2023 European Control Conference (ECC)
June 13-16, 2023. Bucharest, Romania

On the Worst-Case Analysis of Cyclic Coordinate-Wise Algorithms on Smooth Convex Functions

Yassine Kamri, Julien M. Hendrickx, François Glineur
ICTEAM, UCLouvain
Louvain-La-Neuve, B-1348, Belgium
{yassine.kamri,julien.hendrickx,francois.glineur}@uclouvain.be

***Abstract*— We propose a unifying framework for the automated computer-assisted worst-case analysis of cyclic block coordinate algorithms (BCD) in the unconstrained smooth convex optimization setup. We compute exact worst-case bounds for the cyclic coordinate descent and the alternating minimization algorithms over the class of smooth convex functions, and provide sublinear upper and lower bounds on the worst-case rate for the standard class of functions with coordinate-wise Lipschitz gradients. We obtain in particular a new upper bound for cyclic coordinate descent that outperforms the best available ones by an order of magnitude. We also demonstrate the flexibility of our approach by providing new numerical bounds using simpler and more natural assumptions than those normally made for the analysis of block coordinate algorithms. Finally, we provide numerical evidence for the fact that a standard scheme that provably accelerates random coordinate descent to a $O(1/k^2)$ complexity is actually inefficient when used in a (deterministic) cyclic algorithm.**

I. Introduction

Large-scale optimization problems are the cornerstone of many engineering applications, such as in machine learning or signal processing. With the widespread availability of data, the scale of some of these problems is constantly increasing, to the point where standard full-gradient optimization methods are often becoming computationally too expensive. Fortunately, many of these problems possess a structure that allows the use of *partial gradient* methods. An important subclass of these methods are the block coordinated descent algorithms, which only need access to a subset of gradient coordinates at each iteration. These can generally be separated in three main categories depending on how the blocks of coordinates are selected and updated [1], [2]: (i) Gauss-Southwell methods greedily select the coordinates that lead to the largest improvements (i.e the coordinates with largest gradient norm), (ii) randomized methods select coordinates according to a probability distribution and (iii) cyclic methods update the coordinates in a cyclic predefined order. Although greedy methods can exhibit good performances, their update rule typically requires having access to full gradients. Hence randomized and cyclic methods have been more heavily used and studied.

The theoretical convergence analysis of random coordinate descents has proven easier than the analysis of their deterministic counterparts. Sampling coordinates with replacement from a suitable probability distribution implies indeed that the expectation of each coordinate step is the full gradient, making the analysis mostly similar to that of full gradient descent. Consequently, many random coordinate descent algorithms with theoretical guarantees have been proposed for convex optimization problems, including accelerated and proximal variants for a variety of probability distributions [3]–[9]. However, these performances are only guaranteed in expectation or with high probability, and the sampling technique can be computationally costly. Hence there is also an interest for cyclic block coordinate methods, which appear simpler and more efficient to implement in practice, but much harder to analyse. The difficulty appears to reside in establishing a link between the block of coordinates at each step and the full gradient. Some convergence results are already known for cyclic coordinate descent, but they are often obtained under quite restrictive assumptions such as the isotonicity of the gradient [10], or with worst-case bounds that are relatively conservative and under initial conditions that are not entirely standard for unconstrained convex optimization [11]. Better convergence results exist for the specific case of quadratic optimization problems [12]–[17].

In this paper, we propose an alternative analysis based on Performance Estimation Problems (PEP). The underlying idea of PEP, which is to compute performance guarantees for first-order methods thanks to semidefinite programming (SDP) originated in [18]. It was further developed in [19] where the authors used convex interpolation to derive tight results and provide examples of worst-case functions for various types of first-order methods. A related approach was proposed in [20] where worst-case convergence analysis is performed through the lens of control theory and finding Lyapunov functions. As such, the worst-case guarantees rely on smaller SDPs, but are only asymptotic.

Contributions. Our main contributions are as follows

- By extending the PEP framework to first-order block coordinate algorithms, we provide a unifying framework for the analysis of such algorithms. We illustrate the flexibility of our framework by analysing three variants of block coordinate algorithms: block coordinate descent, alternating minimization and a cyclic version of the random accelerated block coordinate descent algorithm from [5].
- We compute the exact worst-case convergence rate of block coordinate algorithms over the class of smooth

978-3-907144-08-4 ©2023 EUCA

convex functions for a range of optimization setups. Furthermore, we provide sublinear upper and lower bound valid for the class of functions with coordinate-wise Lipschitz gradients, which is frequently considered when analysing such methods. For the block coordinate descent algorithm, our bounds outperforms the best known bound from [11] by an order of magnitude. For the alternating minimization algorithm our bounds suggest that the bound from [11] is asymptotically tight.
- We provide numerical worst-case convergence analysis for cyclic block coordinate algorithms under more natural and less restrictive assumptions than those usually made for the analysis of cyclic block coordinate algorithms [3], [11].
- We show that acceleration schemes for random block-wise algorithms do not necessarily generalize to cyclic variants; we provide indeed a numerical lower bound on the convergence rate of the latter that suggests it is asymptotically worse than that of the original random version.

Related work. As mentioned above, cyclic block coordinate descent has attracted less attention than its random counterparts. Although a convergence rate have been established in [11] for the unconstrained smooth convex minimization setting, it is quite conservative and the assumptions made on the first iterate are not very practical.

