Computational Optimization and Applications (2019) 73:411–452
https://doi.org/10.1007/s10589-019-00082-0

An almost cyclic 2-coordinate descent method for singly linearly constrained problems

Andrea Cristofari[1]

Received: 15 June 2018 / Published online: 25 February 2019

Abstract
A block decomposition method is proposed for minimizing a (possibly non-convex) continuously differentiable function subject to one linear equality constraint and simple bounds on the variables. The proposed method iteratively selects a pair of coordinates according to an almost cyclic strategy that does not use first-order information, allowing us not to compute the whole gradient of the objective function during the algorithm. Using first-order search directions to update each pair of coordinates, global convergence to stationary points is established for different choices of the stepsize under an appropriate assumption on the level set. In particular, both inexact and exact line search strategies are analyzed. Further, linear convergence rate is proved under standard additional assumptions. Numerical results are finally provided to show the effectiveness of the proposed method.

Keywords Block coordinate descent methods · Block decomposition methods · Linear convergence rate · SVM

Mathematics Subject Classification 90C06 · 90C30 · 65K05

1 Introduction

Block coordinate descent methods, also known as block decomposition methods, are algorithms that iteratively update a suitably chosen subset of variables, usually referred to as *working set*, trough an appropriate *optimization step*. Numerous variants of block coordinate descent methods have been proposed in the literature that essentially differ from each other in two aspects: the working set selection and the optimization step (see, e.g., [31] and the references therein for an overview of block coordinate descent methods in unconstrained optimization). In the last decades, block coordinate descent

✉ Andrea Cristofari
andrea.cristofari@unipd.it

[1] Department of Mathematics, University of Padua, Via Trieste, 63, 35121 Padua, Italy

methods gained great popularity, especially to address large structured problems, such as those arising in machine learning, where classical algorithms may not be so efficient and, sometimes, even not applicable for computational reasons. Moreover, block coordinate descent methods are well suited for parallelization, allowing to exploit modern computer architectures.

In this paper, we are concerned with the minimization of a continuously differentiable function subject to one linear equality constraint and simple bounds on the variables. Many relevant problems can be formulated in this way, such as, e.g., Support Vector Machine training, the continuous quadratic knapsack problem, resource allocation problems, the page rank problem, the Chebyshev center problem and the coordination of multi-agent systems.

In the literature, most of the block coordinate descent methods proposed for this class of problems use gradient information to identify a subset of variables that violate some optimality condition and guarantee, once updated, a certain decrease in the objective function [1,9,11,13,14,24,26]. Other methods select variables in order to satisfy an appropriate descent condition with a decreasing tolerance [12], or follow a Gauss–Seidel (cyclic) strategy [16]. Different working set selection rules, based on sufficient predicted descent, have also been studied in [30]. Moreover, a Jacobi-type algorithm has been devised in [15] and a class of parallel decomposition methods for quadratic objective functions has been proposed in [18].

In addition to the above algorithms, different versions of random coordinate descent methods have been proposed in [19–21,25,28], characterized by the fact that the working set is randomly chosen from a given probability distribution. From a theoretical point of view, random coordinate descent methods have good convergence properties, given in expectation, and they turn out to be efficient also in practice. In particular, since random coordinate descent methods do not use first-order information to choose the working set, the whole gradient of the objective function does not need to be computed during the iterations, leading to good performances when the objective function has cheap partial derivatives.

In our method, a working set of two coordinates is iteratively chosen according to the following almost cyclic strategy: one coordinate is selected in a cyclic manner, while the other one is obtained by considering the distance of each variable from its nearest bound in some points produced by the algorithm. We see that this working set selection rule does not use first-order information. So, similarly as in random coordinate descent methods, the whole gradient of the objective function does not need to be computed during the algorithm and high efficiency is still achieved when the partial derivatives of the objective function are cheap. Anyway, differently from random coordinate descent methods, the proposed algorithm has deterministic convergence properties.

More in detail, once a pair of coordinates is selected in the working set, our algorithm performs a minimization step by moving along a first-order search direction with a certain stepsize. We first give a general condition on the stepsize that guarantees global convergence to stationary points, under the assumption that every point of the level set has at least one component strictly between the lower and the upper bound. Note that this assumption is automatically satisfied in many cases: e.g., when the feasible set is the unit simplex, or when at least one variable has no bounds. Then, we describe some practical ways to compute the stepsize, considering different classes of objective functions: the Armijo line search can be used for general non-convex objective functions, overestimates of the Lipschitz constant can be used for objective functions with Lipschitz continuous gradient, and the exact line search can be used when the objective function is strictly convex.

We also show that the proposed method converges linearly under standard additional assumptions. In particular, two different results are given: asymptotic linear convergence rate is proved when there are finite bounds on some (or all) of the variables, while non-asymptotic linear convergence rate is proved when there are no bounds on the variables.

Lastly, experimental simulations performed on different classes of test problems show promising results of the proposed algorithm in comparison with other block coordinate descent methods.

The rest of the paper is organized as follows. In Sect. 2, we introduce the notation and recall some preliminary results. In Sect. 3, we present the algorithm and carry out the convergence analysis. In Sect. 4, we describe some practical line search strategies. In Sect. 5, we analyze the convergence rate of the algorithm. In Sect. 6, we show the numerical results. Finally, we draw some conclusions in Sect. 7.

2 Preliminaries and notation

Let us introduce the notation. Given a vector $x \in \mathbb{R}^n$, we indicate by x_i the ith entry of x. We denote by $e_i \in \mathbb{R}^n$ the vector made of all zeros except for the ith entry that is equal to 1. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient of f is indicated by ∇f, the ith partial derivative of f is indicated by $\nabla_i f$ and the Hessian matrix of f is indicated by $\nabla^2 f$. The derivative of a function $f : \mathbb{R} \to \mathbb{R}$ is denoted by $\dot{f}$. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is indicated by $\|x\|$.

