A RANDOM COORDINATE DESCENT METHOD ON LARGE-SCALE OPTIMIZATION PROBLEMS WITH LINEAR CONSTRAINTS

I. NECOARA*, Y. NESTEROV† AND F. GLINEUR† *

Abstract. In this paper we develop a random block coordinate descent method for minimizing large-scale convex problems with linearly coupled constraints and prove that it obtains in expectation an €-accurate solution in at most $O(\frac{1}{\epsilon})$ iterations. However, the numerical complexity per iteration of the new method is usually much cheaper than that of methods based on full gradient information. We focus on how to choose the probabilities to make the randomized algorithm to converge as fast as possible and we arrive at solving a sparse SDP. Analysis for rate of convergence in probability is also provided. For strongly convex functions the new method converges linearly. Numerical tests confirm that on large optimization problems with cheap coordinate derivatives the new method is much more efficient than methods based on full gradient.

Key words. Large convex optimization problems, Lipschitz continuous gradient functions, linearly coupled constraints, block-coordinate descent method, random algorithm, connected graphs.

1. Introduction. The performance of a network composed of interconnected subsystems can be increased if the traditionally separated subsystems are jointly optimized. Recently, parallel and distributed optimization methods have emerged as a powerful tool for solving large network optimization problems: e.g. resource allocation [7, 8, 20], telecommunications [1, 13], coordination in multi-agent systems [20], estimation in sensor networks [14], distributed control [14], image processing [4], traffic equilibrium problems [1], network flow [1] and other areas [10, 16, 22].

The goal of this paper is to develop an efficient distributed algorithm with cheap iteration for solving large separable convex problems with linearly coupled constraints that arise in network applications. The problems that we consider in this paper have the following features: the size of data is very large so that usual methods based on whole gradient computations are prohibitive. Moreover the incomplete structure of information (e.g. the data are distributed over the all nodes of the network, so that at a given time we need to work only with the data available then) may also be an obstacle for whole gradient computations. In this case, an appropriate way to approach these problems is through coordinate descent methods. These methods were among the first optimization methods studied in literature [1] but until recently they haven’t received much attention.

The main differences in all variants of coordinate descent methods consist in the criterion of choosing at each iteration the coordinate over which we minimize the objective function and the complexity of this choice. Two classical criteria used often in these algorithms are the cyclic [1] and the greedy descent coordinate search [19], which significantly differs by the amount of computations required to choose the appropriate index. For cyclic coordinate search estimates on the rate of convergence were given recently in [3], while for the greedy coordinate search (e.g. Gauss-Southwell rule) the convergence rate is given e.g. in [19]. Another interesting approach is based on random choice rule, where the coordinate search is random. Recent complexity results on random coordinate descent methods for smooth convex functions were obtained by

*Automatic Control and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania, (ion.necoara@acse.pub.ro).
†Center for Operations Research and Econometrics, Catholic University of Louvain, Louvain-la-Neuve, B-1348, Belgium ({Yuri.Nesterov, Francois.Glineur}@uclouvain.be).
Nesterov in [12]. The extension to composite functions was given in [17]. However, in most of the previous work the authors considered optimization models where the constraint set is decoupled (i.e. characterized by Cartesian product).

In this paper we develop a random block coordinate descent method suited for large optimization problems in networks where the information cannot be gather centrally, but rather the information is distributed over the network. Moreover, in our paper we focus on optimization problems with linearly coupled constraints (i.e. the constraint set is coupled). Due to the coupling in the constraints we introduce a 2-block variant of random coordinate descent method, that involves at each iteration the closed form solution of an optimization problem only with respect to two block variables while keeping all the other variables fixed. We prove for the new algorithm an expected convergence rate of order $O\left(\frac{1}{k}\right)$ for the function values, where $k$ is the number of iterations. We focus on how to design the probabilities to make this algorithm to converge as fast as possible and we prove that this problem can be recast as a sparse SDP. We also show that for functions with cheap coordinate derivatives the new method is faster than schemes based on full gradient information or based on greedy coordinate descent. Analysis for rate of convergence in probability is also provided. For strongly convex functions we prove that the new method converges linearly. While the most obvious benefit of randomization is that it can lead to faster algorithms, either in worst case complexity analysis and/or numerical implementation, there are also other benefits of our algorithm that are at least as important. For example, the use of randomization leads to a simpler algorithm that is easier to analyze, produces a more robust output and can often be organized to exploit modern computational architectures (e.g. distributed and parallel computer architectures).

The paper is organized as follows. In Section 2 we introduce the optimization model analyzed in this paper and the main assumptions. In Section 3 we present and analyze a random 2-block coordinate descent method for solving our optimization problem, derive the convergence rate in expectation and in probability and provide means to choose the probability distribution. We also compare our algorithm with the full projected gradient method and other existing methods and show that on problems with cheap coordinate derivatives our method has better arithmetic complexity. In section 4 we extend our algorithm to more than a pair of indexes. We also provide extensive numerical tests to prove the efficiency of the proposed algorithm.

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 Euclidian inner product $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$ and Euclidian norm $\|x\| = \langle x, x \rangle^{1/2}$. We use the same notation $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ for spaces of different dimension. The inner product on the space of symmetric matrices is denoted with $\langle W_1, W_2 \rangle = \text{trace}(W_1 W_2)$ for all $W_1, W_2$ symmetric matrices. We decompose the full space $\mathbb{R}^{Nn} = \prod_{i=1}^{N} \mathbb{R}^{n}$. We also define the corresponding partition of the identity matrix: $I_{Nn} = [U_1 \cdots U_N]$, where $U_i \in \mathbb{R}^{n \times n}$. Then for any $x \in \mathbb{R}^{Nn}$ we write $x = \sum_{i=1}^{N} U_i x_i$. We denote $e \in \mathbb{R}^N$ the vector with all entries 1 and $e_i \in \mathbb{R}^N$ the vector with all entries zero except the component $i$ equal to 1. Furthermore, we define: $U = [I_n \cdots I_n] = e^T \otimes I_n \in \mathbb{R}^{n \times Nn}$ and $V_i = [0 \cdots U_i \cdots 0] = e^T \otimes U_i \in \mathbb{R}^{Nn \times Nn}$, where $\otimes$ denotes the Kronecker product. Given a vector $\nu = [\nu_1 \cdots \nu_N]^T \in \mathbb{R}^N$, we define the vector $\nu^p = [\nu_1^p \cdots \nu_N^p]^T$ for any integer $p$ and $\text{diag}(\nu)$ denotes the diagonal matrix with the entries $\nu_i$ on the diagonal. For a positive semidefinite matrix $W \in \mathbb{R}^{N \times N}$ we consider the following order on its eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and the notation $\|x\|_W^2 = x^T W x$ for any $x$. 

I. Necoara, Y. Nesterov, F. Glineur
We consider large network optimization problems where each agent in the network is associated a local variable so that their sum is fixed and we need to minimize a separable convex objective function:

\[
\begin{align*}
 f^* &= \min_{x_i \in \mathbb{R}^n} f_1(x_1) + \cdots + f_N(x_N) \\
 \text{s.t.: } x_1 + \cdots + x_N &= 0.
\end{align*}
\] (2.1)

Optimization problems with linearly coupled constraints (2.1) arise in many areas such as resource allocation in economic systems [8] or distributed computer systems [9], in signal processing [4], in traffic equilibrium and network flow [1] or distributed control [14]. For problem (2.1) we associate a network composed of several nodes \( V = \{1, \ldots, N\} \) that can exchange information according to a communication graph \( \mathcal{G} = (V, E) \), where \( E \) denotes the set of edges, i.e. \((i, j) \in E \subseteq V \times V\) models that node \( j \) sends information to node \( i \). We assume that the graph \( \mathcal{G} \) is undirected and connected. The local information structure imposed by the graph \( \mathcal{G} \) should be considered as part of the problem formulation. Note that constraints of the form \( \alpha_1 x_1 + \cdots + \alpha_N x_N = b \), where \( \alpha_i \in \mathbb{R} \), can be easily handled in our framework by a change of coordinates.

The goal of this paper is to devise a distributed algorithm that iteratively solves the convex problem (2.1) by passing the estimate of the optimizer only between neighboring nodes. There is great interest in designing such distributed 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:

\[
S = \left\{ x \in \mathbb{R}^{Nn} : \sum_{i=1}^{N} x_i = 0 \right\},
\]

that has the orthogonal complement the subspace \( T = \{ u \in \mathbb{R}^{Nn} : u_1 = \cdots = u_N \} \).

