CONVERGENCE RATE ANALYSIS OF RANDOMIZED AND CYCLIC COORDINATE DESCENT FOR CONVEX OPTIMIZATION THROUGH SEMIDEFINITE PROGRAMMING

HADI ABBASZADEHPEIVASTI, ETIENNE DE KLERK*, MOSLEM ZAMANI

Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands

Abstract. In this paper, we study randomized and cyclic coordinate descent for convex unconstrained optimization problems. We improve the known convergence rates in some cases by using the numerical semidefinite programming performance estimation method. As a spin-off we provide a method to analyse the worst-case performance of the Gauss-Seidel iterative method for linear systems where the coefficient matrix is positive semidefinite with a positive diagonal.
Keywords. Cyclic and randomized coordinate descent; Gauss-Seidel method; Semidefinite programming.
2020 Mathematics Subject Classification. 90C25, 90C22.

1. INTRODUCTION

We consider the unconstrained optimization problem

$$f^\star = \min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex. We assume that f attains its minimum and $f^\star$ denotes the optimal value. In addition, we assume that f is an L-smooth function, that is,

$$\|\nabla f(y) - \nabla f(x)\| \le L\|y - x\|, \quad \forall y, x \in \mathbb{R}^n.$$

Moreover, we denote the component Lipschitz constants by ℓ_i ($i \in \{1, \dots, n\}$), i.e.,

$$|[\nabla f(x + te_i)]_i - [\nabla f(x)]_i| \le \ell_i |t|, \quad \forall x \in \mathbb{R}^n, t \in \mathbb{R}, \tag{1.2}$$

where e_i is the ith standard unit vector. Let $\ell_{\max} := \max_{1 \le i \le n} \ell_i$, and note that $1 \le \frac{L}{\ell_{\max}} \le n$.

Due to the simplicity and small per-iteration cost, coordinate descent methods have been employed extensively for large-scale optimization problems [1, 2].

The generic coordinate descent method is demonstrated in Algorithm 1.

Algorithm 1 Generic coordinate descent

Set N and $\{t_k\}_{k=0}^{N-1}$ (step lengths) and pick $x^0 \in \mathbb{R}^n$.
For $k = 0, 1, \ldots, N-1$ perform the following steps:
(1) Choose an index i_k from $\{1, 2, \ldots, n\}$.
(2) $x^{k+1} = x^k - t_k [\nabla f(x^k)]_{i_k} e_{i_k}$.

In this paper, we revisit the worst-case convergence rate analysis for Algorithm 1 for two of the best known variants, namely *randomized coordinate descent*, and *cyclic coordinate descent*. In the former, the index i_k is chosen uniformly at random from $\{1, 2, \ldots, n\}$, and in the latter, the cyclic ordering is used.

We will improve the best-known convergence rates from the literature for some specific values of the parameters n, L, N, t_k for $k \in \{0, 1, \ldots, N-1\}$ and ℓ_i for $i \in \{1, \ldots, n\}$. Finally, the Gauss-Seidel iterative method for positive semidefinite linear systems is a special case of cyclic coordinate descent for convex quadratic functions, and we will investigate the implications of our analysis for this classical method as well.

Recently, Taylor and Bach [3, Appendix I] and Kamri et al. [4] studied the convergence of the coordinate descent algorithm using the semidefinite programming (SDP) performance estimation method, that was introduced by Drori and Teboulle [5]. We will also use SDP performance estimation in our analysis, but in a different way than Taylor and Bach [3, Appendix I], and our main contribution may be seen as the extension and refinement of the approach by Kamri et al. [4]. SDP performance estimation has been applied to the analysis of many iterative methods (other than coordinate descent); the interested reader may consult [6, 7, 8, 9, 10] and the references therein. For general background information on SDP, see, e.g., [11].

Notation and background results. We use $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to denote the Euclidean inner product and norm, respectively, unless indicated otherwise. The column vector e_i represents the i-th standard unit vector and I stands for the identity matrix. For a matrix A, a_{ij} denotes its (i, j)-th entry, and $A^\top$ represents the transpose of A. The function $f : \mathbb{R}^n \to \mathbb{R}$ is called μ-strongly convex function if the function $x \mapsto f(x) - \frac{\mu}{2}\|x\|^2$ is convex. Clearly, any convex function is 0-strongly convex. We denote the set of real-valued convex functions which are L-smooth and μ-strongly convex by $\mathscr{F}_{\mu,L}(\mathbb{R}^n)$.

Let $\mathscr{I}$ be a finite index set and let $\{x^i; g^i; f^i\}_{i \in \mathscr{I}} \subseteq \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$. A set $\{x^i; g^i; f^i\}_{i \in \mathscr{I}}$ is called $\mathscr{F}_{\mu,L}$-interpolable if there exists $f \in \mathscr{F}_{\mu,L}(\mathbb{R}^n)$ with $f(x^i) = f^i$, $g^i \in \partial f(x^i)$ $i \in \mathscr{I}$.

The following theorem gives necessary and sufficient conditions for $\mathscr{F}_{\mu,L}$-interpolablity.

Theorem 1.1. [6] *Let $L \in (0, \infty)$ and $\mu \in [0, \infty)$ and $f \in \mathscr{F}_{\mu,L}(\mathbb{R}^n)$. For any $x, y \in \mathbb{R}^n$, we have*

$$\frac{1}{2(1-\frac{\mu}{L})} \left(\frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2 + \mu \|x - y\|^2 - \frac{2\mu}{L} \langle \nabla f(y) - \nabla f(x), y - x \rangle \right) \tag{1.3}$$
$$\leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle.$$

Conversely, if $\mathscr{I}$ is a finite index set and $\{x^i; g^i; f^i\}_{i \in \mathscr{I}} \subseteq \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$ are given data, then the data set is $\mathscr{F}_{\mu,L}$-interpolable if it satisfies (1.3) in the sense that, for each pair $i, j \in \mathscr{I}$:

$$\frac{1}{2(1-\frac{\mu}{L})} \left(\frac{1}{L} \|g^i - g^j\|^2 + \mu \|x^i - x^j\|^2 - \frac{2\mu}{L} \langle g^j - g^i, x^j - x^i \rangle \right) \leq f^i - f^j - \langle g^j, x^i - x^j \rangle.$$

2. CONVERGENCE RATE OF RANDOMIZED COORDINATE DESCENT

The randomized coordinate descent method is shown in Algorithm 2 for easy reference.