Throughout the paper, we focus on the following singly linearly constrained problem with lower and upper bounds on the variables:

$$\begin{aligned} &\min \ f(x) \\ &\sum_{i=1}^{n} x_i = b \\ &l_i \le x_i \le u_i, \quad i = 1, \dots, n, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $b \in \mathbb{R}$ and, for all $i = 1, \dots, n$, we have $l_i < u_i$, with $l_i \in \mathbb{R} \cup \{-\infty\}$ and $u_i \in \mathbb{R} \cup \{\infty\}$. By slight abuse of standard mathematical notation, we allow variable bounds to be infinite.

Note that every problem of the form

$$\min \{\omega(s) \colon \sum_{i=1}^{n} a_i s_i = b, \ \bar{l}_i \le s_i \le \bar{u}_i, \ i = 1, \dots, n\}$$

with $\omega \colon \mathbb{R}^n \to \mathbb{R}$, $b \in \mathbb{R}$ and $a_i \neq 0$, $\bar{l}_i < \bar{u}_i$, $i = 1, \dots, n$, can be rewritten as in (1) via the following variable transformation: $x_i = a_i s_i$, $i = 1, \dots, n$, thus considering the objective function $f(x):=\omega\big((x_1/a_1),\dots,(x_n/a_n)\big)$ and setting the lower and the upper bound on x according to the above transformation.

From now on, we denote the feasible set of problem (1) by $\mathcal{F}$. Throughout the paper, we assume that $\mathcal{F}\neq\emptyset$. Moreover, the terms *variable* and *coordinate* will be used interchangeably to indicate each x_i, $i=1,\dots,n$.

Finally, let us recall the following characterization of stationary points of problem (1), which can be easily derived from KKT conditions.

Proposition 1 *A feasible point x^* of problem* (1) *is stationary if and only if there exists $\lambda^*\in\mathbb{R}$ such that, for all $i=1,\dots,n$,*

$$\nabla_i f(x^*)\begin{cases}\ge\lambda^*, & \textit{if } x_i^*=l_i,\\ =\lambda^*, & \textit{if } x_i^*\in(l_i,u_i),\\ \le\lambda^*, & \textit{if } x_i^*=u_i.\end{cases} \tag{2}$$

3 The almost cyclic 2-coordinate descent (AC2CD) method

In this section, we present the algorithm for solving problem (1) and we analyze its convergence properties to stationary points.

3.1 Description of the algorithm

The proposed almost cyclic 2-coordinate descent (AC2CD) method is a block decomposition method that iteratively performs a minimization step with respect to a working set of two coordinates, chosen by an almost cyclic strategy (note that two is the smallest number of variables that can be updated to maintain feasibility). A remarkable feature of AC2CD is that the working set selection rule does not use first-order information, allowing us not to compute the whole gradient of the objective function during the algorithm.

To describe the proposed method, we have to distinguish between *outer iterations*, indicated by the integer k, and *inner iterations*, indicated by the pair of integers (k,i), where $k=0,1,\dots$ and $i=1,\dots,n$. Each outer iteration k starts with a feasible point, denoted by x^k, and has n inner iterations $(k,1),\dots,(k,n)$. Each inner iteration (k,i) starts with a feasible point, denoted by $z^{k,i}$, and produces the successive (feasible) $z^{k,i+1}$ by performing a minimization step with respect to a suitably chosen pair of coordinates. Outer and inner iterations are linked by the following relation:

$$z^{k,1}=x^k \quad\text{and}\quad x^{k+1}=z^{k,n+1},\quad \forall\, k\ge 0.$$

Namely, each cycle of inner iterations $(k,1),\dots,(k,n)$ starts from x^k and returns the successive x^{k+1}.

To be more specific, given any x^k produced by the algorithm, first we choose a variable index $j(k)\in\{1,\dots,n\}$ (as to be described later). Then, in the cycle of inner iterations $(k,1),\dots,(k,n)$, we adopt the following rule to choose the working set:

one variable index is selected in a cyclic manner, following an arbitrary order with no repetition, while the second variable index remains fixed and equal to $j(k)$. Since only one variable index is selected in a cyclic manner, we name this working set selection rule *almost cyclic*.

From now on, for every inner iteration (k, i) we denote by p_i^k the variable index of the working set that is selected in a cyclic manner. So, each minimization step is performed with respect to the two coordinates $z_{p_i^k}^{k,i}$ and $z_{j(k)}^{k,i}$. In this minimization step, we compute the first-order search direction $d^{k,i} = [\nabla_{j(k)} f(z^{k,i}) - \nabla_{p_i^k} f(z^{k,i})](e_{p_i^k} - e_{j(k)})$. Clearly, we have $d_h^{k,i} = 0$ for all $h \in \{1, \dots, n\} \setminus \{p_i^k, j(k)\}$. Then, we set

$$z^{k,i+1} = z^{k,i} + \alpha^{k,i} d^{k,i},$$

where $\alpha^{k,i} \in \mathbb{R}$ is a feasible stepsize (which will be described later on).

Now, we explain how to choose the index $j(k)$ at the beginning of an outer iteration k. Roughly speaking, $x_{j(k)}^k$ must be "sufficiently far" from its nearest bound. Formally, for each index $h \in \{1, \dots, n\}$, let us define the operator $D_h \colon \mathcal{F} \to [0, \infty) \cup \{\infty\}$ that takes as input a feasible point x and returns the distance of x_h from its nearest bound:

$$D_h(x) := \min\{x_h - l_h, u_h - x_h\}. \tag{3}$$

So, for a given x^k, we choose $j(k)$ as any index that satisfies

$$D_{j(k)}(x^k) \geq \tau D^k, \tag{4}$$

where $\tau \in (0, 1]$ is a fixed parameter and

$$D^k := \max_{h=1,\dots,n} D_h(x^k). \tag{5}$$

In other words, the distance between $x_{j(k)}^k$ and its nearest bound must be greater than or equal to a certain fraction (equal to τ) of the maximum distance between each component of x^k and its nearest bound.