We also use the notation:

\[
x = [x_1^T \cdots x_N^T]^T = \sum_{i=1}^{N} U_i x_i \in \mathbb{R}^{Nn}, \quad f(x) = f_1(x_1) + \cdots + f_N(x_N).
\]

The basic assumption considered in this paper is the following:

**Assumption A.1.** We assume that the functions \( f_i \) are convex and have Lipschitz continuous gradient, with Lipschitz constants \( L_i > 0 \), i.e.:

\[
||\nabla f_i(x_i) - \nabla f_i(y_i)|| \leq L_i ||x_i - y_i|| \quad \forall x_i, y_i \in \mathbb{R}^n, \quad i \in V.
\] (2.2)

From the Lipschitz property of the gradient (2.2), the following inequality holds (see e.g. Section 2 in [11]):

\[
f_i(x_i + d_i) \leq f_i(x_i) + \langle \nabla f_i(x_i), d_i \rangle + \frac{L_i}{2} ||d_i||^2 \quad \forall x_i, d_i \in \mathbb{R}^n.
\] (2.3)

The following inequality, which is central in our paper, is a straightforward consequence of (2.3) and holds for all \( x \in \mathbb{R}^{Nn} \) and \( d_i, d_j \in \mathbb{R}^n \):

\[
f(x + U_i d_i + U_j d_j) \leq f(x) + \langle \nabla f_i(x_i), d_i \rangle + \frac{L_i}{2} ||d_i||^2 + \langle \nabla f_j(x_j), d_j \rangle + \frac{L_j}{2} ||d_j||^2.
\] (2.4)
We denote with $X^*$ the set of optimal solutions for problem (2.1). The optimality conditions for optimization problem (2.1) become: $x^*$ is optimal solution for the convex problem (2.1) if and only if
\[
\sum_{i=1}^{N} x_i^* = 0, \quad \nabla f_i(x_i^*) = \nabla f_j(x_j^*) \quad \forall i \neq j \in V.
\]

2.1. Previous work. We briefly review some well-known methods from the literature for solving the optimization model (2.1). In [7, 20] distributed weighted gradient methods were proposed to solve a similar problem as in (2.1), in particular the authors in [20] consider strongly convex functions $f_i$ with positive definite Hessians. These papers propose a class of center-free algorithms (in these papers the term center-free refers to the absence of a coordinator) with the following iteration:
\[
x_i^{k+1} = x_i^k - \sum_{j \in \mathcal{N}_i} w_{ij} (\nabla f_j(x_j^k) - \nabla f_i(x_i^k)) \quad \forall i \in V, \; k \geq 0,
\]
where $\mathcal{N}_i$ denotes the set of neighbors of node $i$ in the graph $\mathcal{G}$. Under the strong convexity assumption and provided that the weights $w_{ij}$ are chosen as a solution of a certain SDP, linear rate of convergence is obtained. Note however, that this method requires at each iteration the computation of the full gradient and the iteration complexity is $O(N(n + n_f))$, where $n_f$ is the number of operations for evaluating the gradient of any function $f_j$ at the current iterate for all $i \in V$.

In [19] Tseng studied optimization problems with linearly coupled constraints and composite objective functions of the form $f + h$, where $h$ is convex nonsmooth function, and developed a block coordinate descent method based on the Gauss-Southwell choice rule. The principal requirement for this method is that at each iteration a subset of indexes $\mathcal{I}$ needs to be chosen with respect to the Gauss-Southwell rule and then the update direction is a solution of the following QP problem:
\[
d_H(x; \mathcal{I}) = \arg \min_{s: \sum_{j \in \mathcal{I}} s_j = 0} \langle \nabla f(x), s \rangle + \frac{1}{2} \|s\|^2_H + h(x + s),
\]
where $H$ is a positive definite matrix chosen at the initial step of the algorithm. Using this direction and choosing an appropriate step size $\alpha^k$, the next iterate is defined as: $x^{k+1} = x^k + \alpha^k d_H(x^k; \mathcal{I}^k)$. The total complexity per iteration of this method is $O(N(n + n_f))$. In [19], the authors proved, for the particular case of a single linear constraint and the nonsmooth part $h$ of the objective function is piece-wise linear and separable, that after $k$ iterations a sublinear convergence rate of order $O\left(\frac{NnL^2}{k}\right)$ is attained for the function values, where $L = \max_{i \in V} L_i$ and $R_0$ is the Euclidian distance of the starting iterate to the set of optimal solutions. In [2] a 2-coordinate 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 the convex case, when all the variables are lower bounded but not upper bounded, the author shows that the sequence of function values converges at a sublinear rate $O\left(\frac{NnL^2}{k}\right)$, while the complexity per iteration is at least $O(N(n + n))$.

A random coordinate descent algorithm for an optimization model with smooth objective function an separable constraints was analyzed by Nesterov in [12], where a complete rate analysis is provided. The main feature of his randomized algorithm is the cheap iteration complexity of order $O(n_f + n + \ln N)$, while still keeping sublinear
rate of convergence. The generalization of this algorithm to composite objective function has been studied in [16,17]. However, none of these papers studied the application of random coordinate descent algorithms to smooth convex problems with linearly coupled constraints. In this paper we develop a random coordinate descent method for this type of optimization model as described in (2.1).

3. A random block coordinate descent method. In this section we devise a randomized block coordinate descent algorithm for solving the separable convex problem (2.1) and analyze its convergence. We present a distributed method where only two neighbors need to communicate with each other. At a certain iteration having a feasible estimate \( x \in S \) of the optimizer, we choose randomly a pair \((i,j) \in E\) with probability \( p_{ij} > 0 \). Since we assume an undirected graph \( G = (V, E) \) associated to problem (2.1) (the generalization of the present scheme to directed graphs is straightforward), we consider \( p_{ij} = p_{ji} \). We assume that the graph \( G \) is connected. For a feasible \( x \in S \) and a randomly chosen pair of indexes \((i,j), \) with \( i < j \), we define the next feasible step \( x^+ \in \mathbb{R}^n \) as follows:

\[
x^+ = x + U_i d_i + U_j d_j.
\]

Derivation of the directions \( d_i \) and \( d_j \) is based on the inequality (2.4):

\[
f(x^+) \leq f(x) + \langle \nabla f_i(x_i), d_i \rangle + \langle \nabla f_j(x_j), d_j \rangle + \frac{L_i}{2} \| d_i \|^2 + \frac{L_j}{2} \| d_j \|^2.
\]

Minimizing the right hand side of inequality (3.1), but imposing additionally feasibility for the next iterate \( x^+ \) (i.e. we require \( d_i + d_j = 0 \)), we arrive at the following local minimization problem:

\[
[d_i^T \ d_j^T]^T = \arg \min_{s_i, s_j \in \mathbb{R}^n \ : \ s_i + s_j = 0} \langle \nabla f_i(x_i), s_i \rangle + \langle \nabla f_j(x_j), s_j \rangle + \frac{L_i}{2} \| s_i \|^2 + \frac{L_j}{2} \| s_j \|^2
\]

that has the closed form solution

\[
d_i = -\frac{1}{L_i + L_j} (\nabla f_i(x_i) - \nabla f_j(x_j)), \quad d_j = -d_i.
\]

We also obtain from (2.4) the following decrease in the objective function, which shows that our method is a descent method:

\[
f(x^+_i) \leq f(x) - \frac{1}{2(L_i + L_j)} \| \nabla f_i(x_i) - \nabla f_j(x_j) \|^2.
\]

Now, let the starting point \( x^0 \) be feasible for our problem (2.1) and assume some probability distribution \((p_{ij})_{(i,j) \in E}\) available over the undirected graph \( G \), then we can present the new random coordinate descent method:

\[
\textbf{Algorithm (RCD): (Random 2-Block Coordinate Descent Method)}
\]

For \( k \geq 0 \) iterate:
1. Choose \((i_k, j_k) \in E\) with probability \( p_{i_k,j_k} \)
2. Update \( x^{k+1} = x^k - \frac{1}{L_{i_k} + L_{j_k}} (U_{i_k} - U_{j_k}) (\nabla f_{i_k}(x_{i_k}^k) - \nabla f_{j_k}(x_{j_k}^k)) \).

Clearly, Algorithm (RCD) is distributed since only two neighboring nodes in the graph need to communicate at each iteration. Further, at each iteration only two
components of \( x \) are updated, so that our method has low complexity per iteration and is very efficient on functions with cheap derivatives (we need to compute only two partial gradients \( \nabla f_i(x_i) \nabla f_j(x_j) \) in \( \mathbb{R}^{2n} \) compared to full gradient methods where the full gradient \( \nabla f(x) \) in \( \mathbb{R}^{Nn} \) is required). Finally, in our algorithm we maintain feasibility at each iteration, i.e. \( x_1^k + \cdots + x_N^k = 0 \) for all \( k \geq 0 \).