Algorithm 2 Randomized coordinate descent

Set N and $\{t_k\}_{k=0}^{N-1}$ (step lengths) and pick $x^0 \in \mathbb{R}^n$.
For $k = 0, 1, \ldots, N-1$ perform the following steps:
(1) Choose index i_k with uniform probability from $\{1, 2, \ldots, n\}$.
(2) $x^{k+1} = x^k - t_k [\nabla f(x^k)]_{i_k} e_{i_k}$.

We proceed to revisit its worst-case convergence rate for three classes of functions, namely convex L-smooth functions, convex quadratic functions, and strongly convex, L-smooth functions.

2.1. **The case of L-smooth functions.**
Regarding the convergence of Algorithm 2 for L-smooth convex functions, the following is known. (We state the result as in the survey [2, Theorem 1], but it is originally due to Nesterov [1]).

Theorem 2.1. [2, Theorem 1] *Let $f : \mathbb{R}^n \to \mathbb{R}$ is an L-smooth convex function for some $L > 0$. If $t_k = \frac{1}{\ell_{\max}}$ for all k, then, for each $k > 0$,*

$$\mathbb{E}\left(f(x^k)\right) - f^\star \leq \left(\frac{2n\ell_{\max}}{k}\right) R_0^2, \tag{2.1}$$

where R_0 satisfies $\max_{x^\star \in \mathbb{S}} \max_x \{\|x - x^\star\| : f(x) \leq f(x^0)\} \leq R_0$ and $\mathbb{S}$ denotes the optimal solution set.

In this section, we study the behaviour of randomized coordinate descent method for L-smooth convex functions. The worst-case convergence rate of the Algorithm 2 can be formulated as follows

$$
\begin{aligned}
\max \ & \mathbb{E}[f(x^N)] - f(x^\star) \\
\text{s.t.} \ & f \in \mathscr{F}_{0,L}(\mathbb{R}^n), \\
& f \text{ satisfies (1.2) for every } x \in \mathbb{R}^n \text{ for some } \ell_i, i \in \{1, \ldots, n\}, \\
& \|x^0 - x^\star\|^2 \leq \Delta, \\
& x^k \ k \in \{1, 2, \ldots, N\} \text{ are generated by Algorithm 2 with respect to } x^0 \text{ and step length } t_k, \\
& x^0 \in \mathbb{R}^n, \ \nabla f(x^\star) = 0,
\end{aligned} \tag{2.2}
$$

where $f, x^k, x^\star$ are decision variables and t, L, n and $\ell_i, i \in \{1, \ldots, n\}$, are the given parameters. Problem (2.2) in general is intractable. Moreover, note that x^k depends on the index i_k which is chosen uniformly at random from the set $\{1, \ldots, n\}$ therefore (2.2) is a stochastic programming problem. To deal with this we introduce a random variable d^k which depends on the index i_k

and is defined by $d^k := [\nabla f(x^k)]_{i_k} e_{i_k}$. Note that d^k has the following properties:

$$
\begin{aligned}
\mathbb{E}\left[\|d^k\|^2\right] &= \tfrac{1}{n}\mathbb{E}\left[\|\nabla f(x^k)\|^2\right], \\
\mathbb{E}\left[\left\langle d^k, \nabla f(x^k)\right\rangle\right] &= \tfrac{1}{n}\mathbb{E}\left[\|\nabla f(x^k)\|^2\right], \\
\mathbb{E}\left[\left\langle d^k, x^k\right\rangle\right] &= \tfrac{1}{n}\mathbb{E}\left[\left\langle \nabla f(x^k), x^k\right\rangle\right],
\end{aligned}
\tag{2.3}
$$

where the expectation again refers to the joint distribution of all the random variables d^k for $k \in \{0, 1, \ldots, N\}$ and $x^k, \nabla f(x^k), f(x^k)$ for $k \in \{0, 1, \ldots, N\}$. By Taylor's theorem and (1.2), we have

$$
f(x^{k+1}) \leq f(x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k\right\rangle + \tfrac{\ell_{\max}}{2}\|x^{k+1} - x^k\|^2,
\tag{2.4}
$$

$$
f(x^k) \leq f(x^{k+1}) + \left\langle \nabla f(x^{k+1}), x^k - x^{k+1}\right\rangle + \tfrac{\ell_{\max}}{2}\|x^k - x^{k+1}\|^2,
$$

where, $\ell_{\max} = \max_{i \in \{1, \ldots, n\}} \ell_i$ as before. Therefore, the relaxation of problem (2.2) is given by

$$
\begin{aligned}
\max \ & \mathbb{E}[f(x^N)] - f(x^\star) \\
\text{s.t. } & \{(x^k; \nabla f(x^k); f(x^k))\} \text{ satisfy (1.3) for } k \in \{0, 1, \cdots, N, \star\} \text{ w.r.t. } \mu = 0, L, \\
& \{(x^k; \nabla f(x^k); f(x^k))\} \text{ satisfy (2.4) for } k \in \{0, 1, \cdots, N\} \text{ w.r.t. } \ell_{\max}, \\
& \{x^k; \nabla f(x^k); d^k\} \text{ satisfies (2.3)} \quad (k \in \{0, \cdots, N\}), \\
& \|x^0 - x^\star\|^2 \leq \Delta, \\
& x^{k+1} = x^k - t_k d^k, \\
& x^0 \in \mathbb{R}^n, \ \nabla f(x^\star) = 0,
\end{aligned}
\tag{2.5}
$$

where $f(x^k), x^k, x^\star, \nabla f(x^k)$ and d^k are decision variables. Note that because the problem (1.1) is invariant under translation, without loss of generality we may assume that $x^\star$ is the zero vector. Since $x^{k+1} = x^k - t_k d^k$ is a recursive relation, x^k can be written as linear combination of x^0 and d^is. In this way, all the unknowns appear as entries in the following matrix:

$$
G = \begin{pmatrix}
\mathbb{E}[\|x^0\|^2] & \mathbb{E}[\langle x^0, \nabla f(x^0)\rangle] & \cdots & \mathbb{E}[\langle x^0, \nabla f(x^N)\rangle] & \mathbb{E}[\langle x^0, d^0\rangle] & \cdots & \mathbb{E}[\langle x^0, d^N\rangle] \\
\mathbb{E}[\langle \nabla f(x^0), x^0\rangle] & \mathbb{E}[\|\nabla f(x^0)\|^2] & \cdots & \mathbb{E}[\langle \nabla f(x^0), \nabla f(x^N)\rangle] & \mathbb{E}[\langle \nabla f(x^0), d^0\rangle] & \cdots & \mathbb{E}\langle \nabla f(x^0), d^N\rangle \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathbb{E}[\langle d^N, x^0\rangle] & \mathbb{E}[\langle d^N, \nabla f(x^0)\rangle] & \cdots & \mathbb{E}[\langle d^N, \nabla f(x^N)\rangle] & \mathbb{E}[\langle d^N, d^0\rangle] & \cdots & \mathbb{E}[\|d^N\|^2]
\end{pmatrix}.
$$

Note that G is the expectation of a random Gram matrix. Since every realization of this random matrix is positive semidefinite, and the expectation preserves positive semidefiniteness, it follows that G is positive semidefinite as well. Therefore, problem (2.5) can be written as an SDP problem, where the variables are G and $\mathbb{E}[f(x^i)]$.

In what follows we compare the convergence rate derived by solving the problem (2.5) and the bound by Wright (2.1) for some specific values of the parameters n, L, N, t_k for $k \in \{0, 1, \cdots, N\}$ and ℓ_i for $i \in \{1, \cdots, n\}$. All the figures in this paper were obtained by solving the SDP problems with the solver Mosek [12], using the Yalmip [13] Matlab interface.
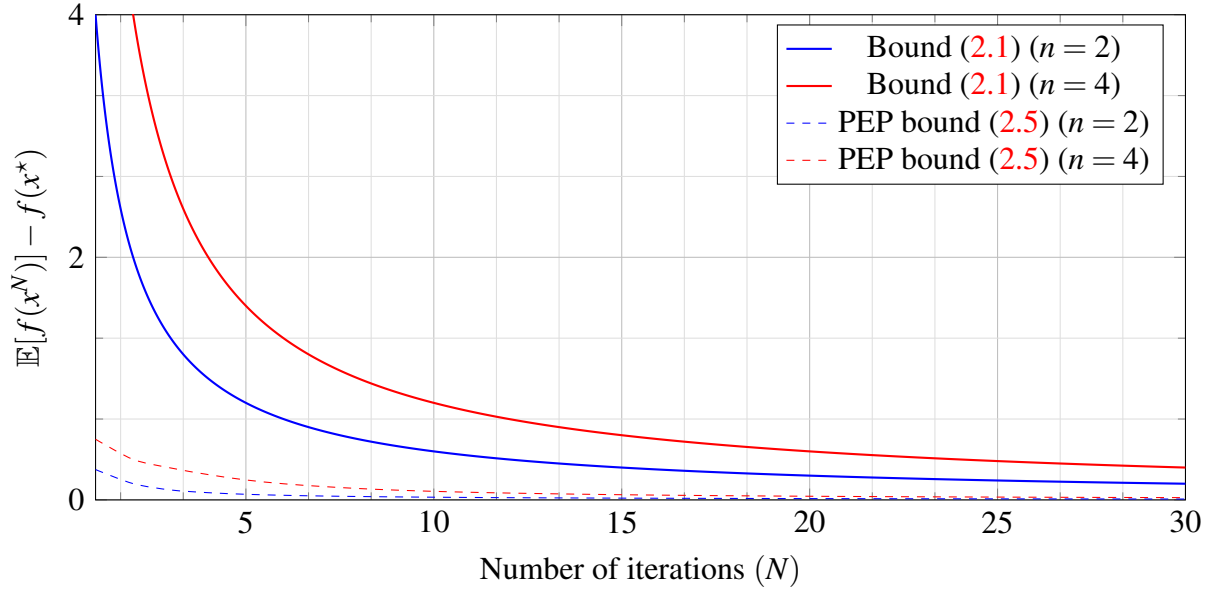
FIGURE 1. Convergence rate for Algorithm 2 computed by performance estimation problem (2.5) (dashed lines) and the bound given by (2.1) (thick lines) for $L = 2, l_{\max} = 1, t = 1, \Delta = 1$ and different n
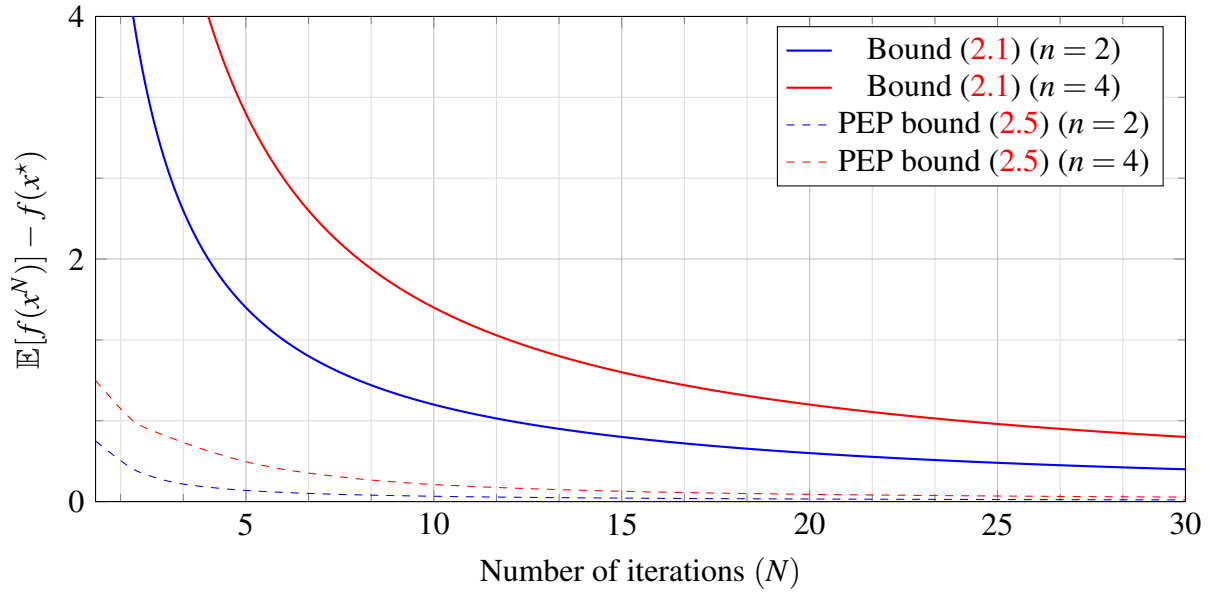