Relying on the performance estimation problem to establish worst-case convergence bounds for cyclic coordinate descent has been attempted in [21]. Although the bound they obtain is significantly better then the bound in [11], it is established under very restrictive assumptions such as the equality of the dimensions of the blocks. In [22], an asymptotic worst-case convergence bound (not guaranteed to be tight) is established for random coordinate descent using Lyapunov functions.

II. Block coordinate algorithms

We consider the general unconstrained minimization setup

$$\min_{x\in\mathcal{R}^d} f(x), \tag{1}$$

where the function f is convex, differentiable and defined over the entire space $\mathcal{R}^d$. We assume that the optimal set X^* is non-empty and select an arbitrary optimal point $x^{(*)} \in X^*$. As we will analyze block-coordinate algorithms, we further consider a partition of the space $\mathcal{R}^d$ into p subspaces

$$\mathcal{R}^d = \mathcal{R}^{d_1} \times ..... \times \mathcal{R}^{d_p}, \tag{2}$$

and introduce selection matrices $U_i \in \mathcal{R}^{n\times n_i}$ such that

$$(U_1, \ldots, U_p) = \mathcal{I}_n,$$

allowing to write for every $x \in \mathcal{R}^d$

$$x = (x_1; \ldots; x_p) \text{ with } x_i = U_i^T x \in \mathcal{R}^{d_i}\ \forall i \in 1, \ldots, p. \tag{3}$$

If x is given as in (3), we also have that $x = \sum_{i=1}^p U_i x_i$.

Definition 2.1: We define the partial gradient of f in x_i by

$$\nabla_i f(x) \triangleq U_i^T \nabla f(x). \tag{4}$$

We now present three specific cyclic block-coordinate algorithms that we will analyse, though our approach can be directly applied to a wide class of methods. First the cyclic block coordinate descent (CCD) performs at each iteration a gradient step with respect to a block of variables chosen in a cyclic order. Note that K denotes the number of cycles,

Algorithm CCD Cyclic coordinate descent

Inputs: function f defined over $\mathcal{R}^d$ with p blocks, starting point $x^{(0)} \in \mathcal{R}^d$, number of cycles K and step-size α.
Define $N = pK$. For $n = 1 \ldots N$:
Set $i = \text{mod}(n, p) + 1$
$x^{(n)} = x^{(n-1)} - \alpha U_i \nabla_i f(x^{(n-1)})$
Output $x^{(N)}$

where each block of coordinate is updated once in a predefined order, and p the number of blocks. Thus the number of partial gradient steps in Algorithm (CCD) is $N = pK$.

We also consider cyclic alternating minimization (CAM) where at each step we perform an exact minimization along a chosen block of coordinates instead of using a partial gradient step. Finally, we also analyse a deterministic cyclic version

Algorithm CAM Cyclic alternating minimization

Inputs: function f defined over $\mathcal{R}^d$ with p blocks, starting point $x^{(0)} \in \mathcal{R}^d$, number of cycles K.
Define $N = pK$. For $n = 1 \ldots N$:
Set $i = \text{mod}(n, p) + 1$
$x^{(n)} = \arg\min_{z = x^{(n-1)} + U_i \Delta x_i,\ \Delta x_i \in \mathcal{R}^{d_i}} f(z)$
Output $x^{(N)}$

of the accelerated random coordinate descent algorithm from [5], which we denote (CACD).

Algorithm CACD Cyclic accelerated coordinate descent

Inputs: function f defined over $\mathcal{R}^d$ with p blocks, starting point $x^{(0)} \in \mathcal{R}^d$, number of cycles K, step parameter α.
Define $N = pK$, $z^{(0)} = x^{(0)}$ and $\theta_0 = \frac{1}{p}$. For $n = 1 \ldots N$:
Set $i = \text{mod}(n, p) + 1$
$y^{(n-1)} = (1 - \theta_n)x^{(n-1)} + \theta_{n-1} z^{(n-1)}$
$z^{(n)} = z^{(n-1)} - \frac{\alpha}{p\theta_n} U_i \nabla_i f(y^{(n-1)})$
$x^{(n)} = y^{(n-1)} + p\theta_{n-1}(z^{(n)} - z^{(n-1)})$
$\theta_n = \frac{\sqrt{\theta_{n-1}^4 + 4\theta_{n-1}^2} - \theta_{n-1}^2}{2}$
Output $x^{(N)}$

III. The class of convex smooth functions and convex smooth interpolation

In the sequel, We will consider two classes of functions in our worst-case analysis.

Definition 3.1: Given a positive constant L, a function f defined over $\mathcal{R}^d$ is L-smooth if and only if

$$\forall x \in \mathcal{R}^d,\ \forall h \in \mathcal{R}^d,\ \|\nabla f(x + h) - \nabla f(x)\| \leqslant L\|h\| \tag{5}$$

where $\|\cdot\|$ denotes the standard Euclidean norm.

We denote the class of convex smooth functions by $\mathcal{F}_L$. However, in the context of block coordinate algorithms, most of the existing work assumes instead a form of coordinate-wise smoothness.

Definition 3.2: Given a vector $\mathbf{L} = (L_1, \ldots, L_p)$ of p nonnegative constants, f is $\mathbf{L}$-coordinate-wise smooth if and only if we have $\forall i \in 1, \ldots, p$

$$\forall x \in \mathcal{R}^d,\ \forall h_i \in \mathcal{R}^{d_i},\ \|\nabla_i f(x+U_i h_i) - \nabla_i f(x)\| \leqslant L_i \|h_i\| \tag{6}$$

We denote the class of convex coordinate-wise smooth functions by $\mathcal{F}_{\mathbf{L}}^{\text{coord}}$.