### 3.1. Efficiency of (RCD) method in expectation.

In this section we analyze the convergence rate of Algorithm (RCD) for the expected values of the objective function and in probability. After \( k \) iterations of the previous algorithm, we generate a random output \( (x^k, f(x^k)) \), which depends on the observed implementation of random variable:

\[
\eta^k = \{(i_0, j_0), \ldots, (i_k, j_k)\}.
\]

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

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

For simplicity of the exposition we use the following notation: given the current iterate \( x \), denote \( x^+ = x + U_id_i + U_jd_j \) the next iterate, where directions \( (d_i, d_j) \) are given by (3.2) for some random chosen pair \((i, j)\) w.r.t. a probability distribution. For brevity, we also adapt the notation of expectation upon the entire history, i.e. \((\phi, \eta)\) instead of \((\phi^k, \eta^k)\). For a feasible \( x \), taking the expected value over the random variable \((i, j)\), we obtain:

\[
f(x) - E[f(x^+) | \eta] = \sum_{(i,j) \in E} p_{ij}[f(x) - f(x^+)]
\]

\[
\geq \sum_{(i,j) \in E} \frac{p_{ij}}{2(L_i + L_j)} \|\nabla f_i(x_i) - \nabla f_j(x_j)\|^2
\]

\[
= \nabla f(x)^T \left( \sum_{(i,j) \in E} \frac{p_{ij}}{2(L_i + L_j)} G_{ij} \right) \nabla f(x),
\]

where \( G_{ij} = (e_i - e_j)(e_i - e_j)^T \otimes I_n \in \mathbb{R}^{Nn \times Nn} \). We introduce the weighted Laplacian of the underlying graph \( \mathcal{G} \) as being the matrix \( \mathcal{L} = \mathcal{L}(p_{ij}, L_i) \in \mathbb{R}^{N \times N} \) defined as:

\[
\mathcal{L}_{ij} = \begin{cases} 
-\frac{p_{ij}}{L_i + L_j} & \text{if } (i,j) \in E \\
\sum_{l \in N_i} \frac{p_{ij}}{L_i + L_l} & \text{if } i = j \\
0 & \text{if } (i,j) \notin E,
\end{cases}
\]

where \( N_i \) denotes the set of neighbors of node \( i \) in the graph \( \mathcal{G} \). Note that the Laplacian matrix \( \mathcal{L} \) is positive semidefinite and \( \mathcal{L}e = 0 \), i.e. it has the smallest eigenvalue \( \lambda_1(\mathcal{L}) = 0 \) with the associated eigenvector \( e \). Since the graph is connected, then it is well known that the eigenvalue \( \lambda_1(\mathcal{L}) = 0 \) is simple, i.e. \( \lambda_2(\mathcal{L}) > 0 \). We introduce the following set:

\[
\mathcal{M} = \left\{ \mathcal{L} \in \mathbb{R}^{N \times N} : \mathcal{L} \text{ defined in } (3.4), p_{ij} = p_{ji}, \sum_{(i,j) \in E} p_{ij} = 1 \right\}.
\]
A randomized coordinate descent method on large convex problems

Then, we have that the matrix $G_2 \in \mathbb{R}^{N \times N}$ defined as:

$$G_2 = \sum_{(i,j) \in E} \frac{p_{ij}}{L_i + L_j} G_{ij} = \mathcal{L} \otimes I_n$$

is also positive semidefinite. In conclusion we obtain the following useful inequality that shows the decrease of the objective function in expectation:

$$f(x) - E[f(x^+ | \eta)] \geq \frac{1}{2} \nabla f(x)^T G_2 \nabla f(x).$$

(3.5)

On the extended subspace $S$, we now define a norm that will be used subsequently for measuring distances in this subspace. We define the extended primal norm induced by the matrix $G_2$ as:

$$\|u\|_{G_2} = \sqrt{u^T G_2 u} \quad \forall u \in \mathbb{R}^N \setminus T.$$ 

On the subspace $S$, we introduce its extended dual norm:

$$\|x\|_{G_2}^* = \max_{u: \|u\|_{G_2} \leq 1} \langle x, u \rangle = \max_{u: \langle G_2 u, u \rangle \leq 1} \langle x, u \rangle \quad \forall x \in S.$$ 

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

$$\langle u, x \rangle \leq \|u\|_{G_2} \cdot \|x\|_{G_2}^* \quad \forall x \in S, u \in \mathbb{R}^N.$$ 

Let us compute this dual norm for any $x \in S$:

$$\|x\|_{G_2}^* = \max_{u \in \mathbb{R}^N: \langle G_2 u, u \rangle \leq 1} \langle x, u \rangle$$

$$= \max_{u: \langle G_2 u, u \rangle \leq 1} \langle x, u - \frac{1}{N} \sum_{i=1}^N u_i \rangle \leq \max_{u: \langle G_2 u, u \rangle \leq 1, U^T U u \leq 0} \langle x, u \rangle$$

$$= \min_{\mu, \nu \geq 0} \mu + \frac{1}{\nu} \|G_2 + \nu U^T U\|^{-1} x, x \rangle$$

$$= \min_{\mu, \nu \geq 0} \mu + \frac{1}{\nu} \|G_2 + \frac{\nu}{\mu} U^T U\|^{-1} x, x \rangle$$

$$= \min_{\xi \geq 0} \sqrt{\langle (G_2 + \xi \mathcal{L})^{-1} \otimes I_n x, x \rangle}.$$ 

(3.6)

In conclusion, we obtain an extended dual norm that is well defined in $S$:

$$\|x\|_{G_2}^* = \min_{\xi \geq 0} \sqrt{\langle (\mathcal{L} + \xi \mathcal{L})^{-1} \otimes I_n x, x \rangle} \quad \forall x \in S.$$ 

Using the eigenvalue decomposition of the Laplacian $\mathcal{L} = \Xi \text{diag}(0, \lambda_2, \ldots, \lambda_N) \Xi^T$, where $\lambda_i$ are the positive eigenvalues and $\Xi = [e \xi_2 \cdots \xi_N]$ such that $\langle e, \xi_i \rangle = 0$ for all
I. Necoara, Y. Nesterov, F. Glineur

\( i \in V \), then \((L + ee^T)^{-1} = \Xi \text{diag}(\|e\|^2, \lambda_2, \ldots, \lambda_N)^{-1} \Xi^T \). It is straightforward to see that our defined norm has the following closed form:

\[
\|x\|_{G_2}^* = \sqrt{x^T (L^+ \otimes I_n) x} \quad \forall x \in S,
\]

where \( L^+ = \Xi \text{diag}(0, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_N}) \Xi^T \) denotes the pseudoinverse of the matrix \( L \). On the other hand if we define \( L^{[N-1]} \) as the leading matrix of dimension \( N - 1 \) of \( L \) and \( x_{1:N-1} = [x_1^T \cdots x_{N-1}^T]^T \in \mathbb{R}^{(N-1)n} \), from the definition of the norm we also have:

\[
\|x\|_{G_2}^* = \max_{u : (G_2(u - e_N \otimes u_N), u - e_N \otimes u_N) \leq 1} \langle x, u - e_N \otimes u_N \rangle = \max_{u_{1:N-1} : ((L^{[N-1]} \otimes I_n) u_{1:N-1}, u_{1:N-1}) \leq 1} \langle x_{1:N-1}, u_{1:N-1} \rangle.
\]

The optimality condition in the previous maximization problem is given by:

\[
(L^{[N-1]} \otimes I_n) u_{1:N-1} = x_{1:N-1},
\]

In conclusion, we have:

\[
(3.7) \quad \|x\|_{G_2}^* = \sqrt{x_{1:N-1}^T (L^{[N-1]} \otimes I_n)^{-1} x_{1:N-1}} \quad \forall x \in S.
\]

Let us compute our defined norm for some important graphs:

1. For a cycle graph if we define the vector of inverse probabilities as:

\[
p = [1 \ p_{12} \ 1 \ p_{23} \ \ldots \ 1 \ p_{N1}]^T
\]

and the lower triangular matrix \( W \in \mathbb{R}^{N \times N} \) with all entries equal with 1 in the lower part, then the norm takes the closed form:

\[
(||x||_{G_2}^*)^2 = x^T \left( W^T (\text{diag}(p) - \frac{1}{e^T p} pp^T) W \right) x \quad \forall x \in S.
\]

2. For a star-shaped graph if we define the vector of inverse probabilities as:

\[
p = [1 \ p_{1N} \ 1 \ p_{2N} \ \ldots \ 1 \ p_{N-1N} \ 0]^T,
\]

then the norm takes the closed form:

\[
(||x||_{G_2}^*)^2 = x^T (\text{diag}(p) \otimes I_n) x \quad \forall x \in S.
\]

3. For a complete graph, if we take for the probabilities the expressions:

\[
(3.8) \quad p_{ij}^s = \frac{L_i + L_j}{N(N-1)L_{av}}, \quad L_{av} = \frac{1}{N} \sum_i L_i,
\]

then we can see immediately that

\[
(3.9) \quad L_s = \frac{1}{N(N-1)L_{av}} (NI_N - ee^T)
\]
and thus using matrix inversion lemma we get
\[(L_s^{[N-1]})^{-1} = (N-1)L_{av}(I_{N-1} + ee^T).\]

In this case from (3.7) we get:
\[(3.10) \quad \left(\|x\|^2_{G_2, x}\right)^2 = (N-1) \sum_{i=1}^{N} L_i \|x_i\|^2 \quad \forall x \in S.
\]

On the other hand, if we take for the probabilities the expressions:
\[(3.11) \quad p_{ij}^{inv} = \frac{1}{L_i} + \frac{1}{L_j} \left(\frac{N-1}{\sum_{i} 1/L_i}\right),
\]
then we can see immediately that
\[(3.12) \quad L_{inv} = \frac{1}{N-1} \left(\text{diag}(L^{-1}) - \frac{1}{\sum_{i} 1/L_i} L^{-1}(L^{-1})^T\right)
\]
and thus using again matrix inversion lemma we get
\[\quad \left(\mathcal{L}^{[N-1] \text{inv}}\right)^{-1} = (N-1)(\text{diag}(L_1 \ldots L_{N-1}) + L_N ee^T).
\]

In this case from (3.7) we obtain:
\[(3.13) \quad \left(\|x\|^2_{G_2, \text{inv}}\right)^2 = (N-1) \sum_{i=1}^{N} L_i \|x_i\|^2 \quad \forall x \in S.
\]

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 (see inequality (3.3)):
\[R(x^0) = \max_{x \in S : f(x) \leq f(x^0)} \max_{x^* \in X^*} \|x - x^*\|_{G_2},
\]
which measures the size of the level set of f given by x^0. We assume that this distance is finite for the initial iterate x^0. We now state and prove the main result of this section:

**Theorem 3.1.** Let Assumption 1 hold for the optimization problem (2.1) and the sequence \((x^k)_{k \geq 0}\) be generated by Algorithm (RCD). Then, we have the following rate of convergence for the expected values of the objective function:
\[(3.14) \quad \phi_k - f^* \leq \frac{2R^2(x^0)}{k}.
\]

**Proof.** From convexity of f and the definition of the norm \(\cdot\|_{G_2}\) we get:
\[f(x^l) - f^* \leq \langle \nabla f(x^l), x^l - x^* \rangle \leq \|x^l - x^*\|_{G_2} \cdot \|\nabla f(x^l)\|_{G_2} \leq R(x^0) \cdot \|\nabla f(x^l)\|_{G_2} \quad \forall l \geq 0.
\]

Combining this inequality with (3.5), we obtain:
\[f(x^l) - E[f(x^{l+1}) \mid \eta^l] \geq \frac{(f(x^l) - f^*)^2}{2R^2(x^0)},
\]
or equivalently

\begin{equation}
E[f(x^{l+1}) | \eta^l] - f^* \leq f(x^l) - f^* - \frac{(f(x^l) - f^*)^2}{2R^2(x^0)}.
\end{equation}

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 \( \Delta_{l+1} \leq \Delta_l \) 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 \( l = 0, \cdots, k - 1 \) we get that \( 0 \leq \frac{1}{\Delta_0} \leq \frac{1}{\Delta_k} - \frac{k}{2R^2(x^0)} \) from which we obtain the statement (3.14) of the theorem. \( \square \)

Theorem 3.1 shows that for smooth functions Algorithm (RCD) has a sublinear rate of convergence in expectation but with a low complexity per iteration. More specifically, the iteration complexity is of order \( O(n_f + n + \ln N) \), where we recall that \( n_f \) is the maximum cost of computing the gradient of each function \( f_i \) for all \( i \in V \), \( O(n) \) is the cost of updating \( x^+_{(i,j)} \) from \( x \) and \( \ln N \) is the cost of choosing randomly a pair of indices \((i, j)\) for a given probability distribution \((p_{ij})_{(i,j) \in E}\), where \( N \) is the number of nodes in the graph \( \mathcal{G} \). The convergence rate of our method (RCD) can be explicitly expressed for a complete graph and under some specific choice for probabilities, according the the discussion above. In particular, let us assume that we know some constants \( R_i > 0 \) such that for any \( x^* \in X^* \) and any \( x \) satisfying \( f(x) \leq f(x^0) \) we have:

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

and define \( R = [R_1 \cdots R_N]^T \). Since our Algorithm (RCD) is a descent method (see inequality (3.3)), it follows that:

\[ R(x_0) \leq \max_{x \in S: \|x - x^*_i\| \leq R_i} \max_{\forall i \in V} \max_{x^* \in X^*} \|x - x^*\|_{\mathcal{G}_x}. \]

For a complete graph and using probabilities in the form (3.8), it follows immediately the following convergence rate for Algorithm (RCD) (see (3.10)):

\begin{equation}
\phi_k - f^* \leq (N - 1) \frac{2 \sum_{i=1}^{N} L_{av} R_i^2}{k}.
\end{equation}

For a complete graph and using probabilities in the form (3.11), it follows the following convergence rate for Algorithm (RCD) (see (3.13)):

\begin{equation}
\phi_k - f^* \leq (N - 1) \frac{2 \sum_{i=1}^{N} L_i R_i^2}{k}.
\end{equation}
3.2. Design of the probabilities. We have several choices for the probabilities \((p_{ij})_{(i,j)\in E}\), which the randomized block coordinate descent Algorithm \((RCD)\) depends on. For example, we can choose probabilities dependent on the Lipschitz constants \(L_{ij}\):

\[
p_{ij}^\alpha = \frac{L_{ij}^\alpha}{L^\alpha}, \quad L^\alpha = \sum_{(i,j)\in E} L_{ij}^\alpha, \quad \alpha \geq 0.
\]

(3.18)

Note that for \(\alpha = 0\) we recover the uniform probabilities. Finally, we can design the probabilities from the convergence rate of the method. From the definition of the constants \(R_i\) it follows that:

\[
R(x_0) \leq \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} \|x\|^*_G_2.
\]

We have the freedom to choose the matrix \(G_2\) that depends on the probabilities (we recall that \(G_2 = \mathcal{L} \otimes I_n\) and \(\mathcal{L}\) depends linearly on \(p_{ij}\)). Therefore, we search for the probabilities \(p_{ij}\) that are the optimal solution of the following optimization problem:

\[
R^*(x_0) = \min_{p_{ij}} R(x_0) \leq \min_{G_2: G_2 = \mathcal{L} \otimes I_n, \mathcal{L} \in M} \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} \|x\|^*_G_2.
\]

In the next theorem we derive an easily computed upper bound on \(R(x_0)\) and we provide a way to suboptimally select the probabilities \(p_{ij}\):

**Theorem 3.2.** A suboptimal choice of probabilities \((p_{ij})_{(i,j)\in E}\) can be obtained as a solution of the following SDP problem whose optimal value is an upper bound on \(R^2(x_0)\), i.e.:

\[
(R^*(x_0))^2 \leq \min_{\mathcal{L} \in M, \zeta \geq 0, \nu \geq 0} \left\{ \langle \nu, R^2 \rangle : \begin{bmatrix} \mathcal{L} + \zeta e e^T & I_N \\ I_N & \text{diag}(\nu) \end{bmatrix} \succeq 0 \right\}.
\]

(3.19)

**Proof.** The previous optimization problem can be written as follows:

\[
R^2(x_0) = \min_{G_2: G_2 = \mathcal{L} \otimes I_n, \mathcal{L} \in M} \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} \left( \|x\|^*_G_2 \right)^2
\]

\[
= \min_{G_2: G_2 = \mathcal{L} \otimes I_n, \mathcal{L} \in M} \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} \min_{\zeta \geq 0} ((G_2 + \zeta U^T U)^{-1} x, x)
\]

\[
= \min_{G_2: G_2 = \mathcal{L} \otimes I_n, \mathcal{L} \in M} \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} ((G_2 + \zeta U^T U)^{-1} x, x)
\]

\[
= \min_{G_2: G_2 = \mathcal{L} \otimes I_n, \mathcal{L} \in M} \max_{x : \|x_i\| \leq R_i, \forall i, \sum_{i=1}^N x_i = 0} ((G_2 + \zeta U^T U)^{-1}, xx^T).\]
Using the following well-known relaxation from the SDP literature, we have:

$$\min_{G_2 \in X \geq 0, \text{rank} X = 1, (X, V_i) \leq R_i^2 \forall i, (U^T U, X) \geq 0} \langle (G_2 + \zeta U^T U)^{-1}, X \rangle_1$$

$$\leq \min_{G_2 \in X \geq 0, (X, V_i) \leq R_i^2 \forall i, (U^T U, X) \geq 0} \langle (G_2 + \zeta U^T U)^{-1}, X \rangle_1$$

$$= \min_{G_2, \zeta, \theta, Z \geq 0, \nu \geq 0} \max_X \langle (G_2 + \zeta U^T U)^{-1} + Z + \theta U^T U, X \rangle_1 + \sum_{i=1}^N \nu_i (R_i^2 - (X, V_i)_1)$$

$$= \min_{G_2, \zeta, \theta, Z \geq 0, \nu \geq 0} \max_X \langle (G_2 + \zeta U^T U)^{-1} + Z + \theta U^T U - \sum_i \nu_i V_i, X \rangle_1 + \sum_i \nu_i R_i^2$$

$$= \min_{G_2, \zeta, Z \geq 0, \nu \geq 0, (G_2 + \zeta U^T U)^{-1} + Z + \theta U^T U - \sum_i \nu_i V_i = 0} \nu_i R_i^2$$

$$= \min_{G_2, \zeta, Z \geq 0, \nu \geq 0, Z = \sum_i \nu_i V_i - (G_2 + \zeta U^T U)^{-1} - \theta U^T U} \langle \nu, R^2 \rangle$$

$$= \min_{G_2, \zeta, \theta, Z \geq 0, \nu \geq 0, (G_2 + \zeta U^T U)^{-1} - \theta U^T U \geq 0} \langle \nu, R^2 \rangle$$

$$= \min_{G_2, \zeta, \theta \geq 0} (G_2 + \zeta U^T U)^{-1} \leq \sum_i \nu_i V_i - \theta U^T U \geq 0$$

$$= \min_{G_2, \zeta, \theta \geq 0, (G_2 + \zeta U^T U)^{-1} \leq \sum_i \nu_i V_i} \langle \nu, R^2 \rangle$$

$$= \min_{G_2, \zeta \geq 0, \nu \geq 0, (G_2 + \zeta U^T U)^{-1} \leq \text{diag}(\nu) \otimes I_n} \langle \nu, R^2 \rangle$$

$$= \min_{G_2, \zeta \geq 0, \nu \geq 0, G_2 + \zeta U^T U \geq \text{diag}(\nu^{-1}) \otimes I_n} \langle \nu, R^2 \rangle,$$

where $\nu = [\nu_1 \cdots \nu_N]^T$. Now, taking into account that $G_2 = \mathcal{L} \otimes I_n$, where we recall that $\mathcal{L} \in \mathcal{M}$, and $\zeta \geq 0$, we get:

$$\min_{\mathcal{L} \in \mathcal{M}, \zeta \geq 0, \nu \geq 0, (\mathcal{L} + \zeta e e^T) \otimes I_N \geq \text{diag}(\nu^{-1}) \otimes I_n} \langle \nu, R^2 \rangle = \min_{\mathcal{L} \in \mathcal{M}, \zeta \geq 0, \nu \geq 0, \mathcal{L} + \zeta e e^T \geq \text{diag}(\nu^{-1})} \langle \nu, R^2 \rangle.$$