FIGURE 2. Convergence rate for Algorithm 2 computed by performance estimation problem (2.5) (dashed lines) and the bound given by (2.1) (thick lines) for $L = 4, l_{\max} = 2, t = 0.5, \Delta = 1$ and different n

Note that the convergence rate provided by solving performance estimation is strictly better than the bound given by Wright. In other words, the bound (2.1) is not tight for the values of the parameters that we considered. Moreover, the bound given by performance estimation can also be calculated for different step lengths than the fixed step lengths $1/\ell_{\max}$ in the bound (2.1).

2.2. **The case of convex quadratic functions.** In this section we study the convergence of rate of the randomized coordinate descent method in case that the objective function is quadratic function of the form

$$\min_{x \in \mathbb{R}^n} f(x) := \tfrac{1}{2} x^\top A x - b^\top x, \tag{2.6}$$

where, A is a symmetric positive semidefinite matrix. To study this case we need to add additional constraint to restrict our model to quadratic functions. The following necessary condition for f to be a quadratic function can be verified easily, and has been used in SDP performance analysis by Drori et al [8]:

$$\tfrac{1}{2} \langle \nabla f(x) - \nabla f(y), x - y \rangle = f(x) - f(y) - \langle \nabla f(y), x - y \rangle.$$

Since this constraint holds for every point in the domain we just consider the relaxed constraint that only holds for the point generated by the method in addition to the initial point and the optimal point. In this case we add the following constraint to the problem (2.5).

$$\tfrac{1}{2} \langle \nabla f(x^i) - \nabla f(x^j), x^i - x^j \rangle = f(x^i) - f(x^j) - \langle \nabla f(x^j), x^i - x^j \rangle \quad \forall i, j \in \{0, \ldots, N, *\}. \tag{2.7}$$

In what follows, we compare the convergence rate of the randomized coordinate descent method for the general problem (2.5) to the convergence rate for the quadratic problems.
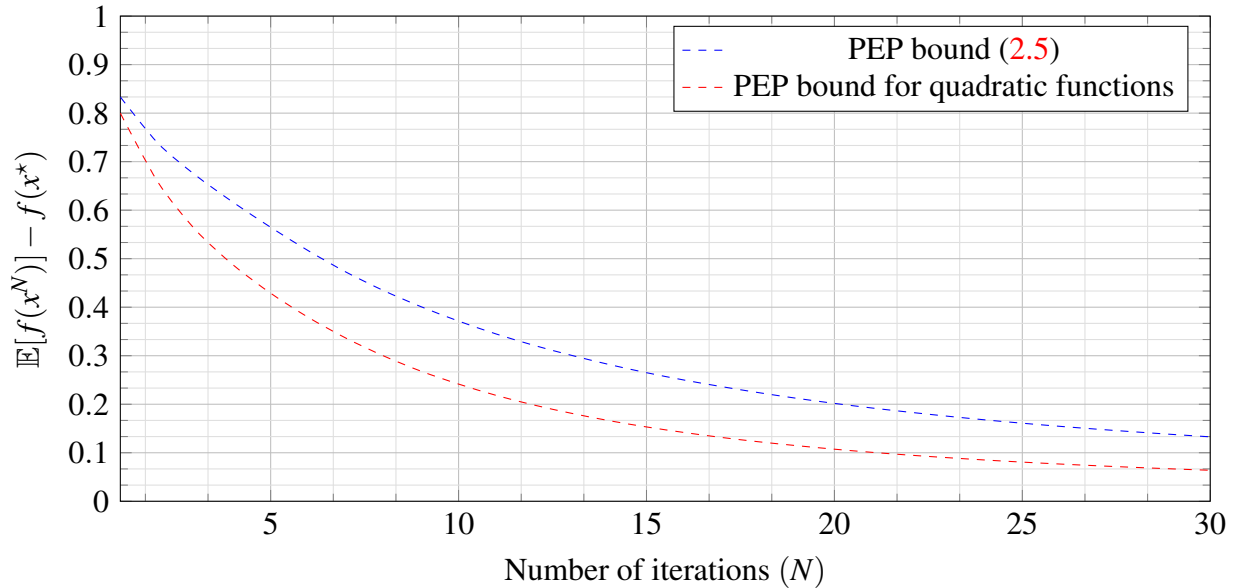