In Algorithm 1, we report the scheme of AC2CD. Hereinafter, we indicate

$$g^{k,i} := \nabla_{j(k)} f(z^{k,i}) - \nabla_{p_i^k} f(z^{k,i}) \qquad \text{and} \qquad d^{k,i} := g^{k,i}(e_{p_i^k} - e_{j(k)}), \tag{6}$$

as they are defined at steps 6 and 7 of Algorithm 1. Note that

$$\nabla f(z^{k,i})^T d^{k,i} = -[\nabla_{j(k)} f(z^{k,i}) - \nabla_{p_i^k} f(z^{k,i})]^2 = -(g^{k,i})^2, \tag{7}$$

i.e., $d^{k,i}$ is a descent direction at $z^{k,i}$ if $g^{k,i} \neq 0$ (equivalently, every nonzero $d^{k,i}$ is a descent direction at $z^{k,i}$).

Remark 1 For the sake of simplicity, within each outer iteration k of AC2CD we consider n inner iterations, but they actually are $n - 1$, since no pair of coordinates is updated when $p_i^k = j(k)$.

Algorithm 1 `Almost Cyclic 2-Coordinate Descent (AC2CD) method`

```
0  Given $x^0 \in \mathcal{F}$ and $\tau \in (0, 1]$
1  For $k = 0, 1, \dots$
2      Choose a variable index $j(k) \in \{1, \dots, n\}$ that satisfies (4)
3      Choose a permutation $\{p_1^k, \dots, p_n^k\}$ of $\{1, \dots, n\}$
4      Set $z^{k,1} = x^k$
5      For $i = 1, \dots, n$
6          Let $g^{k,i} = \nabla_{j(k)} f(z^{k,i}) - \nabla_{p_i^k} f(z^{k,i})$
7          Compute the search direction $d^{k,i} = g^{k,i}(e_{p_i^k} - e_{j(k)})$
8          Compute a feasible stepsize $\alpha^{k,i}$ and set $z^{k,i+1} = z^{k,i} + \alpha^{k,i} d^{k,i}$
9      End for
10     Set $x^{k+1} = z^{k,n+1}$
11 End for
```

Now, let us focus on the computation of the stepsize $\alpha^{k,i}$ (step 8 of Algorithm 1). First, for any inner iterate $z^{k,i}$ we define $\bar{\alpha}^{k,i}$ as the largest feasible stepsize along the direction $d^{k,i}$. Since the equality constraint is clearly satisfied by the choice of $d^{k,i}$, by simple calculations we obtain

$$\bar{\alpha}^{k,i} = \begin{cases} (1/g^{k,i}) \, \min\{u_{p_i^k} - z^{k,i}_{p_i^k}, z^{k,i}_{j(k)} - l_{j(k)}\}, & \text{if } g^{k,i} > 0, \\ (1/|g^{k,i}|) \, \min\{z^{k,i}_{p_i^k} - l_{p_i^k}, u_{j(k)} - z^{k,i}_{j(k)}\}, & \text{if } g^{k,i} < 0, \\ 0, & \text{if } g^{k,i} = 0. \end{cases} \tag{8}$$

In particular, the choice to define $\bar{\alpha}^{k,i} = 0$ when $g^{k,i} = 0$ is for convenience of exposition (and without loss of generality), since it implies that $\bar{\alpha}^{k,i} = 0$ if $d^{k,i} = 0$ (equivalently, $d^{k,i} \neq 0$ if $\bar{\alpha}^{k,i} > 0$).

A general rule for the computation of the stepsize $\alpha^{k,i}$ is stated in the following Stepsize Condition 1 (SC 1). As to be shown, this rule can be easily satisfied in practice by different line search strategies and guarantees global convergence to stationary points under an appropriate assumption on the level set.

SC (Stepsize Condition) 1 *It must hold that*

(i) $f(z^{k,i+i}) \le f(z^{k,i})$, *with* $z^{k,i+1} \in \mathcal{F}$, *for every inner iteration* (k,i)*;*

(ii) *if* $\{f(x^k)\}$ *converges, then* $\lim\limits_{k\to\infty} \|z^{k,i+1} - z^{k,i}\| = 0$, $i = 1,\dots,n$*;*

(iii) *for every pair of indices* $\hat{\imath}, \hat{\jmath} \in \{1,\dots,n\}$ *and every convergent subsequence* $\{z^{k,\hat{\imath}}\}_{K\subseteq\{0,1,\dots\}}$ *such that* $p^k_{\hat{\imath}}$ *is constant for all* $k \in K$ *and* $j(k) = \hat{\jmath}$ *for all* $k \in K$, *if a real number* $\xi > 0$ *exists such that* $\liminf\limits_{\substack{k\to\infty\\ k\in K}} \bar{\alpha}^{k,\hat{\imath}} = \xi$, *then*

$$\lim_{\substack{k\to\infty\\ k\in K}} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}} = 0.$$

Let us spend a few words on the meaning of SC 1. Point (i) requires that every inner iterate is feasible and does not increase the objective value. Point (ii) is a condition usually needed for convergence of block decomposition methods, requiring that the distance of two successive inner iterates goes to zero. Finally, point (iii) requires that the directional derivatives converge to zero over appropriate subsequences when, for sufficiently large k, the largest feasible stepsizes are bounded from below by a positive constant over the same subsequences.

Let us conclude this section by stating the following lemma, which will be useful in the sequel. It shows that, if points (i)–(ii) of SC 1 are satisfied, then every limit point of $\{x^k\}$ is a limit point of $\{z^{k,i}\}$ for every fixed $i = 1,\dots,n$.