We will focus on the analysis of L-smooth function $\mathcal{F}_L$, which proves simpler. However, our results will have direct implications for the class of coordinate-wise smooth functions $\mathcal{F}_L^{\text{coord}}$ thanks to the following lemma, whose proof is direct.

Lemma 3.1: Given any vector $\mathbf{L} \in \mathcal{R}_+^p$, we have $\mathcal{F}_{L_{min}} \subset \mathcal{F}_L^{\text{coord}} \subset \mathcal{F}_{\overline{L}}$ where $\overline{L} = \sum_{i=1}^p L_i$ and $L_{min} = \min_{i=1,\ldots,p} L_i$.

This lemma shows that our exact worst-case bound for block coordinate algorithms on (globally) smooth functions also automatically provide valid lower and upper bounds for the class of coordinate-wise smooth functions.

IV. Smooth convex interpolation.

We now review the smooth convex interpolation result of [19], which is crucial when deriving exact worst-case bounds for first-order optimization algorithms over the class of smooth convex functions, and will be instrumental in our analysis in Section V.

Definition 4.1: A set $\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=1,\ldots N} \subset \mathcal{R}^d \times \mathcal{R}^d \times \mathcal{R}$ such as $x_i^{(n)} = U_i^T x^{(n)}$ and $g_i^{(n)} = U_i^T g^{(n)}$, $\forall i = 1, \ldots, p$ is $\mathcal{F}_L$-interpolable if and only if there exists a function $f \in \mathcal{F}_L$ such that

$$\begin{aligned} &f^{(n)} = f(x^{(n)})\ \forall n \in \{1, \ldots, N\} \\ &g_i^{(n)} = \nabla_i f(x^{(n)})\ \forall i \in \{1, \ldots, p\},\ \forall n \in \{1, \ldots, N\} \end{aligned} \tag{7}$$

Theorem 4.1: The set $\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=1,\ldots,N}$ is $\mathcal{F}_L$-interpolable if and only if

$$\begin{aligned} &\forall n, l \in \{1, \ldots, N\}, \\ &f^{(n)} \geqslant f^{(l)} + \sum_{i=1}^p \langle g_i^{(l)}, x_i^{(n)} - x_i^{(l)} \rangle + \frac{1}{2L} \sum_{i=1}^p \|g_i^{(n)} - g_i^{(l)}\|^2 \end{aligned} \tag{8}$$

Any set $\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=1,\ldots N}$ satisfying (8) is thus consistent with an actual globally defined smooth convex function. We refer the reader to [19] for a detailed proof of this result expressed in terms of the full gradient of f. The reformulation in terms of partial gradients shown in (8) is direct and more suitable to handle block coordinate algorithms.

V. Worst-case behaviour of coordinate descent-like algorithms.

We now present the performance estimation framework that will allow us to derive exact worst-case bounds for block coordinate algorithms. The idea behind the PEP framework is to cast the performance analysis of an optimization method as an optimization problem itself over the class of functions $\mathcal{F}_L$ (see [18] and [19]).

We first examine K cycles of the (CCD) algorithm applied to a function with p blocks. We consider a performance measure equal to the difference between the objective function value of the last iterate and the optimal value, which is equal to $f(x^{(pK)}) - f(x^{(*)})$ where $x^{(*)}$ is any minimizer of f. We also assume that the distance from the starting point $x^{(0)}$ to that minimizer is bounded by a constant R. The exact worst-case of (CCD) for that performance measure over all L-smooth convex functions will be denoted by $\mathcal{W}_L^{\text{CCD}(\alpha)}(p, K, R)$. For clarity, we will sometimes omit the parameters of the worst-case and refer to it by $\mathcal{W}_L^{\text{CCD}(\alpha)}$.

For pedagogical reasons, we describe first a simple example of a PEP formulation for one cycle of (CCD) in the case of two blocks, whose iterations are written $x^{(1)} = x^{(0)} - \alpha U_1 \nabla_1 f(x^{(0)})$ and $x^{(2)} = x^{(1)} - \alpha U_2 \nabla_2 f(x^{(1)})$. The corresponding worst-case value $\mathcal{W}_L(p = 2, K = 1, R)$ is then given by the optimal value of the following problem:

$$\begin{aligned} \max_{f, x^{(0)}, x^{(1)}, x^{(2)}, x^{(*)}} \quad & f(x^{(2)}) - f(x^{(*)}) \\ & f \in \mathcal{F}_L \\ & \|x^{(0)} - x^{(*)}\|^2 \leqslant R^2 \\ & x^{(1)} = x^{(0)} - \alpha U_1 \nabla_1 f(x^{(0)}) \\ & x^{(2)} = x^{(1)} - \alpha U_2 \nabla_2 f(x^{(1)}) \\ & x^{(*)} \text{ is a minimizer of } f \end{aligned} \tag{9}$$

Problem (9) is an infinite-dimensional problem over the class of functions $\mathcal{F}_L$ and cannot be directly solved numerically. Convex interpolation transforms such problems into tractable finite-dimensional convex semidefinite programs (SDPs). The idea is to replace f by a set of variables of the form $\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=0,1,2,*}$ and require them to satisfy the interpolation conditions (8), so that they are $\mathcal{F}_L$- interpolable.