FIGURE 3. Convergence rate for Algorithm 2 computed by performance estimation problem for quadratic functions (red line) and the bound given by (2.5) (blue line) for $n = 10, L = 2, l_{\max} = 1, t = 1, \Delta = 1$.
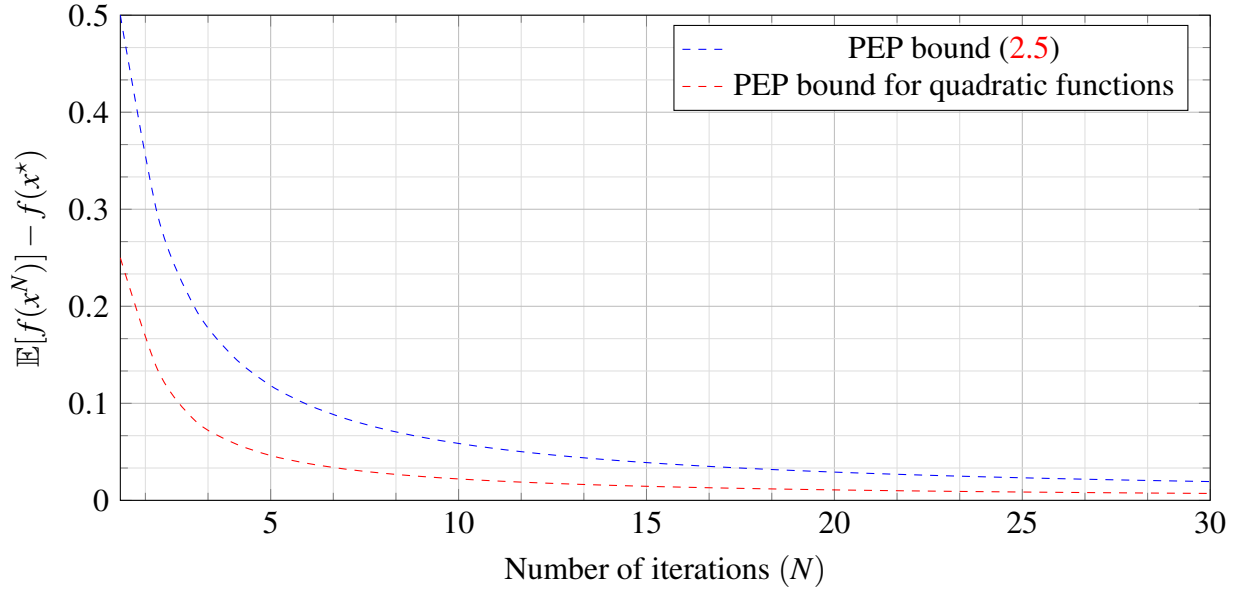
FIGURE 4. Convergence rate for Algorithm 2 computed by performance estimation problem for quadratic functions (red line) and the bound given by (2.5) (blue line) for $n = 2, L = 2, l_{\max} = 1, t = 1, \Delta = 1$.

Note that the convergence rate for the quadratic problem is sightly better than that of the general case.

2.3. The case of μ-strongly convex L-smooth functions.

In this section, we study the convergence rate of the μ-strongly convex L-smooth functions. If $\mu > 0$ the optimal value of problem (2.5) for one iteration of Algorithm 2, i.e. $N = 1$, appears to be the same as the following bound (2.8) given by Wright [2].

Theorem 2.2. [2, Theorem 1] *Let $f \in \mathscr{F}_{\mu,L}(\mathbb{R}^n)$. If $t_k = \frac{1}{\ell_{\max}}$ for each k, and $\mu > 0$, then, for all $N > 0$,*

$$\mathbb{E}\left[f(x^N)\right] - f^\star \leq \left(1 - \frac{\mu}{n\ell_{\max}}\right)^N \left(f(x^0) - f^\star\right). \tag{2.8}$$

The fact that the two bounds seem to coincide does not suggest that (2.8) is tight since the SDP bound (2.5) is a relaxation, and not exact.

3. CYCLIC COORDINATE DESCENT

Cyclic coordinate descent is one of the most important coordinate descent algorithms due to its simplicity. The convergence rate of cyclic coordinate descent method for the class of L-smooth convex functions is studied by Kamri et al using the performance estimation method [4]. This method is described in Algorithm 3.

Algorithm 3 Cyclic coordinate descent

Set number of cycles K, $\{t_k\}_{k=0}^{N-1}$ (step lengths), pick $x^0 \in \mathbb{R}^n$ and set $N = nK$.
For $k = 0, 1, 2, \ldots, N-1$ perform the following steps:
(1) Set $i = k \pmod{n} + 1$
(2) $x^{k+1} = x^k - t_k [\nabla f(x^k)]_i e_i$.

In each iteration, the method updates the current point over one of the coordinates in cyclic order.

The following result is known about the rate of convergence. We present it as in [2], but it is originally due to Beck and Tetruashvili [14].

Theorem 3.1. [2, Theorem 3] *Let $f : \mathbb{R}^n \to \mathbb{R}$ is an L-smooth convex function for some $L > 0$. If $t_k = \frac{1}{\ell_{\max}}$ for all k, then, for $k = n, 2n, 3n, \ldots$,*

$$\left(f(x^k) \right) - f^\star \leq \left(\frac{4nR_0^2 \ell_{\max}(1 + nL^2/\ell_{\max}^2)}{k+8} \right), \tag{3.1}$$

where R_0 satisfies $\max_{x^\star \in \mathbb{S}} \max_x \{ \|x - x^\star\| : f(x) \leq f(x^0) \} \leq R_0$ and $\mathbb{S}$ denotes the optimal solution set. If f is also strongly convex with parameter $\mu > 0$, then one has, for $k = n, 2n, 3n, \ldots$,

$$\left(f(x^k) \right) - f^\star \leq \left(1 - \frac{\mu}{2\ell_{\max}(1 + nL^2/\ell_{\max}^2)} \right)^{k/n} \left(f(x^0) - f^\star \right).$$

For easy reference, we recall the interpolation conditions from Theorem 1.1 in the case that $\mu = 0$: The set $\{x^i, \nabla f(x^i), f(x^i)\}$ for $i \in \{0, 1, \cdots, N, \star\}$ is $\mathscr{F}_{0,L}-$ interpolable if and only if

$$f(x^i) \geq f(x^j) + \langle \nabla f(x^j), x^i - x^j \rangle + \frac{1}{2L} \left\| \nabla f(x^i) - \nabla f(x^j) \right\|^2, \quad \forall i, j \in \{0, 1, \cdots, N, \star\}. \tag{3.2}$$

Using these conditions, we may formulate the worst-case convergence rate as performance estimation problem.

$$\begin{aligned}
\max \ & f(x^N) - f(x^\star) \\
\text{s.t. } & \{(x^i; \nabla f(x^i); f(x^i))\} \text{ satisfy (3.2) for } i \in \{0, 1, \cdots, N, \star\} \text{ w.r.t. } L, \\
& \|x^0 - x^\star\|^2 \leq \Delta, \\
& x^k \ (k \in \{1, 2, \cdots, N\}) \text{ are generated using Algorithm 3}, \\
& x^0 \in \mathbb{R}^n, \ \nabla f(x^\star) = 0.
\end{aligned} \tag{3.3}$$

Problem (3.3) can be relaxed to a semidefinite programming problem, and this is precisely what was done by Kamri et al. [4].