Finally, the SDP (3.19) is obtained from Schur complement formula applied to the previous optimization problem. □

Since we assumed that the graph $\mathcal{G}$ is connected, we have that $\lambda_1(\mathcal{L}) = 0$ is simple and consequently $\lambda_2(\mathcal{L}) > 0$. Note that the following equivalence holds:

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

since the spectrum of the matrix $\mathcal{L} + \zeta e e^T$ is $\{\zeta \|e\|^2, \lambda_2(\mathcal{L}), \cdots, \lambda_N(\mathcal{L})\}$. It follows that $\zeta = \frac{t}{\|e\|^2}$, $\nu_i = \frac{1}{t}$ for all $i$, and $\mathcal{L}$ such that $t \leq \lambda_2(\mathcal{L})$ is feasible for the SDP problem (3.19). We conclude that:

$$\left(R^*(x_0)\right)^2 \leq \min_{\mathcal{L} \in \mathcal{M}, \zeta \geq 0, \nu \geq 0, \mathcal{L} + \zeta e e^T \geq \text{diag}(\nu^{-1})} \langle \nu, R^2 \rangle$$

$$\leq \min_{\mathcal{L} \in \mathcal{M}, \zeta \geq 0, \nu \geq 0, \lambda_2(\mathcal{L})} \sum_{i=1}^N R_i^2 \frac{1}{t} \leq \frac{\sum_i R_i^2}{\lambda_2(\mathcal{L})} \quad \forall \mathcal{L} \in \mathcal{M}.$$  

Then, according to Theorem 3.1 we obtain the following upper bound on the rate of convergence for the expected values of the objective function:

$$\phi_k - f^* \leq \frac{2 \sum_{i=1}^N R_i^2}{\lambda_2(\mathcal{L}) \cdot k} \quad \forall \mathcal{L} \in \mathcal{M}.$$
From the convergence rate for Algorithm (RCD) given in (3.21) it follows that we can choose the probabilities such that we maximize the second eigenvalue of $L$: \[ \max_{L \in \mathcal{M}} \lambda_2(L). \] In conclusion, in order to find some suboptimal probabilities \((p_{ij})_{(i,j) \in E}\), we can solve the following simpler SDP problem than the one given in (3.19):

\[
(3.22) \quad p_{ij}^* = \arg \max_{t, L \in \mathcal{M}} \left\{ t : L \succeq t \left( I_N - \frac{ee^T}{\|e\|^2} \right) \right\}.
\]

Note that the matrices on both sides of the LMI from (3.22) have the common eigenvalue zero associated to 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 this LMI with the following equivalent LMI:

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

If the probabilities are chosen as the optimal solution of the previous SDP problem (3.22), then from (3.21) we obtain the following upper bound on the rate of convergence for the expected values of the objective function:

\[
(3.23) \quad \phi_k - f^* \leq (N - 1) \frac{2 \sum_{i=1}^N \frac{1}{(N-1)} \lambda_2(L^i) R^2_i}{k}.
\]

Here $L^*$ denotes the optimal solution of the SDP (3.22). In conclusion, we have:

\[
(3.24) \quad \phi_k - f^* \leq \left( R^*(x^0) \right)^2 \leq \min_{p_{ij}} SDP(3.19) \leq \min_{p_{ij}} \frac{\sum_i R^2_i}{SDP(3.22)} \leq \sum_i \frac{R^2_i}{\lambda_2(L)} \quad \forall p_{ij}
\]

and consequently

\[
(3.25) \quad \phi_k - f^* \leq \frac{2 \left( R^*(x^0) \right)^2}{k} \leq \frac{2 \min_{p_{ij}} SDP(3.19)}{k} \leq (N - 1) \frac{2 \sum_{i=1}^N \frac{1}{(N-1)} \lambda_2(L^i) R^2_i}{k} \leq (N - 1) \frac{2 \sum_{i=1}^N \frac{1}{(N-1)} \lambda_2(L^i) R^2_i}{k} \quad \forall p_{ij}.
\]

Finally, we also have the following results:

1. If we assume complete graph and the probabilities are taken in the form (3.8), then the Laplacian matrix has the expression given in (3.9), i.e. $L_s = \frac{1}{(N-1) L_{av}} \left( I_N - \frac{1}{N} ee^T \right)$. For this matrix we can show immediately that

   \[ \lambda_2(L_s) = \frac{1}{(N-1)L_{av}}. \]

   In conclusion, we get the convergence rate (3.16), which shows that $R^2(x^0) \leq (N - 1) \sum_i L_{av} R^2_i = \sum_i \frac{R^2_i}{\lambda_2(L_{av})}$. 

2. If we assume a complete graph and the probabilities are taken in the form (3.11), then the Laplacian matrix has the expression given in (3.12), i.e. $L_{inv} = \frac{1}{(N-1) \sum_i L_i} \left( \frac{1}{\sum_i 1/L_i} \text{diag}(L^{-1}) - L^{-1} (L^{-1})^T \right)$. For this matrix we have that $R^2(x^0) \leq (N - 1) \sum_i L_i R^2_i \leq \sum_i \frac{R^2_i}{\lambda_2(L_{inv})}$. 

3.3. Comparison with the full projected gradient method. Based on Assumption 1 we can derive the following inequality:

\[
    f(x + s) \leq \sum_{i=1}^{N} f_i(x_i) + \langle \nabla f_{i}(x_{i}), s_{i} \rangle + \sum_{i=1}^{N} \frac{L_{i}}{2} \|s_{i}\|^2
\]

(3.25)

\[
    = f(x) + \langle \nabla f(x), s \rangle + \frac{1}{2} \|s\|^2_{\text{diag}(L)} \quad \forall x,s \in \mathbb{R}^{Nn}.
\]

Thus, we also have:

(3.26) \quad f(x + s) \leq f(x) + \langle \nabla f(x), s \rangle + \frac{\bar{L}}{2} \|s\|^2 \quad \forall x,s \in \mathbb{R}^{Nn},

where we recall that \(\bar{L} = \max_{i} L_{i}\). Therefore, if we measure distances in the extended space \(\mathbb{R}^{Nn}\) with the Euclidean norm, we can take \(\bar{L}\) as a Lipschitz constant for \(f\). Let us apply the full projected gradient method for solving the optimization problem (2.1). Given \(x \in S\), we define the following iteration:

\[
    x^{+} = x + d,
\]

where \(d\) is the optimal solution of the following optimization problem (see (3.26)):

\[
    d = \arg \min_{s \in \mathbb{R}^{Nn} : \sum_{i=1}^{N} s_{i} = 0} f(x) + \langle \nabla f(x), s \rangle + \frac{\bar{L}}{2} \|s\|^2.
\]

Since we assume local Euclidian norms on \(\mathbb{R}^{n}\), we obtain the following solution:

\[
    d_{i} = \frac{1}{N\bar{L}} \sum_{j=1}^{N} (\nabla f_{j}(x_{j}) - \nabla f_{i}(x_{i})) \quad \forall i \in V.
\]

In conclusion, if we consider the Euclidian norm in the extended space \(\mathbb{R}^{Nn}\), then from Assumption 1 it follows that the function \(f\) has Lipschitz continuous gradient with Lipschitz constant \(\bar{L}\) (according to (3.26)) and then the convergence rate of the projected gradient method is given by [11]:

(3.27) \quad f(x^{k}) - f^{*} \leq \frac{2\sum_{i=1}^{N} \bar{L}R_{i}^2}{k}.