Lemma 1 *Let* $\{x^k\}$ *be a sequence of points produced by AC2CD that satisfies points* (i)–(ii) *of SC* 1*, and let* $\{x^k\}_{K\subseteq\{0,1,\dots\}}$ *be a subsequence converging to* $\bar{x}$*. Then,*

$$\lim_{\substack{k\to\infty\\ k\in K}} z^{k,i} = \bar{x}, \quad i = 1,\dots,n.$$

Proof For $i = 1$ the result is trivial, since $z^{k,1} = x^k$ for all k. For any $i \in \{2,\dots,n\}$, we have $z^{k,i} - z^{k,1} = \sum_{t=1}^{i-1} z^{k,t+1} - z^{k,t}$, and then

$$\|z^{k,i} - x^k\| = \|z^{k,i} - z^{k,1}\| \le \sum_{t=1}^{i-1} \|z^{k,t+1} - z^{k,t}\|. \tag{9}$$

By continuity of f, we have $\{f(x^k)\}_K \to f(\bar{x})$. By point (i) of SC 1, we also have $f(x^{k+1}) = f(z^{k,n+1}) \le f(z^{k,n}) \le \dots \le f(z^{k,1}) = f(x^k)$ for all $k \ge 0$, and then, $\lim\limits_{k\to\infty} f(x^k) = f(\bar{x})$. Therefore, using point (ii) of SC 1, we can write $\lim\limits_{k\to\infty} \|z^{k,t+1} - z^{k,t}\| = 0$ for all $t = 1,\dots,n$. Combining this relation with (9), we get

$$\lim_{k\to\infty} \|z^{k,i} - x^k\| = 0. \tag{10}$$

Moreover, $\|z^{k,i} - \bar{x}\| = \|z^{k,i} - x^k + x^k - \bar{x}\| \le \|z^{k,i} - x^k\| + \|x^k - \bar{x}\|$. Then, the result is obtained by combining this inequality with (10) and the fact that $\{x^k\}_K \to \bar{x}$. □

3.2 Convergence to stationary points

In this subsection, we are concerned with the convergence analysis of AC2CD.

For a given starting point x^0, let us first define the level set

$$\mathcal{L}^0 := \{x \in \mathcal{F} \colon f(x) \le f(x^0)\}.$$

To prove global convergence of AC2CD to stationary points, we need the following assumption, which requires that every point of $\mathcal{L}^0$ has at least one component strictly between the lower and the upper bound.

Assumption 1 $\forall\, x \in \mathcal{L}^0,\ \exists\, i \in \{1, \dots, n\} \colon x_i \in (l_i, u_i)$.

Let us discuss the role played by this assumption, when combined with SC 1, in the convergence analysis of AC2CD. First, it guarantees that, for every outer iteration $k \ge 0$, at least one pair of coordinates can be updated if and only if x^k is non-stationary. Indeed, every x^k remains in $\mathcal{L}^0$ by point (i) of SC 1 and, by the rule used to choose the index $j(k)$, if Assumption 1 holds we have

$$l_{j(k)} < x^k_{j(k)} < u_{j(k)}, \quad \forall k \ge 0.$$

So, from the stationarity conditions (2), it is straightforward to verify that a variable index p^k_i exists such that $\nabla f(z^{k,i})^T d^{k,i} < 0$ with $\bar{\alpha}^{k,i} > 0$ if and only if x^k is non-stationary, that is, at least one pair of coordinates can be updated during the inner iterations if and only if x^k is non-stationary.

Second, as to be shown in the proof of Theorem 1, if Assumption 1 holds we can prove that $\{\nabla_{j(k)} f(x^k)\}$ converges, over certain subsequences, to the KKT multiplier λ^* defined as in (2). So, for sufficiently large k, we can measure the stationarity violation for each coordinate x^k_h by $\nabla_h f(x^k) - \nabla_{j(k)} f(x^k)$ and guarantee that, at the limit, each coordinate satisfies (2).

Let us also remark that Assumption 1 is automatically satisfied in many cases, for example when $\mathcal{F}$ is the unit simplex, or when at least one variable has no bounds, that is, if an index $i \in \{1, \dots, n\}$ exists such that $l_i = -\infty$ and $u_i = \infty$. Further, in [14], where the same assumption is used, it is shown that, for the Support Vector Machine training problem, this assumption is satisfied if the smallest eigenvalue of $\nabla^2 f$ is sufficiently large and the starting point x^0 is such that $f(x^0) < 0$ (see Appendix B in [14] for more details).

Now, we are ready to show that, under Assumption 1, AC2CD converges to stationary points.

Theorem 1 *Let Assumption 1 hold and let $\{x^k\}$ be a sequence of points produced by AC2CD that satisfies SC 1. Then, every limit point of $\{x^k\}$ is stationary for problem (1).*

Proof Let x^* be a limit point of $\{x^k\}$ and let $\{x^k\}_{K \subseteq \{0,1,\dots\}}$ be a subsequence converging to x^*. From the instructions of the algorithm, x^* is a feasible point. Moreover, from Lemma 1 we can write

$$\lim_{\substack{k\to\infty \\ k\in K}} z^{k,i} = x^*, \quad i = 1,\dots,n. \tag{11}$$

By continuity of f, we have that $\{f(x^k)\}_K$ converges to $f(x^*)$. Since $\{f(x^k)\}$ is monotonically non-increasing (by point (i) of SC 1) we have that $\{f(x^k)\}$ converges to $f(x^*)$. Therefore, by point (ii) of SC 1 it follows that

$$\lim_{k\to\infty} \|z^{k,i+1} - z^{k,i}\| = 0, \quad i = 1,\dots,n. \tag{12}$$

Using the fact that the set of indices $\{1,\dots,n\}$ is finite, there exist an index $\hat{\jmath} \in \{1,\dots,n\}$ and a further infinite subsequence, that we still denote by $\{x^k\}_K$ without loss of generality, such that

$$\lim_{\substack{k\to\infty \\ k\in K}} x^k = x^* \quad \text{and} \quad j(k) = \hat{\jmath}, \qquad \forall\, k \in K.$$