Since the univariate function $t \mapsto f(x^k + te_i)$ is convex and ℓ_i-smooth, it follows from (3.2) that, for every two consecutive points x^k and x^{k+1} generated by Algorithm 3, the following inequalities hold if $i = k \pmod{n} + 1$:

$$f(x^k) \geq f(x^{k+1}) + \nabla f(x^{k+1})_i(x_i^k - x_i^{k+1}) + \frac{1}{2\ell_i}(\nabla f(x^k)_i - \nabla f(x^{k+1})_i)^2, \qquad (3.4)$$
$$f(x^{k+1}) \geq f(x^k) + \nabla f(x^k)_i(x_i^{k+1} - x_i^k) + \frac{1}{2\ell_i}(\nabla f(x^{k+1})_i - \nabla f(x^k)_i)^2.$$

By adding the above inequalities to (3.3) one can obtain a better upper bound for the worst-case convergence rate, i.e.

$$\max f(x^N) - f(x^\star)$$
$$\text{s.t. } \{(x^i; \nabla f(x^i); f(x^i))\} \text{ satisfy } (3.2) \text{ for } i \in \{1, \cdots, N, \star\} \text{ w.r.t. } L,$$
$$\{(x^i; \nabla f(x^i); f(x^i))\} \text{ satisfy } (3.4) \text{ for } i \in \{1, \cdots, N, \star\} \text{ w.r.t. } \{\ell_1, \cdots, \ell_n\},$$
$$\|x^0 - x^\star\|^2 \leq \Delta, \qquad (3.5)$$
$$x^k \ (k \in \{1, \cdots, N\}) \text{ are generated using Algorithm } 3,$$
$$x^0 \in \mathbb{R}^n, \ \nabla f(x^\star) = 0.$$

In order to obtain an SDP relaxation to (3.5), we proceed in the same way as Kamri et al. [4]. We view (3.5) as a quadratically constrained quadratic program (QCQP) in variables corresponding to the unknowns

$$x_i^k, \ \frac{\partial f(x_i^k)}{\partial x_i}, \ f(x^k) \quad k \in \{0, 1, 2, \cdots, N\}, \ i \in \{1, \ldots, n\},$$

Next, we use the following relations to eliminate variables:

$$x_i^{k+1} - x_i^k = \begin{cases} -t_k \frac{\partial f(x_i^k)}{\partial x_i} & \text{if } i = k \pmod{n} + 1, \\ 0 & else, \end{cases}$$

which hold for all $k \in \{0, 1, 2, \cdots, N-1\}$, and $i \in \{1, \ldots, n\}$. Subsequently we form the standard Shor SDP relaxation (see e.g. [15]) of the resulting QCQP. Note that this is different to the approach we followed for randomized coordinate descent. In particular, the size of the SDP relaxation now depends on n, which was not the case before. This also limits the parameter values for which we may solve the SDP relaxations.

In Figure 5, we compare the SDP upper bounds from (3.3) and (3.5) for various parameter values.