$$\begin{aligned} \max_{\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=0,1,2,*}} \quad & f^{(2)} - f^* \\ & \{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=0,1,2,*} \text{ satisfy (8)} \\ & \|x^{(0)} - x^{(*)}\|^2 \leqslant R^2 \\ & x_1^{(1)} = x_1^{(0)} - \alpha g_1^{(0)} \text{ and } x_2^{(1)} = x_2^{(0)} \\ & x_1^{(2)} = x_1^{(1)} \text{ and } x_2^{(2)} = x_2^{(1)} - \alpha g_2^{(1)} \\ & g_1^{(*)} = g_2^{(*)} = 0 \end{aligned} \tag{PEP-CCD}$$

Problem (PEP-CCD) is an exact reformulation of problem (9) because the interpolation conditions (8) guarantee the existence of a smooth convex function that interpolates the set of variables in problem (PEP-CCD) and therefore provides a solution of problem (9). We now give a more general form

of PEPs for block coordinate algorithms with a fixed arbitrary number of blocks and cycles

$$\begin{aligned}\mathcal{W}_L^{\text{BCD}}(p,K,R) = \max\ & \mathcal{P}(\mathcal{O}_n)\\ & \mathcal{O}_n \text{ satisfies (8)}\\ & \mathcal{I}(\mathcal{O}_n) \leqslant R^2\\ & \{x^{(n)}\}_{n=1,\dots,N} \text{ are computed}\\ & \text{by the considered algorithm}\end{aligned} \tag{PEP-BCD}$$

where $\mathcal{O}_n$ denotes the set $\{(x^{(n)}, g^{(n)}, f^{(n)})\}_{n=0,\dots,N,*}$, $\mathcal{P}(\mathcal{O}_n)$ denotes a performance criterion and $\mathcal{I}(\mathcal{O}_n)$ a measure of the optimality of the initial iterate such as $\|x^{(0)} - x^{(*)}\|^2$. For the (CCD) and (CACD) algorithm, all constraints in the PEP are linear equalities involving the iterates $x^{(n)}$ and the gradients $g^{(n)}$. This can be written in term of Gram matrices of vectors $\{x^{(n)}\}$ and $\{g^{(n)}\}$. Indeed, the scalar products in (8) can then be expressed thanks to $x^{(n)T} g^{(l)} = \sum_{i=1}^{p} x_i^{(n)} g_i^{(l)}$. For the (CAM) algorithm, the partial minimization along a coordinate amounts to imposing a partial gradient equal to zero along this coordinate, which can also be written in terms of Gram matrices. We refer the reader to [19] for the detailed procedure to transform a PEP into a SDP in the case of full gradient methods. The procedure we use to transform (PEP-BCD) into a SDP is very similar to one presented in [19]. Note that this procedure also requires that $\mathcal{P}$ and $\mathcal{I}$ can be written in terms of gram matrices, which is the case for the choices of $\mathcal{P}$ and $\mathcal{I}$ that we make in the sequel. The main difference is that we have to consider multiple blocks by defining a distinct Gram matrix for each of them. For a given number p of blocks, we solve an SDP involving p semidefinite matrices.

Theorem 5.1: If f is a convex L-smooth function, then the performance of K cycles of any (BCD) algorithm over f, denoted by $\mathcal{P}(f)$, verifies the following

$$\mathcal{P}(f) \leqslant \mathcal{W}_L^{\text{BCD}}(p,K,R) \tag{10}$$

Proof: This is a direct consequence of the optimization problem (PEP-BCD) that defines $\mathcal{W}_L^{\text{BCD}}(p,K,R)$. ∎

The previous theorems allow us to analyze the exact performance of block coordinate algorithms on L-smooth functions. However, we already noted that rates are in general expressed over the class of *coordinate-wise smooth functions*. The next result allows us to derive bounds for this class of functions.

Theorem 5.2: For any given vector of p nonnegative constants $\mathbf{L} = (L_1, \dots, L_p)$ and any block coordinate algorithm, we denote by $\mathcal{W}_{\text{coord},\mathbf{L}}^{\text{BCD}}(p,K,R)$ the worst-case of some given (BCD) algorithm over the class of coordinate-wise smooth functions $\mathcal{F}_{\mathbf{L}}^{\text{coord}}$ and we have

$$\mathcal{W}_{L_{min}}^{\text{BCD}}(p,K,R) \leqslant \mathcal{W}_{\text{coord},\mathbf{L}}^{\text{BCD}}(p,K,R) \leqslant \mathcal{W}_{\overline{L}}^{\text{BCD}}(p,K,R) \tag{11}$$

where $\mathcal{W}_{L_{min}}^{\text{BCD}}(p,K,R)$, $\mathcal{W}_{\overline{L}}^{\text{BCD}}(p,K,R)$ denote respectively the bound given by the PEP for the classes of functions $\mathcal{F}_{L_{min}}$ and $\mathcal{F}_{\overline{L}}$.