In the sequel we show that we can consider another norm to measure distances in the subspace \(S\) from the extended space \(\mathbb{R}^{Nn}\) different from the Euclidian norm. We will see that with this norm the convergence rate of the projected gradient is better than that in (3.27), where the Euclidian norm was considered. Since \(f\) has Lipschitz continuous gradient and the descent lemma from (3.25) is valid, by standard reasoning we can argue that the direction \(d^{k}\) in the gradient method can be computed as (see (3.25)):

\[
    d = \arg \min_{s \in \mathbb{R}^{Nn} : \sum_{i=1}^{N} s_{i} = 0} \sum_{i=1}^{N} f_{i}(x_{i}) + \langle \nabla f_{i}(x_{i}), s_{i} \rangle + \frac{L_{i}}{2} \|s_{i}\|^2.
\]

We obtain the following closed form solution:

\[
    d_{i} = \frac{1}{L_{i}} \sum_{j=1}^{N} \frac{1}{L_{j}} (\nabla f_{j}(x_{j}) - \nabla f_{i}(x_{i})) \quad \sum_{j=1}^{N} \frac{1}{L_{j}} \quad \forall i \in V.
\]
From (3.25) we derive the following inequality:

\[
\begin{align*}
    f(x^+) &\leq f(x) - \sum_{i=1}^{N} \frac{1}{2L_i} \left( \frac{1}{\sum_{j=1}^{N} \frac{1}{L_j}} \right)^2 \\
    &\quad \times \sum_{j=1}^{N} \left( \nabla f_j(x_j) - \nabla f_i(x_i) \right)^2 \\
    &\quad + \frac{1}{2} \sum_{j=1}^{N} \frac{1}{L_j} \sum_{i=1}^{N} \frac{1}{L_i} \nabla f_j(x_j) \\
    &\quad \sum_{j=1}^{N} \frac{1}{L_j},
\end{align*}
\]

where \( G_N = L_N \otimes I_n \) and the matrix \( L_N \) is defined as

\[
L_N = \text{diag}(L^{-1}) - \frac{1}{\sum_{i=1}^{N} 1/L_i} L^{-1}(L^{-1})^T,
\]

where we recall that \( L = [L_1 \cdots L_N]^T \). Note that \( L_N \) is still a Laplacian matrix but for a complete graph with \( N \) nodes. As in the previous section, for the matrix \( G_N \) we define the induced norms in the extended primal and dual space:

\[
\|u\|_{G_N} = \sqrt{u^T G_N u}, \quad \|x\|_{G_N}^* = \max_{u: \langle G_N u, u \rangle \leq 1} \langle x, u \rangle \quad \forall x \in S, u \in \mathbb{R}^{Nn} \setminus T.
\]

Based on (3.12) and (3.13) we conclude that

\[
(||x||_{G_N}^*)^2 = \sum_{i=1}^{N} L_i \|x_i\|^2 \quad \forall x \in S.
\]

The full projected gradient iteration at each step \( k \) becomes:

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

Following the same reasoning as in Theorem 3.1, we obtain the following rate of convergence:

\[
f(x^k) - f^* \leq \frac{2 R_{\text{full}}^2(x_0)}{k},
\]

where \( R_{\text{full}}(x_0) = \max_{x: f(x) = f(x_0) \in X} \max_{x^* \in X^*} \|x - x^*\|_{G_N}^* \). Using the expression for the norm \( \|x\|_{G_N}^* \) given in (3.28) and the definition for \( R_i \), we can show that

\[
R_{\text{full}}^2(x_0) \leq \sum_{i=1}^{N} L_i R_i^2,
\]

and thus we get the following convergence rate for the full projected gradient method, when the extended norm (3.28) is considered:

\[
f(x^k) - f^* \leq \frac{2 \sum_{i=1}^{N} L_i R_i^2}{k}.
\]
Moreover, the complexity per iteration of the full gradient is

\[ O(Nn_f + Nn) \]

i.e. \( O(N) \) times more than for Algorithm RCD.

Clearly, the estimate given in (3.30) (where the induced norm defined by the matrix \( G_{\text{full}} \) is considered) is better than the estimate given in (3.27) (where the Euclidian norm is considered). Further, the iteration complexity of the full projected gradient method is \( O\left(N(n + n_f)\right) \). Note that if \( \sum_{i=1}^{N} L_i R_i^2 \leq \sum_{i=1}^{N} L_i R_i^2 \), then the rate of convergence for Algorithm (RCD) is as follows:

\[
\phi_k - f^* \leq \frac{2 \left(R^* (x^0)\right)^2}{k}
\]

\[
p_{ij}^* \leq \frac{2 \sum_{i=1}^{N} L_i R_i^2}{(N - 1)k}
\]

\[
p_{ij}^* \leq \frac{2 \sum_{i=1}^{N} L_i R_i^2}{(N - 1)k}
\]

\[
p_{ij}^{\text{inv}} \leq \frac{2 \sum_{i=1}^{N} L_i R_i^2}{(N - 1)k}
\]

\[
\text{Table 3.1 Comparison of arithmetic complexities for algorithms (RCD), full gradient, [2] and [19] for } n = 1.
\]

<table>
<thead>
<tr>
<th>Alg.</th>
<th>block</th>
<th>iteration</th>
<th>Rate of conv.</th>
<th>Iter. complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full grad.</td>
<td>yes</td>
<td>full</td>
<td>( O \left( \sum_{i=1}^{N} L_i R_i^2 \right) )</td>
<td>( O(Nn_f + N) )</td>
</tr>
<tr>
<td>(RCD)/(p_{ij}^{\text{inv}})</td>
<td>yes</td>
<td>random (i, j)</td>
<td>( O \left( (N - 1) \sum_{i=1}^{N} L_i R_i^2 \right) )</td>
<td>( O(n_f + \ln N) )</td>
</tr>
<tr>
<td>(RCD)/(p_{ij}^*)</td>
<td>yes</td>
<td>random (i, j)</td>
<td>( O \left( (N - 1) \sum_{i=1}^{N} L_i R_i^2 \right) )</td>
<td>( O(n_f + \ln N) )</td>
</tr>
<tr>
<td>Tseng [19]</td>
<td>yes</td>
<td>greedy (i, j)</td>
<td>( O \left( N \sum_{i=1}^{N} L_i R_i^2 \right) )</td>
<td>( O(n_f + N) )</td>
</tr>
<tr>
<td>Beck [2]</td>
<td>no</td>
<td>greedy (i, j)</td>
<td>( O \left( N \sum_{i=1}^{N} L_i R_i^2 \right) )</td>
<td>( O(n_f + N) )</td>
</tr>
</tbody>
</table>

However, the iteration complexity of (RCD) method is usually \( O(N) \) times cheaper than the iteration complexity of the full projected gradient algorithm. Moreover, the full projected gradient method is not a distributed algorithm since it requires a central coordinator. Note that despite the fact that the coordinate descent methods presented in [2, 19] can solve optimization problems with additional box constraints, the arithmetic complexity of those methods are \( O(N) \) times worse than the arithmetic complexity of our Algorithm (RCD). This can be seen in Table 4.1 where we compare the arithmetic complexities off all these four algorithms (full gradient, our method RCD, the coordinate descent method in [19] and the coordinate descent method in [2]) for optimization problems with \( n = 1 \) (scalar case) and a single linear coupled constraint (recall that \( \bar{L} = \max_i L_i \)). Finally, note that the method in [19] has very bad rate of convergence when the number of coupling constraints is larger than one, while the method in [2] is not able to handle more than one single coupling constraint.
3.4. Strongly convex case. Additionally to the assumption of Lipschitz continuous gradient for each function \( f_i \) (see Assumption (1)), we now assume that the function \( f \) is also strongly convex with respect to the extended norm \( \| \cdot \|_{G_2}^* \) with convexity parameter \( \sigma_{G_2} \) on the subspace \( S \). More exactly, the objective function \( f \) satisfies:

\[
f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\sigma_{G_2}}{2} \left( \| x - y \|_{G_2}^* \right)^2 \quad \forall x, y \in S.
\]

Combining the Lipschtiz inequality (3.25) with the previous strong convex inequality (3.33) we get:

\[
\sum_{i=1}^{N} L_i \| x_i - y_i \|^2 \geq \sigma_{G_2} \left( \| x - y \|_{G_2}^* \right)^2 \quad \forall x, y \in S.
\]

Now, if we consider e.g. a full graph and the probabilities given in (3.11), then using the expression for the norm \( \| \cdot \|_{G_2}^* \) given in (3.13) we obtain \( \sigma_{G_2} \leq \frac{1}{N-1} \).

We now state the main result of this section:

**Theorem 3.3.** Under the assumptions of Theorem 3.1, let function \( f \) be also strongly convex with respect to norm \( \| \cdot \|_{G_2}^* \) and convexity parameter \( \sigma_{G_2} \). For the sequence \( (x^k)_{k \geq 0} \) generated by Algorithm (RCD) we have the following linear estimate for the convergence rate in expectation:

\[
\phi_k - f^* \leq (1 - \sigma_{G_2})^k \left( f(x^0) - f^* \right).
\]

**Proof.** From (3.5) we have

\[
2 \left( f(x^k) - E \left[ f(x^{k+1}) \mid \eta^k \right] \right) \geq \| \nabla f(x^k) \|^2_{G_2}.
\]

On the other hand, minimizing both sides of inequality (3.33) over \( x \in S \) we have:

\[
\| \nabla f(y) \|^2_{G_2} \geq 2\sigma_{G_2} (f(y) - f^*) \quad \forall y \in S
\]

and for \( y = x^k \) we get:

\[
\| \nabla f(x^k) \|^2_{G_2} \geq 2\sigma_{G_2} (f(x^k) - f^*)
\]

Combining these two relations and taking expectation in \( \eta_{k-1} \) in both sides, we prove the statement of the theorem. \( \Box \)

We notice that if \( f_i \)‘s are strongly convex functions with respect to the Euclidian norm, with convexity parameter \( \sigma_i \), i.e.

\[
f_i(x_i) \geq f_i(y_i) + \langle \nabla f_i(y_i), x_i - y_i \rangle + \frac{\sigma_i}{2} \| x_i - y_i \|^2 \quad \forall x_i, y_i \in \mathbb{R}^n, \quad i \in V,
\]

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 \( \Sigma = \text{diag}(\sigma) \), where \( \sigma = [\sigma_1 \cdots \sigma_N]^T \), i.e.

\[
f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} \| x - y \|^2_{M \otimes I_n} \quad \forall x, y \in \mathbb{R}^{nN}.
\]