We first want to show that a real number $\rho > 0$ exists such that

$$\min\{z^{k,i}_{\hat{\jmath}} - l_{\hat{\jmath}}, u_{\hat{\jmath}} - z^{k,i}_{\hat{\jmath}}\} \ge \rho, \quad i = 1,\dots,n, \quad \forall\,\text{sufficiently large } k \in K. \tag{13}$$

To this extent, by Assumption 1 we have that an index $\bar{h} \in \{1,\dots,n\}$ exists such that $x^*_{\bar{h}} \in (l_{\bar{h}}, u_{\bar{h}})$. So, using the operator D_h defined as in (3), a positive real number ρ exists such that $D_{\bar{h}}(x^*) \ge (4/\tau)\rho$, where $\tau \in (0,1]$ is the parameter defined at step 0 of Algorithm 1, used to compute $j(k)$. Since $\{x^k\}_K \to x^*$, by continuity of D_h we have that $D_{\bar{h}}(x^k) \ge (2/\tau)\rho$ for all sufficiently large $k \in K$. Using the definition of D^k given in (5), we obtain

$$D^k \ge (2/\tau)\rho, \quad \forall\,\text{sufficiently large } k \in K.$$

By the rule used to choose the index $j(k)$, we can write

$$D_{\hat{\jmath}}(z^{k,1}) = D_{\hat{\jmath}}(x^k) \ge \tau D^k \ge 2\rho, \quad \forall\,\text{sufficiently large } k \in K.$$

The above relation, combined with (12), implies that $D_{\hat{\jmath}}(z^{k,i}) \ge \rho$, $i = 1,\dots,n$, for all sufficiently large $k \in K$. Equivalently, (13) holds.

Now, we want to show that the stationarity conditions (2) are satisfied at x^* with

$$\lambda^* = \nabla_{\hat{\jmath}} f(x^*). \tag{14}$$

Reasoning by contradiction, assume that this is not true. Then, there exists an index $\hat{\imath} \in \{1,\dots,n\}\backslash\{\hat{\jmath}\}$ that violates (2). Using again the fact that the set of indices

$\{1,\dots,n\}$ is finite, there also exist an index $\hat{\imath} \in \{1,\dots,n\}$ and a further infinite subsequence, that we still denote by $\{x^k\}_K$ without loss of generality, such that $p^k_{\hat{\imath}}$ is constant and equal to $\hat{t}$ for all $k \in K$. Since also $j(k)$ is constant (and equal to $\hat{\jmath}$) for all $k \in K$, we thus obtain

$$p^k_{\hat{\imath}} = \hat{t} \quad \text{and} \quad j(k) = \hat{\jmath}, \qquad \forall\, k \in K.$$

By (11) and the continuity of ∇f, it follows that $\{\nabla f(z^{k,i})\}_K$ is bounded for all $i = 1,\dots,n$. So, a real number T exists such that

$$\|\nabla f(z^{k,i})\| \le T, \quad i = 1,\dots,n, \quad \forall\, k \in K.$$

Therefore,

$$|g^{k,\hat{\imath}}| = |\nabla_{\hat{\jmath}} f(z^{k,\hat{\imath}}) - \nabla_{\hat{t}} f(z^{k,\hat{\imath}})| \le 2T, \quad \forall\, k \in K. \tag{15}$$

Since we have assumed $\hat{t}$ to violate (2) with λ^* defined as in (14), one of the following three cases must hold.

(i) $x^*_{\hat{t}} \in (l_{\hat{t}}, u_{\hat{t}})$ and $|\nabla_{\hat{\jmath}} f(x^*) - \nabla_{\hat{t}} f(x^*)| > 0$. Taking into account (11), a real number $\zeta > 0$ exists such that

$$\min\{z^{k,\hat{\imath}}_{\hat{t}} - l_{\hat{t}}, u_{\hat{t}} - z^{k,\hat{\imath}}_{\hat{t}}\} \ge \zeta, \quad \forall\, \text{sufficiently large } k \in K, \tag{16}$$

$$|g^{k,\hat{\imath}}| = |\nabla_{\hat{\jmath}} f(z^{k,\hat{\imath}}) - \nabla_{\hat{t}} f(z^{k,\hat{\imath}})| \ge \zeta, \quad \forall\, \text{sufficiently large } k \in K. \tag{17}$$

From (13), (15), (16), (17) and the definition of $\bar{\alpha}^{k,i}$ given in (8), we obtain

$$\bar{\alpha}^{k,\hat{\imath}} \ge \min\left\{\frac{\zeta}{2T}, \frac{\rho}{2T}\right\} > 0, \quad \forall\, \text{sufficiently large } k \in K.$$

Therefore, by point (iii) of SC 1 we get

$$0 = \lim_{\substack{k\to\infty \\ k\in K}} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}} = \lim_{\substack{k\to\infty \\ k\in K}} -\left[\nabla_{\hat{\jmath}} f(z^{k,\hat{\imath}}) - \nabla_{\hat{t}} f(z^{k,\hat{\imath}})\right]^2,$$

where the second equality follows from (7). We thus obtain a contradiction with (17).