The figure demonstrates that the bound can be improved slightly by adding the set of constraints (3.4) to the model provided by Kamri et al. [4]. Moreover, we add the constraint which correspond to the quadratic functions (2.7) to the model (3.5) which provides us with a better bound for quadratic functions. The computed values are much better that the theoretical bound (3.1), to the extent that we do not include this bound in the plot. Indeed, Kamri et al. [4] already mentioned in their paper that the computed values for their model are much better than the theoretical bound (3.1).
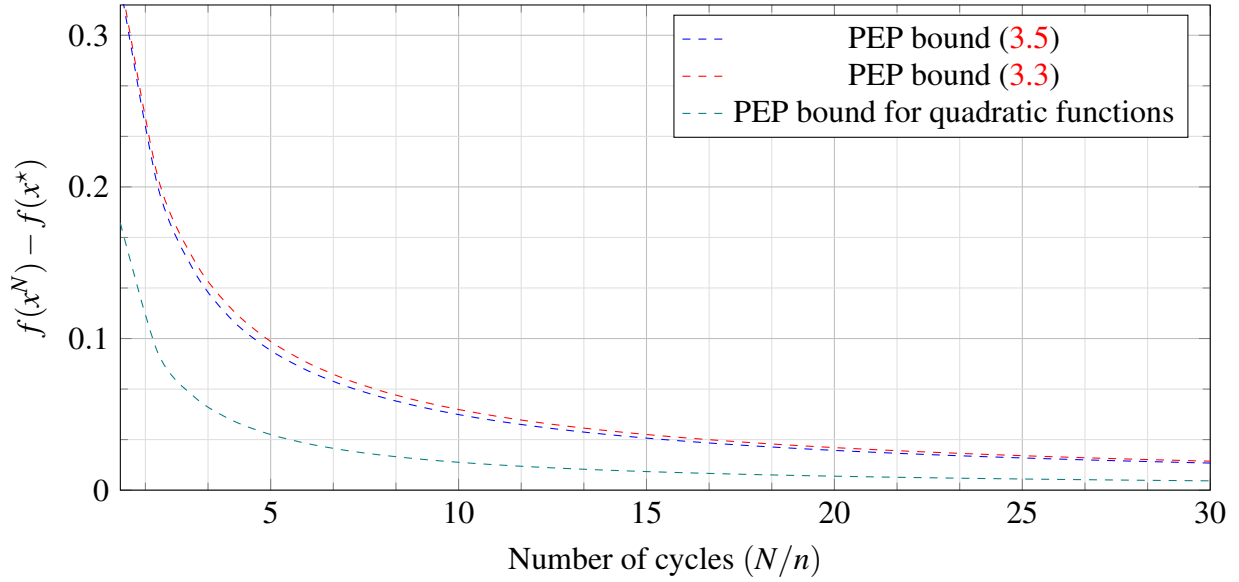
FIGURE 5. Convergence rate Algorithm 3 computed by performance estimation problem (3.5) (blue), the bound given by (3.3) (red) and the bound for quadratic functions (green) for $n = 2, L = 2, \ell_1 = 1, \ell_2 = 1, t = 0.5, \Delta = 1$.
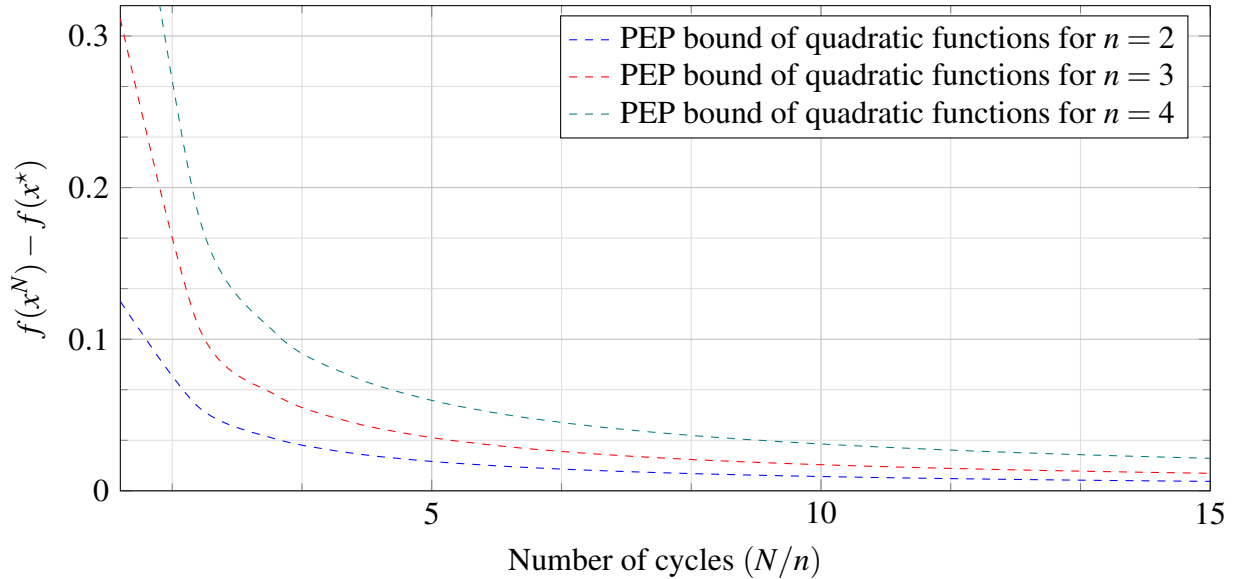

FIGURE 6. Convergence rate for Algorithm 3 computed by performance estimation problem of quadratic functions for $\ell_i = 1$ $i \in \{1, \cdots, n\}, L = \sum_{i=1}^{n} \ell_i, t = 1, \Delta = 1$ and different n.

Note that our discussion for coordinate-wise cyclic coordinate descent in this section could be extended to block-wise cyclic coordinate descent in a similar way as was done by Kamri et al. [4].

3.1. **Relation to the Gauss-Seidel method.** The minimization of the convex quadratic function in (2.6) is equivalent to the solution of the linear system $Ax = b$. Here, we may assume w.l.o.g. that A has a positive diagonal. Cyclic coordinate decent for problem (2.6) is closely related to the iterative Gauss-Seidel method for solving this linear system. For this reason, cyclic coordinate descent is sometimes also referred to as *nonlinear Gauss-Seidel*. It is therefore an interesting question whether the SDP performance estimation framework yields any new insights on the performance of the Gauss-Seidel method.

Denoting $A = (a_{ij})$, the iterative Gauss-Seidel method may be described as follows.

Algorithm 4 Gauss-Seidel method

Set N and pick $x^0 \in \mathbb{R}^n$.
For $k = 0, 1, \ldots, N-1$ perform the following:
$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k \right) \qquad (i = 1, \ldots, n).$$

This is exactly cyclic coordinate descent with unit step lengths if the gradient at a point x is replaced by $D^{-1} \nabla f(x)$, where $f(x) = \frac{1}{2} x^\top A x - b^\top x$ as before, and D is the diagonal matrix with the same diagonal entries as A. To see this, recall that the Fréchet derivative of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is the unique linear operator, say $D_f(x) : \mathbb{R}^n \to \mathbb{R}$, such that

$$\lim_{\|h\| \to 0} \frac{f(x+h) - f(x) - D_f(x)h}{\|h\|} = 0.$$

Once an inner product on $\mathbb{R}^n$ is fixed, say $\langle \cdot, \cdot \rangle$, one may, by the Riesz representation theorem, express $D_f(x)h = \langle g(x), h \rangle$, where $g(x)$ is called the gradient vector of f at x with respect to $\langle \cdot, \cdot \rangle$. In particular, if $\langle \cdot, \cdot \rangle$ is the Euclidean dot product, then $g(x) = \nabla f(x)$. If one changes to the inner product $\langle \cdot, \cdot \rangle_D$ defined by

$$\langle u, v \rangle_D = \sum_{i=1}^{n} a_{ii} u_i v_i \qquad (u, v \in \mathbb{R}^n), \tag{3.6}$$

then the gradient vector at x becomes $D^{-1} \nabla f(x)$, by the uniqueness of the Fréchet derivative.

It was shown in [7] that the interpolation condition in Theorem 1.1 holds for any reference inner product $\langle \cdot, \cdot \rangle$, provided that the gradient vector is interpreted accordingly.