Proof: Let us consider $f \in \mathcal{F}_{\mathbf{L}}^{\text{coord}}$. Thanks to Lemma (3.1), we know that $f \in \mathcal{F}_{\overline{L}}$ thus the performance of the algorithm $\mathcal{P}(f)$ on f verifies

$$\mathcal{P}(f) \leqslant \mathcal{W}_{\overline{L}}^{\text{BCD}}(p,K,R)$$

Since this is true for every function $f \in \mathcal{F}_{\mathbf{L}}^{\text{coord}}$, we have that

$$\mathcal{W}_{\text{coord},\mathbf{L}}^{\text{BCD}}(p,K,R) \leqslant \mathcal{W}_{\overline{L}}^{\text{BCD}}(p,K,R)$$

Similarly by considering the other inclusion given in Lemma (3.1), we obtain the other inequality of Theorem 5.2 ∎

Remark 5.1: Theorem 5.2 applies when the same algorithm, including its coefficient value, is considered for the three classes. Its use requires some care, as algorithm parameters are frequently made implicitly dependent on the function class parameters; step-sizes are e.g. typically described as $\frac{h}{L}$. The application of Theorem 5.2 in such cases require thus first fixing the values of these parameters independently of L.

We will focus on particular instances of PEPs by choosing different performance criteria $\mathcal{P}$ and starting iterate conditions $\mathcal{I}$. For the performance criterion, we will consider the difference between the function value at the last iterate and the optimal value of the function

$$\mathcal{P}(\mathcal{O}_n) = f^{(N)} - f^*$$

and the squared gradient norm of the last iterate

$$\mathcal{P}(\mathcal{O}_n) = \|g^{(N)}\|^2$$

As starting iterate condition, the usual assumption made for the analysis of cyclic coordinate algorithms in [3], [11], is that the set $S = \{x \in \mathcal{R}^d : \ f(x) \leqslant f(x^{(0)})\}$ is compact which implies that

$$\|x^{(pk)} - x^{(*)}\| \leqslant R(x^{(0)}),\ \forall k \in 1, \dots, K \tag{12}$$

with $R(x^{(0)})$ defined as

$$R(x^{(0)}) = \max_{x \in X^{(*)}} \max_{x \in \mathcal{R}^d} \{\|x - x^{(*)}\| : \ f(x) \leqslant f(x^{(0)})\} \tag{13}$$

The convergence proofs in [3], [11] rely on inequality (12). We thus introduce the following Setting ALL. such that the convergence theorems in [11] remain valid in this setting.

Setting ALL. We consider the following assumption on all the iterates

$$\min_{x^{(*)} \in X^{(*)}} \max_{k=1\dots,K} \|x^{(pk)} - x^{(*)}\| \leqslant R_a \tag{14}$$

Theorem 5.3: Let $\{x^{(n)}\}_{n=1\dots,N}$ be a sequence generated by the (CCD) algorithm with a constant step-size $\alpha \leqslant \frac{1}{L}$ then

$$f(x^{(N)}) - f^* \leqslant \frac{4}{\alpha}(1 + p\alpha^2 L^2)\frac{p}{N+8} R_a^2 \tag{15}$$

Proof: We follow the same proof as [11, Theorem 3.6] where we replace inequality (12) by

$$\|x^{(pk)} - x^{(*)}\| \leqslant R_a \ \forall k \in 1, \ldots, K \tag{16}$$

■

Though theoretically convenient, Setting ALL. is not very natural and may be difficult to verify in practice. In addition, Setting ALL. can prove to be unusable for certain class of functions. Indeed, consider the family of smooth functions $f_\epsilon(x, y) = (x - y)^2 + \epsilon(x^2 + y^2)$ and the initial point $(x_0 = 1, y_0 = -1)$ for 2-block coordinate algorithm. We have that $R(x_0) = R_a = \frac{1}{\sqrt{\epsilon}}$ which tends to infinity when ϵ tends to zero. For ϵ small enough the bounds obtained in this setting are very conservative and do not give useful information about the performance of the algorithm. Therefore, we will also consider a more classical setting, albeit less frequently used in the context of deterministic block coordinate algorithms:

Setting INIT. Given the starting point of the block coordinate algorithm $x^{(0)}$ and an optimal point of the function $x^{(*)}$, we have that

$$\|x^{(0)} - x^{(*)}\|^2 \leqslant R_i^2, \tag{17}$$

VI. Bounds on the worst-case of coordinate descent algorithms using the PEP framework

We now exploit our PEP-based approach to revisit and in some cases significantly improve the bounds on the block coordinate algorithms defined in the first section. For simplicity, we focus on algorithms on the case of two blocks $(p = 2)$. Furthermore, we consider without loss of generality smoothness constants $L_1 = L_2 = 1$ and $R_a = R_i = 1$. We can retrieve the value of the worst-case in general thanks to the following scaling properties

$$\mathcal{W}_L^{\text{CCD}(\frac{\alpha}{L})}(p, K, R) = LR^2\mathcal{W}_1^{\text{CCD}(\alpha)}(p, K, 1)$$
$$\mathcal{W}_L^{\text{CAM}}(p, K, R) = LR^2\mathcal{W}_1^{\text{CAM}}(p, K, 1)$$
$$\mathcal{W}_L^{\text{CACD}(\frac{\alpha}{L})}(p, K, R) = LR^2\mathcal{W}_1^{\text{CACD}(\alpha)}(p, K, 1)$$

The bounds presented in this section are the ones on the worst-cases $\mathcal{W}^{\text{BCD}}_{\text{coord},(1,1)}(2, K, 1)$ given by Theorem 5.2

$$\mathcal{W}_1^{\text{BCD}}(2, K, 1) \leqslant \mathcal{W}^{\text{BCD}}_{\text{coord},(1,1)}(2, K, 1) \leqslant \mathcal{W}_2^{\text{BCD}}(2, K, 1)$$

Since a $\mathcal{O}(\frac{1}{K})$ rate of convergence is expected in most cases, we often show the evolution of our performance criterion multiplied by K to facilitate the analysis.