(ii) $x^*_{\hat{t}} = l_{\hat{t}}$ and $\nabla_{\hat{t}} f(x^*) < \nabla_{\hat{\jmath}} f(x^*)$. Taking into account (11), we have

$$u_{\hat{t}} - z^{k,\hat{\imath}}_{\hat{t}} \ge \frac{u_{\hat{t}} - l_{\hat{t}}}{2}, \quad \forall\, \text{sufficiently large } k \in K, \tag{18}$$

and a real number $\zeta > 0$ exists such that

$$g^{k,\hat{\imath}} = \nabla_{\hat{\jmath}} f(z^{k,\hat{\imath}}) - \nabla_{\hat{t}} f(z^{k,\hat{\imath}}) \ge \zeta, \quad \forall\, \text{sufficiently large } k \in K. \tag{19}$$

From (13), (15), (18), (19) and the definition of $\bar{\alpha}^{k,i}$ given in (8), we obtain

$$\bar{\alpha}^{k,\hat{\imath}} \geq \min\left\{\frac{u_{\hat{\imath}} - l_{\hat{\imath}}}{4T}, \frac{\rho}{2T}\right\} > 0, \quad \forall \text{ sufficiently large } k \in K.$$

Therefore, by point (iii) of SC 1 we get

$$0 = \lim_{\substack{k\to\infty \\ k\in K}} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}} = \lim_{\substack{k\to\infty \\ k\in K}} -\big[\nabla_{\hat{\jmath}} f(z^{k,\hat{\imath}}) - \nabla_{\hat{\imath}} f(z^{k,\hat{\imath}})\big]^2,$$

where the second equality follows from (7). We thus obtain a contradiction with (19).

(iii) $x^*_{\hat{\imath}} = u_{\hat{\imath}}$ and $\nabla_{\hat{\imath}} f(x^*) > \nabla_{\hat{\jmath}} f(x^*)$. This is a verbatim repetition of the previous case, which again leads to a contradiction.

We can thus conclude that x^* is a stationary point. □

4 Computation of the stepsize

In this section, we describe some practical ways to compute the stepsize in order to satisfy SC 1. We consider different classes of objective functions: general non-convex functions, those with Lipschitz continuous gradient and strictly convex functions.

4.1 General non-convex objective functions

When the objective function is non-convex, a common way to compute the stepsize is using an inexact line search. Here, we consider the Armijo line search, which is now described. Given any inner iterate $z^{k,i}$ and the direction $d^{k,i}$, first we choose a trial feasible stepsize $\Delta^{k,i}$ and then we obtain $\alpha^{k,i}$ by a backtracking procedure. In particular, we set

$$\alpha^{k,i} = (\delta)^c \, \Delta^{k,i}, \tag{20}$$

where c is the smallest nonnegative integer such that

$$f(z^{k,i} + \Delta^{k,i}(\delta)^c d^{k,i}) \leq f(z^{k,i}) + \gamma \Delta^{k,i} (\delta)^c \, \nabla f(z^{k,i})^T d^{k,i} \tag{21}$$

and $\delta \in (0, 1)$, $\gamma \in (0, 1)$ are fixed parameters.

For what concerns the choice of $\Delta^{k,i}$, first it must be feasible, that is, $\Delta^{k,i} \leq \bar{\alpha}^{k,i}$. Moreover, as to be shown in the proof of the following proposition, $\Delta^{k,i}$ must be bounded from above by a finite positive constant to satisfy point (ii) of SC 1. Namely, we require that $\Delta^{k,i} \leq A^{k,i}$, where $A^{k,i}$ is any scalar satisfying $0 < A_l \leq A^{k,i} \leq A_u < \infty$, with A_l and A_u being fixed parameters. In conclusion, in the Armijo line search we set $\Delta^{k,i} = \min\{\bar{\alpha}^{k,i}, A^{k,i}\}$. In the next proposition, we show that a stepsize computed in this way satisfies SC 1.

Proposition 2 *Let $\delta \in (0,1)$, $\gamma \in (0,1)$ and $0 < A_l \le A_u < \infty$. Then, SC 1 is satisfied by computing, at every inner iteration (k,i), the stepsize $\alpha^{k,i}$ as in (20), where c is the smallest nonnegative integer such that (21) holds, $\Delta^{k,i} = \min\{\bar{\alpha}^{k,i}, A^{k,i}\}$ and $A^{k,i} \in [A_l, A_u]$.*

Proof We first observe that, at every inner iteration (k,i), there exists a nonnegative integer c satisfying (21), since, from (7), we have $\nabla f(z^{k,i})^T d^{k,i} < 0$ for all $d^{k,i} \neq 0$.

Point (i) of SC 1 immediately follows from (21) and the fact that $\alpha^{k,i} \le \Delta^{k,i}$, with $\Delta^{k,i}$ feasible. Now, we prove point (ii) of SC 1. We first show that a real number $\sigma > 0$ exists such that, for all $k \ge 0$, we have

$$f(z^{k,i+1}) \le f(z^{k,i}) - \sigma \|z^{k,i+1} - z^{k,i}\|^2, \quad i = 1,\dots,n. \tag{22}$$

From (20) and (21), for all $k \ge 0$ we can write

$$f(z^{k,i+1}) \le f(z^{k,i}) + \gamma \alpha^{k,i} \nabla f(z^{k,i})^T d^{k,i}, \quad i = 1,\dots,n. \tag{23}$$

Using (7), we obtain

$$f(z^{k,i+1}) \le f(z^{k,i}) - \gamma \alpha^{k,i} (g^{k,i})^2, \quad i = 1,\dots,n. \tag{24}$$

Moreover, $z^{k,i+1} - z^{k,i} = \alpha^{k,i} g^{k,i} (e_{p_i^k} - e_{j(k)})$, and then, $\|z^{k,i+1} - z^{k,i}\|^2 = 2(\alpha^{k,i})^2 (g^{k,i})^2$. Using this equality in (24), and recalling that $\alpha^{k,i} \le A_u < \infty$, we have that

$$f(z^{k,i+1}) \le f(z^{k,i}) - \frac{\gamma}{2A_u} \|z^{k,i+1} - z^{k,i}\|^2, \quad i = 1,\dots,n.$$

Therefore, (22) holds with $\sigma = \gamma/(2A_u)$ for all $k \ge 0$. So, point (ii) of SC 1 is satisfied by combining (22) with the fact that $f(x^{k+1}) = f(z^{k,n+1}) \le f(z^{k,n}) \le \dots \le f(z^{k,1}) = f(x^k)$ for all $k \ge 0$.