In other words, the following SDP performance estimation problem gives a bound on the worst-case performance of the Gauss-Sidel method after N iterations, when A is a symmetric positive semidefinite matrix with a positive diagonal.

$$\max f(x^N) - f(x^\star)$$

s.t. $\{(x^i; \nabla f(x^i); f(x^i))\}$ satisfy (3.2) and (2.7) for $i \in \{0, 1, \cdots, N, \star\}$ w.r.t. $L = \lambda_{\max}(D^{-1}A)$,

$\qquad \{(x^i; \nabla f(x^i); f(x^i))\}$ satisfy (3.4) for $i \in \{0, 1, \cdots, N, \star\}$ w.r.t. $\ell_1 = \ldots = \ell_n = 1$,

$\qquad \|x^0 - x^\star\|^2 \leq \Delta$, $\tag{3.7}$

$\qquad x^k \ (k \in \{1, 2, \cdots, N\})$ is generated using Algorithm 3,

$\qquad x^0 \in \mathbb{R}^n, \ \nabla f(x^\star) = 0,$

where the inner product is now understood to be the one in (3.6), and the norm is the induced norm for this inner product, and $\lambda_{\max}(D^{-1}A)$ denotes the largest eigenvalue of $D^{-1}A$. (Note that the eigenvalues of $D^{-1}A$ are real.) Importantly, the reference inner product is not visible in the SDP performance estimation problem reformulation of (3.7), since only a Gram matrix for this inner product appears. It is therefore equally valid, for any inner product, provided that the inner product and norm are interpreted accordingly. Of course, the Lipschitz constants like (1.2) depend on the norm as well. It is easy to verify that, for the inner product (3.6), and $f(x) = \frac{1}{2}x^\top Ax - b^\top x$, one has $\ell_1 = \ldots = \ell_n = 1$ and $L = \lambda_{\max}(D^{-1}A)$ as is used in (3.7).

In summary, we have shown the following.

Theorem 3.2. *Consider a solvable system of linear equations $Ax = b$ where A is a symmetric positive semidefinite matrix with positive diagonal, and let $x^\star$ denote a solution. Letting $f(x) = \frac{1}{2}x^\top Ax - b^\top x$, after N iterations of the Gauss-Seidel method, an upper bound on $f(x^N) - f(x^\star)$ is given by the optimal value of the SDP problem* (3.7), *provided that the starting point x^0 satisfies $\|x^0 - x^\star\| \le \Delta$ for a given Δ, where the norm is the induced norm of the inner product* (3.6).

The Gauss-Seidel method is known to be convergent when A is symmetric positive-definite, e.g. [16, Theorem 10.1.2], or strictly or irreducibly diagonally dominant, e.g. [17]. The case when A is only positive semidefinite (with positive diagonal) seems to be less well-understood, and our approach sheds more light on this case. In particular, numerical results of the type shown in Figure 6 apply here.

4. CONCLUSION

We studied SDP performance estimation approaches to analyse randomized and cyclic coordinate descent, thereby complementing recent results in [4]. For randomized coordinate descent, we gave the first known SDP performance estimation bound. For cyclic coordinate descent, we were able to improve slightly on the numerical values given in [4], and we also discussed the link with the Gauss-Seidel method in the case of convex quadratic functions. Of course, to obtain new rates of convergence in general, it is necessary to solve the SDP performance estimation problems analytically, as opposed to numerically, but we have been unable to obtain analytic solutions for the SDP problems presented in this paper.

REFERENCES

[1] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM J. Optim. 22 (2012), 341-362.

[2] S.J. Wright, Coordinate descent algorithms, Math. Program. 151 (2015), 3-34.

[3] A.B. Taylor, F. Bach, Stochastic first-order methods: Non-asymptotic and computer-aided analyses via potential functions, In: Proceedings of Machine Learning Research, 32nd Annual Conference on Learning Theory, 99 (2019), 1-59.

[4] Y. Kamri, J.M. Hendrickx, F. Glineur, On the worst-case analysis of cyclic coordinate-wise algorithms on smooth convex functions, arXiv preprint arXiv:2211.17018, 2022.

[5] Y. Drori, M. Teboulle, Performance of first-order methods for smooth convex minimization: A novel approach, Math. Program. 145 (2014), 451-482.

[6] A.B. Taylor, J.M. Hendrickx, F. Glineur, Smooth strongly convex interpolation and exact worst-case performance of first-order methods, Math. Program. 161 (2017), 307-345.

[7] E. De Klerk, F. Glineur, A.B. Taylor, Worst-case convergence analysis of inexact gradient and Newton methods through semidefinite programming performance estimation, SIAM J. Optim. 30 (2020), 2053-2082.

[8] Y. Drori, O. Shamir, The complexity of finding stationary points with stochastic gradient descent, In: International Conference on Machine Learning, pp. 2658-2667, PMLR, 2020.

[9] H. Abbaszadehpeivasti, E. de Klerk, M. Zamani, On the rate of convergence of the difference-of-convex algorithm (DCA), arXiv preprint arXiv:2109.13566, 2021.

[10] T. Rotaru, F. Glineur, P. Panagiotis, Tight convergence rates of the gradient method on hypoconvex functions, arXiv preprint arXiv:2203.00775, 2022.

[11] H. Wolkowicz, R. Saigal, L. Vandenberghe, Handbook of Semidefinite Programming: Theory, Algorithms, and Applications, vol. 27, Springe, 2012.

[12] MOSEK ApS, The MOSEK optimization toolbox for MATLAB manual, Version 9.0., 2019.

[13] J. Löfberg, YALMIP : A toolbox for modeling and optimization in MATLAB,In: In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[14] A. Beck, L. Tetruashvili, On the convergence of block coordinate descent type methods, SIAM J. Optim. 23 (2013), 2037-2060.

[15] A.L. Wang, F. Kilinc-Karzan, A geometric view of SDP exactness in QCQPs and its applications, arXiv preprint arXiv:2011.07155, 2020.

[16] G.H. Golub, C.F. Van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, 2nd edition, 1989.

[17] R. Bagnara, A unified proof for the convergence of Jacobi and Gauss–Seidel methods, SIAM Rev. 37 (1995), 93-97.