Algorithm (CCD). In Figure 1, we provide the upper and lower bound defined in Theorem 5.2 for the worst-case of the 2-block (CCD) with a constant step-size $\alpha = \frac{1}{2}$. We choose this step-size to ensure the convergence of our upper-bound that considers the class of 2-smooth functions. Our bounds are sublinear and improve by one order of magnitude the best know theoretical bound derived in [11, Theorem 3.6]. The lower bound in Figure 1 suggests that our upper bound on the worst-case of (CCD) over the class of coordinate-wise functions cannot be significantly improved. The experiments presented in Figure 1 are performed under Setting ALL defined in the previous section.

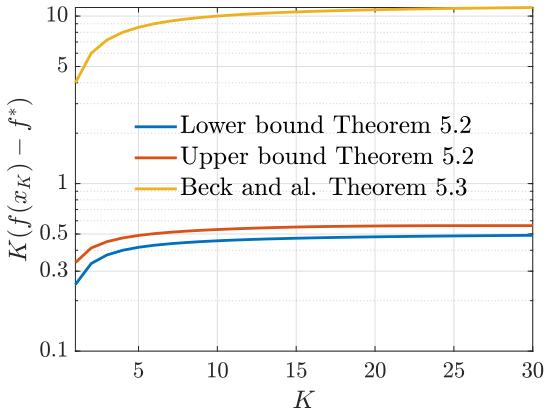

Fig. 1. Comparison between the PEP bounds Theorem 5.2 and the analytical bound in Theorem 5.3 multiplied by the number of cycles K, for the 2-block (CCD) in the Setting ALL.

In order to illustrate the flexibility and usefulness of our framework, we provide bounds for the 2-block (CCD) algorithm in the Setting INIT. In Figure 2 we compare the upper bounds obtained for both settings for block coordinate descent. Our results show that the convergence of cyclic block coordinate descent can also be established under the weaker assumption of Setting INIT, and the performance results suggest that the stronger assumptions made in Setting ALL do not yield a significant improvement in terms of performance.

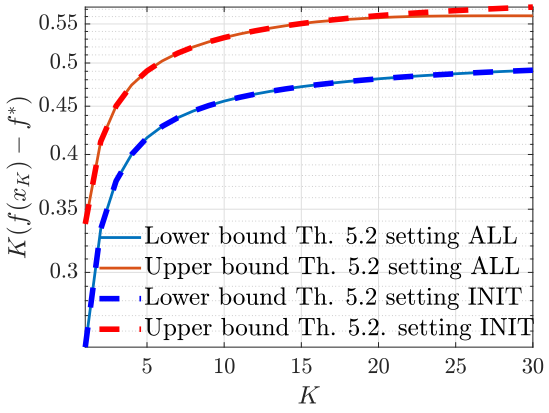

Fig. 2. Comparison between the PEP bounds Theorem 5.2 multiplied by the number of cycles K, for the 2-block (CCD) in Settings ALL (full lines) and INIT.(dashed lines)

We also provide bounds for the (CCD) algorithm with the gradient norm as a performance criterion. We show in Figure 3 the reciprocal of our PEP bound on the residual gradient which suggests that the squared residual gradient norm converges with a rate of $\mathcal{O}(1/K^2)$ which is faster then the convergence of the function accuracy.

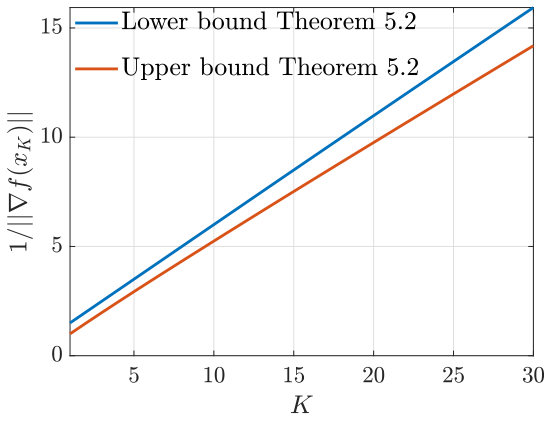

Fig. 3. Reciprocal of our PEP bounds Theorem 5.2 on the residual gradient norm for the 2-block (CCD) in Setting all.

Algorithm (CAM). In Figure 4 we also provide upper and lower bounds for the alternating minimization algorithm in setting ALL and compare it to the bound provided in [11, Theorem 5.2]. Figure 2 shows that our PEP-based approach improves bounds by a factor of two.

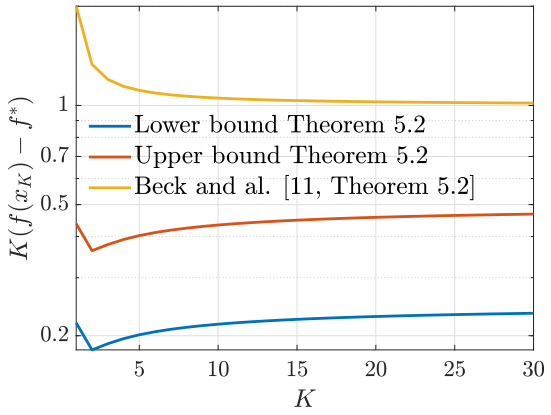

Fig. 4. Comparison between the PEP bounds Theorem 5.2 and the analytical bound in [11, Theorem 5.2] multiplied by the number of cycles K, for the 2-block (CAM) algorithm in Setting ALL.