Now, we prove that also point (iii) of SC 1 holds. Let $\{z^{k,\hat{\imath}}\}_K$, $\hat{\imath}$, $\hat{\jmath}$ and $\bar{\alpha}^{k,\hat{\imath}}$ be defined as in point (iii) of SC 1. Rearranging the terms in (23), we can write

$$f(z^{k,\hat{\imath}}) - f(z^{k,\hat{\imath}+1}) \ge \gamma \alpha^{k,\hat{\imath}} |\nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}}|, \quad \forall k \in K. \tag{25}$$

Moreover, since $\{z^{k,\hat{\imath}}\}_K$ converges and $f(x^{k+1}) = f(z^{k,n+1}) \le f(z^{k,n}) \le \ldots \le f(z^{k,1}) = f(x^k)$ for all $k \ge 0$, by continuity of f we have that $\{f(x^k)\}$ converges, implying that

$$\lim_{k \to \infty} [f(z^{k,i+1}) - f(z^{k,i})] = 0, \quad i = 1,\dots,n. \tag{26}$$

Combining (25) and (26), we obtain

$$\lim_{\substack{k \to \infty \\ k \in K}} \alpha^{k,\hat{\imath}} |\nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}}| = 0. \tag{27}$$

Proceeding by contradiction, we assume that it does not hold that

$$\lim_{\substack{k\to\infty\\ k\in K}} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}} = 0. \tag{28}$$

Since $\{z^{k,\hat{\imath}}\}_K$ converges, then it is bounded and, by continuity of ∇f and the definition of the direction $d^{k,i}$, also $\left\{\nabla f(z^{k,\hat{\imath}})\right\}_K$ and $\{d^{k,\hat{\imath}}\}_K$ are bounded. So, if (28) does not hold, there exist further infinite subsequences, that we still denote by $\{z^{k,\hat{\imath}}\}_K$, $\left\{\nabla f(z^{k,\hat{\imath}})\right\}_K$ and $\{d^{k,\hat{\imath}}\}_K$ without loss of generality, such that

$$\lim_{\substack{k\to\infty\\ k\in K}} z^{k,\hat{\imath}} = \tilde{z} \in \mathbb{R}^n, \qquad \lim_{\substack{k\to\infty\\ k\in K}} d^{k,\hat{\imath}} = \tilde{d} \in \mathbb{R}^n \tag{29}$$

and

$$\lim_{\substack{k\to\infty\\ k\in K}} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}} = \nabla f(\tilde{z})^T \tilde{d} = -\eta \in \mathbb{R}, \tag{30}$$

with $\eta > 0$. From (27) and (30), we get

$$\lim_{\substack{k\to\infty\\ k\in K}} \alpha^{k,\hat{\imath}} = 0. \tag{31}$$

Since $A^{k,\hat{\imath}} \ge A_l > 0$ for all $k \ge 0$ and $\bar{\alpha}^{k,\hat{\imath}} \ge \xi/2 > 0$ for all sufficiently large $k \in K$, using the definition of $\Delta^{k,\hat{\imath}}$ we obtain

$$\Delta^{k,\hat{\imath}} \ge \min\{\xi/2, A_l\} > 0, \quad \forall \text{ sufficiently large } k \in K.$$

Consequently, by (31), an outer iteration $\bar{k} \in K$ exists such that

$$\alpha^{k,\hat{\imath}} < \Delta^{k,\hat{\imath}}, \quad \forall k \ge \bar{k},\ k \in K.$$

The above relation implies that (21) is satisfied with $c > 0$ for all $k \ge \bar{k}$, $k \in K$. Therefore,

$$f\left(z^{k,\hat{\imath}} + \frac{\alpha^{k,\hat{\imath}}}{\delta} d^{k,\hat{\imath}}\right) > f(z^{k,\hat{\imath}}) + \gamma \frac{\alpha^{k,\hat{\imath}}}{\delta} \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}}, \quad \forall k \ge \bar{k},\ k \in K. \tag{32}$$

By the mean value theorem, we can write

$$f\left(z^{k,\hat{\imath}} + \frac{\alpha^{k,\hat{\imath}}}{\delta} d^{k,\hat{\imath}}\right) = f(z^{k,\hat{\imath}}) + \frac{\alpha^{k,\hat{\imath}}}{\delta} \nabla f(\beta^{k,\hat{\imath}})^T d^{k,\hat{\imath}}, \tag{33}$$

where $\beta^{k,\hat{\imath}} = z^{k,\hat{\imath}} + \theta^{k,\hat{\imath}} \frac{\alpha^{k,\hat{\imath}}}{\delta} d^{k,\hat{\imath}}$ and $\theta^{k,\hat{\imath}} \in (0,1)$. Using (32) and (33), we obtain

$$\nabla f(\beta^{k,\hat{\imath}})^T d^{k,\hat{\imath}} > \gamma \nabla f(z^{k,\hat{\imath}})^T d^{k,\hat{\imath}}, \quad \forall k \geq \bar{k},\ k \in K. \tag{34}$$

Since $\theta^{k,\hat{\imath}} \in (0,1)$, and taking into account (29) and (31), it follows that $\{\beta^{k,\hat{\imath}}\}_K \to \tilde{z}$. So, passing to the limit in (34), we have that $\nabla f(\tilde{z})^T \tilde{d} \geq \gamma \nabla f(\tilde{z})^T \tilde{d}$. Using (30), we obtain $-\eta \geq -\gamma\eta$, contradicting the fact that $\eta > 0$ and $\gamma \in (0,1)$. Then, point (iii) of SC 1 holds. □

4.2 Objective functions with Lipschitz continuous gradient

In this subsection, we consider the case where ∇f is Lipschitz continuous over $\mathcal{F}$ with constant L. Namely, we assume that

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|, \quad \forall x, y \in \mathcal{F}.$$

First, let us recall a result due to Beck [1], which will be useful in the sequel.