Note that in this extended norm \( \| \cdot \|_{M \otimes I_n} \) the strongly convex 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 \( \| \cdot \|_{G_2}^* \) with the strongly convex parameter \( \sigma_{G_2} \) satisfying:

\[
\sigma_{G_2} \Sigma^{-1} \preceq \mathbf{L} + \zeta \mathbf{e} \mathbf{e}^T,
\]
for some $\zeta \geq 0$. In conclusion we get the following LMI:

$$\sigma_{G_2} I_N \preceq \Sigma^{1/2} (L + \zeta e e^T) \Sigma^{1/2}.$$  

From Theorem 3.3 it follows that in order to get a better convergence rate we need to search for $\sigma_G$ as large as possible. Therefore, for strongly convex functions the optimal probabilities are chosen as the solution of the following SDP problem:

$$p^*_i = \arg \max_{\zeta \geq 0, \sigma_{G_2} \in \mathcal{M}} \sigma_{G_2}$$

$$\sigma_{G_2} I_N \preceq \Sigma^{1/2} (L + \zeta e e^T) \Sigma^{1/2}.$$  

3.5 Rate of convergence in probability. In this section we estimate the quality of random point $x^k$, quantifying the confidence of reaching the accuracy $\epsilon$ for the optimal value. We denote by $\rho$ a confidence level and we use Theorem 1 from [17] to give the following lemma:

**Lemma 3.4.** Let $\{\xi_k\}_{k \geq 0}$ be a nonnegative, nonincreasing sequence of discrete random variables with one of the properties:

1. $E[\xi_{k+1}|\xi_k] \leq \xi_k - \frac{\xi_0}{2}$ for $k \geq 0$ and constant $r > 1$.
2. $E[\xi_{k+1}|\xi_k] \leq (1 - \frac{1}{r}) \xi_k$ for $k \geq 0$ and constant $r > 1$.

If the first part of the lemma holds and choose $K$ such that

$$K \geq \frac{r}{\epsilon} \left( 1 + \ln \frac{1}{\rho} \right) + \frac{r}{\xi_0} + 2$$

or if the second part holds and choose $K$ satisfying

$$K \geq r \ln \frac{\xi_0}{\epsilon \rho},$$

then we have the following probability $\text{Prob}(\xi_K \leq \epsilon) \geq 1 - \rho$.

**Proof.** The proof use a similar reasoning as Theorem 1 in [17] and is derived from Markov inequality.

Considering now the sequence of random variable $\xi_k = f(x^k) - f^*$ in the previous lemma, we reach the following result:

**Theorem 3.5.** Under the assumptions of Theorem 3.1, let us choose

$$k \geq \frac{R^2(x^0)}{\epsilon} \left( 1 + \ln \frac{1}{\rho} - \frac{\epsilon}{f(x^0) - f^*} \right) + 2,$$

then the random sequence $(x^k)_{k \geq 0}$ generated by Algorithm (RCD) for solving (2.1) satisfies:

$$\text{Prob}(f(x^k) - f^* \leq \epsilon) \geq 1 - \rho.$$  

If additionally, the function $f$ is also strongly convex with respect to norm $\| \cdot \|_{G_2}^*$ and convexity parameter $\sigma_{G_2}$, choosing $k$ to satisfy:

$$k \geq \frac{1}{\sigma_{G_2}} \ln \frac{f(x^0) - f^*}{\epsilon \rho},$$

it is ensured that:

$$\text{Prob}(f(x^k) - f^* \leq \epsilon) \geq 1 - \rho.$$
Proof. From inequality (3.15) we note that random variable $\xi_k = f(x^k) - f^*$ has the property: $\xi_{k+1} \leq \xi_k - \frac{\xi_k}{r}$, where we assume $r = R^2(x^0) > \epsilon$. Note that $\xi_k$ satisfies the first conditions of Lemma 3.4, thus we get the first statement of the theorem.

For the second part, from Markov inequality and relation (3.34) we have:

$$\text{Prob}(f(x^k) - f^* \geq \epsilon) \leq \frac{\phi_k - f^*}{\epsilon} \leq \frac{1}{\epsilon} (1 - \sigma G_k)^k (f(x^0) - f^*)$$

Choosing $k$ as in the statement of the theorem we have that $\text{Prob}(f(x^k) - f^* \geq \epsilon) \leq \rho$, so that the second statement of the theorem is now proved. \[\Box\]

4. Generalization. Parallel implementations of Algorithm (RCD) are possible, i.e. we can choose in parallel different block coordinates and update $x$ in all these components. Usually, Algorithm (RCD) can be accelerated by parallelization, i.e. by using more than one pair per iteration. In this section we extend the main results of the previous sections to general randomized block coordinate descent algorithm that chooses an $M$ -tuple of indices. The $(RCD)_{(i,j)}$ algorithm can be extended to update more than two pairs $(i,j)$ at each iteration. For example, taking a positive integer $M \leq N$, we denote with $N$ any subset of $V$ having cardinality $M$. Here, we do not assume explicitly additional structure such as those imposed by a graph, i.e. we consider all to all communication. Then, we can derive a randomized $M$ block coordinate descent algorithm where we update at each iteration only $M$ blocks in the vector $x$. Let us define $N = (i1, \ldots, iM)$ with $i^t \in V$, $s_N = [s_{i1}, \ldots, s_{iM}]^T \in \mathbb{R}^{Mn}$ and $L_N = [L_{i1}, \ldots, L_{iM}]^T \in \mathbb{R}^M$. Under assumption (2.2) the following inequality holds:

\begin{equation}
\tag{4.1}
f(x + \sum_{i \in N} U_i s_i) \leq f(x) + \langle \nabla f_N(x), s_N \rangle + \frac{1}{2} \|s_N\|_\text{diag}(L_N)^2.
\end{equation}

Based on the inequality (4.1) we can define a general randomized $M$ block coordinate descent algorithm, let us call it $(RCD)_M$. Given an $x$ in the feasible set $S$, we choose the coordinate $M$-tuple $N$ with probability $p_N$. Let the next iterate be chosen as follows:

$$x^+ = x + \sum_{i \in N} U_i d_i,$$

i.e. we update $M$ components in the vector $x$, where the direction $d_N$ is determined by requiring that the next iterate $x^+$ to be also feasible and minimizing the right hand side in (4.1), i.e.:

$$d_N = \arg \min_{s_N: \sum_{i \in N} s_i = 0} f(x) + \langle \nabla f_N(x), s_N \rangle + \frac{1}{2} \|s_N\|^2_\text{diag}(L_N)$$

or explicitly

$$d_i = \frac{1}{L_i} \sum_{j \in N} \frac{1}{\ell_j} (\nabla f_j(x_j) - \nabla f_i(x_i)) \quad \forall i \in N.$$
Based on the inequality (4.1) the following decrease in objective function can be derived:

\[
f(x^+) \leq f(x) - \sum_{i \in \mathcal{N}} \frac{1}{2L_i} \left| \sum_{j \in \mathcal{N}} \frac{1}{L_j} (\nabla f_j(x_j) - \nabla f_i(x_i)) \right|^2 \\
= f(x) - \frac{1}{2} \nabla f(x)^T G_\mathcal{N} \nabla f(x),
\]

where \( G_\mathcal{N} = \mathcal{L}_\mathcal{N} \otimes I_\alpha \) and the matrix \( \mathcal{L}_\mathcal{N} \in \mathbb{R}^{N \times N} \) is defined as

\[
\mathcal{L}_\mathcal{N} = \text{diag}(L^{-1}_\mathcal{N}) - \sum_{i \in \mathcal{N}} 1/L_i L^{-1}_\mathcal{N}(L^{-1}_\mathcal{N})^T,
\]

where we redefine \( L_\mathcal{N} \in \mathbb{R}^N \) as the vector with components zero outside the index set \( \mathcal{N} \) and components \( L_i \) for \( i \in \mathcal{N} \). Note that \( \mathcal{L}_\mathcal{N} \) is still a Laplacian matrix. Therefore, taking the expectation over the random \( M \)-tuple \( \mathcal{N} \subset V \), we obtain the following inequality:

\[
(4.2) \quad E[f(x^+) \mid \eta] \leq f(x) - \frac{1}{2} \nabla f(x)^T G_M \nabla f(x).
\]

The corresponding matrix \( G_M = \sum_{\mathcal{N}} p_\mathcal{N} G_\mathcal{N} \) is still positive semidefinite and has an eigenvalue \( \lambda_1(G_M) = 0 \). Based on the decrease in expectation (4.2), a similar rate of convergence will be obtained for this general algorithm \((RCD)_M\) as in the previous sections, but depending on \( M \). We can consider probabilities of the form:

\[
p^\text{inv}_\mathcal{N} = \frac{\sum_{i \in \mathcal{N}} 1/L_i}{\sum_{i \in \mathcal{N}} 1/L_i}.
\]

We can see that:

\[
\Sigma_p = \sum_{\mathcal{N}} \sum_{i \in \mathcal{N}} 1/L_i = \sum_{j=1}^{(M)} \sum_{i \in \mathcal{N}_j} 1/L_i = \sum_{j=1}^{(M)} \sum_{i=1}^{N} 1_{i,j_N} 1/L_i \\
= \sum_{i=1}^{N} 1/L_i \left( \sum_{j=1}^{(M)} 1_{i,j_N} \right) = \sum_{i=1}^{N} 1/L_i \binom{M}{N-1}.
\]