Algorithm (CACD). Finally, we provide an exact worst-case bound for the cyclic 2-block version of the random accelerated coordinate algorithm (RACD) (CACD) derived in [5] over the class of 1-smooth functions which, thanks to Theorem 5.2, also gives a lower bound on the worst-case for the class of function $\mathcal{F}_L^{\text{coord}}$. We choose a step-size $\alpha = \frac{1}{L} = 1$ since the results established for (RACD) in [5] are for a step-size $\frac{1}{L}$. The (RACD) algorithm from [5] has a $\mathcal{O}(\frac{1}{K^2})$ rate of convergence. In Figure 5, we plot the reciprocal of our PEP bound and provide a numerical linear fit that suggests that the rate of convergence of (CACD) is $\mathcal{O}(\frac{1}{K})$. We do not observe acceleration in the sense that the rate of convergence is slower than $\mathcal{O}(\frac{1}{K^2})$. We numerically fit our lower bound

$$\mathcal{W}_1^{\text{CACD}(1)}(2, K, 1) \approx \frac{0.0945}{K - 3.1026} \tag{18}$$

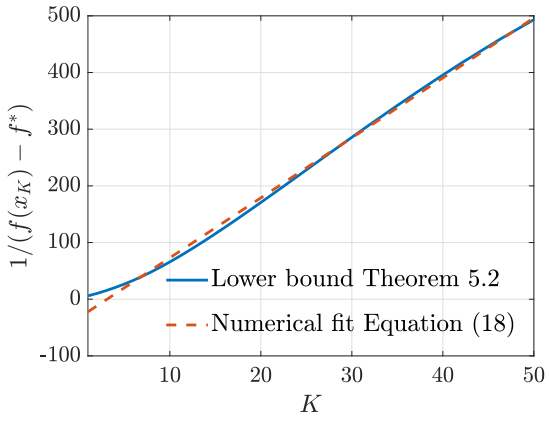

Fig. 5. Reciprocal of the PEP lower bound Theorem 5.2 and its linear numerical fit for the 2-block (CACD), which indicates that the convergence rate is slower than $\mathcal{O}(\frac{1}{K^2})$.

This indicates that using randomness in the choice of the block of coordinates to update plays a crucial role in the acceleration of block coordinate algorithms. To investigate this further, we adapted our PEP framework to compare the cyclic (CACD) and random (RACD) versions of the algorithm presented in [5]. The PEP framework is usually only able to handle deterministic algorithms. To circumvent this issue, we write a PEP that computes simultaneously all possible choices of coordinate steps, and use as a performance criterion the worst-case average of the performances of each combination, which corresponds exactly to the expectation of the performance of the (RACD) algorithm in [5]. For readability, we refer to this approach as the worst-case of the average, and we compare it to the worst-case of the deterministic version of the algorithm in [5] for each possible combination of steps. We consider again 2-blocks of coordinates for N partial steps of gradient. Since there are 2^N possible combinations of steps and the dimension of the SDP grows with that number, we only present preliminary results for $N = 4$. The worst-case of the average in this case denoted by $\mathcal{W}^{\text{RACD}}$ is equal to $\mathcal{W}^{\text{RACD}} = 0.102$.

Table I gives the worst-case for each possible combination of $N = 4$ partial steps. Note that all the worst-cases in Table I are larger than $\mathcal{W}^{\text{RACD}}$ except for the one of the cyclic choice of steps and the one for which the order of steps is reversed after each cycle. However these two worst-cases are very close to the worst-case of the average (random choice of steps). A possible explanation for this is that the worst-case of the average involves choices of steps that are obviously suboptimal such as the case where we choose only to update along one coordinate. In order to confirm this, we computed the worst-case of the average of the 2^8 possible subsets of deterministic step choices by grouping together the symetrical ones. The best performance is obtained for the worst-case of the average of the four best deterministic choices of steps (cyclic and reversing the order after each cycle) which correponds to the worst-case of the expectation of a random choice between these four possible combinaisons. We obtain $\mathcal{W}_{\text{best}} = 0.094$, which is slightly lower then the worst-case of every deterministic choices. In the case of $\mathcal{W}_{\text{best}}$,

the worst-case of the average outperforms the deterministic choices used in the average which was not the case for $\mathcal{W}^{\text{RACD}}$. Although The preliminary results of Table I are not entirely conclusive, combined with the ones of Figure 5, it seems that (CACD) is indeed slower than its random version. Furthermore the results of Table I tend to indicate that for general smooth functions the best deterministic choice of coordinates is the cyclic one.

