after at most \( O\left(\frac{N}{\epsilon^2}\right) \) iterations, and thus the convergence rate depends linearly on the number of (block) components to be updated in parallel. Our methods also converge linearly for strongly convex functions. Preliminary numerical results show that our parallel coordinate descent method compares favorably with existing algorithms, and that variants with \( \tau > 2 \) accelerate compared to the basic \( \tau = 2 \) counterpart.

Acknowledgements This research received funding from UEFISCDI Romania, PNII-RU-TE, project MoCOBiDS, no. 176/01.10.2015. It presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, and of the Concerted Research Action (ARC) programme supported by the Federation Wallonia-Brussels (contract ARC 14/19-060). Support from two WBI-Romanian Academy grants is also acknowledged. Scientific responsibility rests with the authors.

References

1. Necoara, I.: Random coordinate descent algorithms for multi-agent convex optimization over networks. IEEE Trans. Autom. Control 58(8), 2001–2012 (2013)
2. Xiao, L., Boyd, S.: Optimal scaling of a gradient method for distributed resource allocation. J. Optim. Theory Appl. **129**(3), 469–488 (2006)
3. Ishii, H., Tempo, R., Bai, E.: A web aggregation approach for distributed randomized pagerank algorithms. IEEE Trans. Autom. Control **57**(1), 2703–2717 (2012)
4. You, K., Xie, L.: Network topology and communication data rate for consensusability of discrete-time multi-agent systems. IEEE Trans. Autom. Control **56**(10), 2262–2275 (2011)
5. Bauschke, H., Borwein, J.: On projection algorithms for solving convex feasibility problems. SIAM Rev. **38**(3), 367–426 (1996)
6. Combettes, P.: The convex feasibility problem in image recovery. In: Hawkes, P. (ed.) Advances in Imaging and Electron Physics, pp. 155–270. Academic Press, Cambridge (1996)
7. Wright, S.: Accelerated block coordinate relaxation for regularized optimization. SIAM J. Optim. **22**(1), 159–186 (2012)
8. Liu, J., Wright, S.: Asynchronous stochastic coordinate descent: parallelism and convergence properties. SIAM J. Optim. **25**(1), 351–376 (2014)
9. Qin, Z., Scheinberg, K., Goldfarb, D.: Efficient block-coordinate descent algorithms for the group lasso. Math. Program. Comput. **5**(2), 143–169 (2013)
10. Richtarik, P., Takac, M.: Parallel coordinate descent methods for big data optimization. Math. Program. **156**(1), 433–484 (2016)
11. Beck, A., Tretuashvili, L.: On the convergence of block coordinate descent type methods. SIAM J. Optim. **23**(4), 2037–2060 (2013)
12. Tseng, P., Yun, S.: A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. J. Optim. Theory Appl. **140**(3), 513–535 (2009)
13. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(1), 341–362 (2012)
14. Patrascu, A., Necoara, I.: Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. J. Global Optim. **61**(1), 19–46 (2015)
15. Richtarik, P., Takac, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1–2), 1–38 (2014)
16. Necoara, I., Clipici, D.: Parallel coordinate descent methods for composite minimization: convergence analysis and error bounds. SIAM J. Optim. **26**(1), 197–226 (2016)
17. Beck, A.: The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. J. Optim. Theory Appl. **162**(3), 892–919 (2014)
18. Necoara, I., Patrascu, A.: A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Comput. Optim. Appl. **57**(2), 307–337 (2014)
19. Reddi, S., Hefny, A., Downey, C., Dubey, A., Sra, S.: Large-scale randomized-coordinate descent methods with non-separable linear constraints. Tech. rep. (2014). http://arxiv.org/abs/1409.2617.pdf
20. Necoara, I., Nesterov, Y., Glineur, F.: A random coordinate descent method on large optimization problems with linear constraints. Tech. rep. (2011). https://acse.pub.ro/person/ion-necoara
21. Hong, M., Luo, Z.: On the linear convergence of the alternating direction method of multipliers. Math. Program. (2016). doi:10.1007/s10107-016-1034-2
22. Wei, E., Ozdaglar, A., Jadababaie, A.: A distributed Newton method for network utility maximization part II. IEEE Trans. Autom. Control **58**(9), 2176–2188 (2013)
23. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Boston (2004)
24. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer, Berlin (2001)