Using a similar reasoning we can derive that:

\[
\mathcal{L}_M = \sum_{\mathcal{N}} p_\mathcal{N} \mathcal{L}_\mathcal{N} = \frac{1}{\Sigma_p} \sum_{\mathcal{N}} \left( \sum_{i \in \mathcal{N}} 1/L_i \right) \text{diag}(L^{-1}_\mathcal{N}) - L^{-1}_\mathcal{N}(L^{-1}_\mathcal{N})^T \\
= \frac{1}{\Sigma_p} \left( \binom{M}{N-2} \sum_{i=1}^{N} 1/L_i \right) \text{diag}(L^{-1}) - L^{-1}(L^{-1})^T \\
= \frac{M-1}{N-1} \left[ \text{diag}(L^{-1}) - \frac{1}{\sum_{i=1}^{N} 1/L_i} L^{-1}(L^{-1})^T \right].
\]
In conclusion, for this choice of the probabilities we get the following convergence rate for Algorithm \((RCD)_M\) in the expected values of the objective function:

\[
\phi_k - f^* \leq \frac{N - 1}{M - 1} \cdot \frac{2 \sum_i L_i R_i^2}{k}.
\]

5. Applications. Problem (2.1) arises in many real applications, e.g. resource allocation in economic systems [8] or distributed computer systems [9], in distributed control [14], in traffic equilibrium problems or network flow [1] and other areas. For example, we can interpret it 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 exchange and \(|x_i|_j < 0\) means that agent \(i\) contributes \(|x_i|_j\) of good \(j\) to exchange.

Problem (2.1) can also be seen as the dual problem corresponding to an optimization of a sum of convex functions. Consider the following convex optimization problem that arises in many engineering applications such as signal processing, distributed control and network flow [1, 4–6, 13–15]:

\[
g^* = \min_{v \in \cap_{i=1}^N Q_i} \sum_{i=1}^N g_i(v),
\]

where \(g_i\) are all convex functions and \(Q_i\) are convex sets. This problem can be reformulated as:

\[
\min_{u_i \in Q_i, u_i = v \forall i \in V} \sum_{i=1}^N g_i(u_1) + \cdots + g_N(u_N).
\]

Let us define \(u = [u_1^T \cdots u_N^T]^T\) and \(g(u) = g_1(u_1) + \cdots + g_N(u_N)\). By duality, using the Lagrange multipliers \(x_i\) for the constraints \(u_i = v\), we obtain the separable convex problem (2.1), where \(f_i(x_i) = \tilde{g}_i^*(x_i)\) and \(\tilde{g}_i^*\) is the convex conjugate of the function \(\tilde{g}_i = g_i + 1_{Q_i}\). Note that if \(g_i\) is strongly convex, then the convex conjugate \(f_i\) is well-defined and has Lipschitz continuous gradient, so that Assumption 1 holds.

A particular application is the problem of finding the projection of a point \(v_0\) in the intersection of the convex sets \(\cap_{i=1}^N Q_i\). This problem can be written as an optimization problem in the form:

\[
\min_{v \in \cap_{i=1}^N Q_i} \sum_{i=1}^N p_i \|v - v_0\|^2,
\]

where \(p_i > 0\) such that \(\sum_i p_i = 1\). This is a particular case of the separable problem (5.1). Note that since the functions \(g_i(v) = p_i \|v - v_0\|^2\) are strongly convex, then \(f_i\) have Lipschitz continuous gradient with Lipschitz constants \(L_i = 1/p_i\) for all \(i\). We can also consider the problem of finding a point in the intersection of some convex sets:

\[
\min_{v \in \cap_{i=1}^N Q_i} c_i^T v + \cdots + c_N^T v,
\]

which again applying duality as above, it will lead to the separable convex problem (2.1). Since in this case the objective functions \(g_i(v) = c_i^T v\) are linear, it follows that the functions \(f_i\) are not smooth anymore (i.e. Assumption 1 will not hold in this case) and smoothing techniques need to be applied. In this scenario we can consider smoothing (5.2) as follows:

\[
\min_{v \in \cap_{i=1}^N Q_i} \sum_{i=1}^N (c_i^T v + p_i \|v\|^2) + \cdots + (c_N^T v + p_N \|v\|^2).
\]
5.1. Recovering approximate primal solutions from full gradient algorithm. From Section 3.3 we have obtain the following convergence rate for the full projected gradient method:

\[ f(x^k) - f^* \leq \frac{2 \sum_i L_i R_i^2}{k} \quad \forall k > 0. \]

We also want to provide estimates on the convergence rate of the primal sequence \( \{u^k\}_{k \geq 0} \). Let us apply 2k iterations of the full projected gradient method. From the previous discussion we have that the following inequality holds:

\[ f(x^l+1) \leq f(x^l) - \frac{1}{2} \|\nabla f(x^l)\|^2_{G_N} \quad \forall l = k, \ldots, 2k - 1. \]

Adding these inequalities for \( l = k, \ldots, 2k - 1 \) we obtain:

\[ f(x^k) - f(x^{2k}) \geq \frac{1}{2} \sum_{i=k}^{2k-1} \|\nabla f(x^i)\|^2_{G_N} \geq \frac{k}{2} \|\nabla f(x^k)\|^2_{G_N}, \]

where

\[ k^* = \arg \min_{l = k, \ldots, 2k-1} \|\nabla f(x^l)\|^2_{G_N}. \]

Taking into account that after \( k \) steps we have sublinear convergence in the form \( O\left(\frac{2 \sum_i L_i R_i^2}{k}\right) \), we get:

\[ \|\nabla f(x^k)\|^2_{G_N} \leq \frac{2(f(x^k) - f(x^{2k}))}{k} \leq \frac{2(f(x^k) - f^*)}{k} \leq \frac{4 \sum_i L_i R_i^2}{k^2} \]

\[ f(x^{k^*}) - f^* \leq f(x^k) - f^* - \sum_{i=k}^{k^*-1} \frac{1}{2} \|\nabla f(x^i)\|^2_{G_N} \leq \frac{2 \sum_i L_i R_i^2}{k}. \]

Further, we have that:

\[ \|\nabla f(x^{k^*})\|^2_{G_N} = \sum_{i=1}^{N} \frac{1}{2L_i} \left\| \nabla f_i(x^{k^*}) - \sum_{j=1}^{N} \frac{1}{L_j} \nabla f_j(x^{k^*}) \right\|^2 = \sum_{i=1}^{N} \frac{1}{2L_i} \left\| u_i^{k^*} - \sum_{j=1}^{N} \frac{1}{L_j} u_j^{k^*} \right\|^2. \]

Now, if we define \( v^{k^*} = \frac{\sum_{i=1}^{N} \frac{1}{L_i} u_i^{k^*}}{\sum_{j=1}^{N} \frac{1}{L_j}} \), then the full projected gradient method produces after 2k iterations the dual variables \( x^{k^*} \) with \( \sum_i x_i^{k^*} = 0 \) and primal variables \( u_i^{k^*} \in Q_i \) for all \( i \in V \) such that with their convex combination \( v^{k^*} \) satisfies the following dual suboptimality:

\[ (5.3) \quad f(x^{k^*}) - f^* \leq \frac{2 \sum_i L_i R_i^2}{k}. \]

and primal feasibility violation:

\[ (5.4) \quad \|u_j^{k^*} - v^{k^*}\|^2 \leq \frac{8L_j (\sum_{i=1}^{N} L_i R_i^2)}{k^2} \quad \forall j \in V. \]
5.2. Numerical experiments.

5.2.1. First application. In this section we perform simulations for solving the optimization problem \((2.1)\) where the functions \(f_i\) are taken as in [20]:

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

where the coefficients \(a_i \geq 0, b_i, c_i, d_i\) are generated randomly with uniform distributions on \([-2, 2]\). We also generate random graphs with \(N\) nodes, where each node has at most \(p\) neighbors. Note that \(\sigma_i = a_i\) and \(L_i = a_i + \frac{1}{4}b_i^2\).

5.2.2. Second application. In this section we consider the following application that is a particular case of optimization problem \((5.1)\):

\[
(5.6) \quad \min_{v \in \cap_{i=1}^N Q_i} p_1 \|v - a_1\|^2 + \cdots + p_N \|v - a_N\|^2,
\]

where \(a_i\) are given vectors in \(\mathbb{R}^n\), \(Q_i\)’s are simple convex sets and the weights \(p_i > 0\) such that \(\sum_i p_i = 1\). We also associate a local information structure to our problem imposed by a connected graph \(G = (V, E)\). The application consists of finding the closest point from the intersection of some given sets \(Q_i\) to the average \(c = \frac{1}{N} \sum_i p_i a_i\) in a distributed fashion. Let us note that for \(Q_i = \mathbb{R}^n\) the problem reduces at computing distributively the average \(\sum_i p_i a_i\) that has many applications in engineering [15, 21], while for \(a_i = 0\) for all \(i\) the problem can be interpreted as finding a point in common to the given convex sets \(Q_i\)’s [4-6, 15]. We solve the primal problem \((5.6)\) via its dual formulation \((2.1)\). Note that the gradient of each function \(f_i\) can be efficiently computed as long as the projection on the set \(Q_i\) is easy and further the gradient is Lipschitz continuous with constant \(L_i = \frac{1}{F_i}\).

REFERENCES