Block choice type	Ordered block choices	Worst-case
Cyclic (= $\mathcal{W}^{\text{CACD}}$)	1 2 1 2 or 2 1 2 1	0.098
Fixed choice	1 2 2 1 or 2 1 1 2	0.100
Fixed choice	1 2 1 1 or 2 1 2 2	0.135
Fixed choice	1 1 2 1 or 2 2 1 2	0.152
Fixed choice	1 2 2 2 or 2 1 1 1	0.155
Fixed choice	1 1 2 2 or 2 2 1 1	0.1667
Fixed choice	1 1 1 2 or 2 2 2 1	0.188
Fixed choice	1 1 1 1 or 2 2 2 2	0.500
Random (= $\mathcal{W}^{\text{RACD}}$)	at each step: pick 1 or 2	0.102
Random best	at each cycle: pick 1 2 or 2 1	0.094

TABLE I

ACCELERATED COORDINATE DESCENT WORST-CASE ($p = 2$, $K = 2$) FOR SOME DETERMINISTIC/RANDOM BLOCK CHOICES

VII. CONCLUSION.

We have developed a flexible framework for the automated worst-case analysis of block coordinate algorithms, and provided exact worst-case bounds over the class of smooth functions, which lead to upper and lower bounds on the class of coordinate-wise smooth functions. We have provided improved numerical bounds, sometimes by an order of magnitude, for three types of block coordinate algorithms: cyclic coordinate descent (CCD), cyclic alternating minimization (CAM) and a cyclic version of the accelerated random coordinate descent in [5] (CACD). In addition, we highlighted the importance of randomness for existing acceleration schemes, since our numerical experiments suggest that deterministic cyclic algorithms do not accelerate i.e they do not achieve a $\mathcal{O}(\frac{1}{K^2})$ rate of convergence. Further research could involve developing interpolation conditions for the class of coordinate-smooth functions, performing more numerical experiments in a wider range of settings, with more blocks and different step-sizes, as well as searching with our PEP-based approach an efficient acceleration schemes for cyclic block coordinate algorithms over the class of coordinate-wise smooth functions.

ACKNOWLEDGEMENT

Y. Kamri is supported by the European Union's MARIE SKŁODOWSKA-CURIE Actions Innovative Training Network (ITN)-ID 861137, TraDE-OPT.

REFERENCES

[1] H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin, "A primer on coordinate descent algorithms," *arXiv: Optimization and Control*, 2016. [Online]. Available: https://arxiv.org/abs/1610.00040

[2] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, p. 3–34, jun 2015. [Online]. Available: https://doi.org/10.1007/s10107-015-0892-3

[3] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012. [Online]. Available: https://doi.org/10.1137/100802001

[4] Q. Lin, Z. Lu, and L. Xiao, "An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2244–2273, 2015. [Online]. Available: https://doi.org/10.1137/141000270

[5] O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 1997–2023, 2015. [Online]. Available: https://doi.org/10.1137/130949993

[6] J. Diakonikolas and L. Orecchia, "Alternating randomized block coordinate descent," *Proc. ICML'18*, 2018. [Online]. Available: https://arxiv.org/abs/1805.09185

[7] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan, "Even faster accelerated coordinate descent using non-uniform sampling," *Proc. ICML'16*, 2016. [Online]. Available: https://arxiv.org/abs/1512.09103

[8] F. Hanzely and P. Richtarik, "Accelerated coordinate descent with arbitrary sampling and best rates for minibatches," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 304–312. [Online]. Available: https://proceedings.mlr.press/v89/hanzely19a.html

[9] Y. Nesterov and S. U. Stich, "Efficiency of the accelerated coordinate descent method on structured optimization problems," *SIAM Journal on Optimization*, vol. 27, no. 1, pp. 110–123, 2017. [Online]. Available: https://doi.org/10.1137/16M1060182

[10] A. Saha and A. Tewari, "On the nonasymptotic convergence of cyclic coordinate descent methods," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 576–601, 2013. [Online]. Available: https://doi.org/10.1137/110840054

[11] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013. [Online]. Available: https://doi.org/10.1137/120887679

[12] R. Sun and M. Hong, "Improved iteration complexity bounds of cyclic block coordinate descent for convex problems," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf

[13] X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong, "An improved convergence analysis of cyclic block coordinate descent-type methods for strongly convex minimization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 491–499. [Online]. Available: https://proceedings.mlr.press/v51/li16c.html

[14] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, "Iteration complexity analysis of block coordinate descent methods," *Mathematical programming*, 2017.

[15] S. J. Wright and C.-P. Lee, "Analyzing random permutations for cyclic coordinate descent," *Mathematics of computation*, vol. 89, no. 325, pp. 2217–2248, Jan. 2020.

[16] M. Gurbuzbalaban, A. Ozdaglar, P. A. Parrilo, and N. Vanli, "When cyclic coordinate descent outperforms randomized coordinate descent," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/0e7c7d6c41c76b9ee6445ae01cc0181d-Paper.pdf

[17] B. Goujaud, D. Scieur, A. Dieuleveut, A. B. Taylor, and F. Pedregosa, "Super-acceleration with cyclical step-sizes," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 3028–3065. [Online]. Available: https://proceedings.mlr.press/v151/goujaud22a.html

[18] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: a novel approach." *Math. Program.*, vol. 145, pp. 451–482, 2014. [Online]. Available: https://doi.org/10.1007/s10107-013-0653-0

[19] A. B. Taylor, J. M. Hendrickx, and F. Glineur, "Smooth strongly convex interpolation and exact worst-case performance of first-order methods." *Math. Program.*, vol. 161, pp. 307–345, 2017. [Online]. Available: https://doi.org/10.1007/s10107-016-1009-3

[20] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016. [Online]. Available: https://doi.org/10.1137/15M1009597

[21] Z. Shi and R. Liu, "Better worst-case complexity analysis of the block coordinate descent method for large scale machine learning," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 889–892.

[22] A. B. Taylor and F. R. Bach, "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions," in *COLT*, ser. Proceedings of Machine Learning Research, vol. 99. PMLR, 2019, pp. 2934–2992.