Lemma 2 *Let us assume that ∇f is Lipschitz continuous over $\mathcal{F}$ with constant L. For any point $z \in \mathcal{F}$ and any pair of indices $i, j \in \{1,\dots,n\}$, define the function*

$$\phi_{i,j,z}(t) := f\big(z + t(e_i - e_j)\big), \quad t \in I_{i,j,z},$$

where $I_{i,j,z}$ is the interval that comprises the feasible stepsizes. Namely,

$$I_{i,j,z} := \{t \in \mathbb{R} \colon z + t(e_i - e_j) \in \mathcal{F}\}. \tag{35}$$

Then, every $\dot{\phi}_{i,j,z}$ is Lipschitz continuous over $I_{i,j,z}$ with constant $L_{i,j} \leq 2L$, that is,

$$|\dot{\phi}_{i,j,z}(t) - \dot{\phi}_{i,j,z}(s)| \leq L_{i,j}|t - s|, \quad \forall z \in \mathcal{F}, \quad \forall t, s \in I_{i,j,z}. \tag{36}$$

Proof See Sect. 3 in [1]. □

Now, let $\bar{L}_{i,j}$ be some positive overestimates of $L_{i,j}$, with $L_{i,j}$ being the Lipschitz constants defined in Lemma 2. Namely,

$$\bar{L}_{i,j} \geq L_{i,j}, \quad \text{such that} \quad \bar{L}_{i,j} > 0, \qquad i, j = 1,\dots,n. \tag{37}$$

We will show that these overestimates can be used to compute, in closed form, a stepsize satisfying SC 1. In particular, for a given fixed parameter $\gamma \in (0,1)$, at every inner iteration (k,i) we can set

$$\alpha^{k,i} = \min\left\{\bar{\alpha}^{k,i}, \frac{2(1-\gamma)}{\bar{L}_{p_i^k, j(k)}}\right\}. \tag{38}$$

As to be pointed out in the proof of the following proposition, this stepsize can be seen as a particular case of the Armijo stepsize defined in Proposition 2. Note also that, since $L_{i,j} \le 2L$, every positive overestimate of $2L$ can be used in (38).

Proposition 3 *Let us assume that ∇f is Lipschitz continuous over $\mathcal{F}$ with constant L and let $\gamma \in (0,1)$. Then, SC 1 is satisfied by computing, at every inner iteration (k,i), the stepsize $\alpha^{k,i}$ as in (38).*

Proof Let $i, j \in \{1,\dots,n\}$ be any pair of indices and consider inequality (36). Observing that $0 \in I_{i,j,z}$ for every feasible z, and using known results on functions with Lipschitz continuous gradient (see, e.g., [23]), we can write

$$\phi_{i,j,z}(t) \le \phi_{i,j,z}(0) + t\,\dot{\phi}_{i,j,z}(0) + \frac{L_{i,j}}{2}t^2, \quad \forall\, z \in \mathcal{F}, \quad \forall\, t \in I_{i,j,z}.$$

Since $\dot{\phi}_{i,j,z}(t) = \nabla_i f\big(z + t(e_i - e_j)\big) - \nabla_j f\big(z + t(e_i - e_j)\big)$, it follows that

$$f\big(z + t(e_i - e_j)\big) \le f(z) + t[\nabla_i f(z) - \nabla_j f(z)] + \frac{L_{i,j}}{2}t^2, \quad \forall\, z \in \mathcal{F}, \quad \forall\, t \in I_{i,j,z}. \tag{39}$$

To prove the assertion, it is sufficient to show that $\alpha^{k,i}$, defined as in (38), is a particular case of the Armijo stepsize defined in Proposition 2. To this extent, we can set $A^{k,i} = 2(1-\gamma)/\bar{L}_{p_i^k,j(k)}$ (all these quantities are positive and finite) and, by (38), we obtain $\alpha^{k,i} = \Delta^{k,i}$, with $\Delta^{k,i}$ defined as in Proposition 2. So, all we need is to prove that (21) holds with $c = 0$ for this choice of $\Delta^{k,i}$. Namely, we want to show that

$$f(z^{k,i} + \Delta^{k,i} d^{k,i}) \le f(z^{k,i}) + \gamma \Delta^{k,i} \nabla f(z^{k,i})^T d^{k,i}. \tag{40}$$

Since $\Delta^{k,i} \le \bar{\alpha}^{k,i}$, and taking into account the definition of $d^{k,i}$ given in (6), we have that

$$z^{k,i} + \Delta^{k,i} d^{k,i} = z^{k,i} + \Delta^{k,i} g^{k,i} (e_{p_i^k} - e_{j(k)}) \in \mathcal{F},$$

that is, $\Delta^{k,i} g^{k,i} \in I_{p_i^k, j(k), z^{k,i}}$ by definition (35). Therefore, using (39) with i and j replaced by p_i^k and $j(k)$, respectively, $z = z^{k,i}$ and $t = \Delta^{k,i} g^{k,i}$, we obtain

$$\begin{aligned} f(z^{k,i} + \Delta^{k,i} d^{k,i}) &\le f(z^{k,i}) - \Delta^{k,i}(g^{k,i})^2 + \frac{\bar{L}_{p_i^k,j(k)}}{2}(\Delta^{k,i})^2 (g^{k,i})^2 \\ &= f(z^{k,i}) + \Delta^{k,i} \nabla f(z^{k,i})^T d^{k,i} \left(1 - \frac{\bar{L}_{p_i^k,j(k)}}{2}\Delta^{k,i}\right), \end{aligned}$$

where the equality follows from (7). Since $0 \le \Delta^{k,i} \le 2(1-\gamma)/\bar{L}_{p_i^k,j(k)}$, we get (40). □

Remark 2 As appears from the proof of Proposition 3, the stepsize given in (38) satisfies SC 1 by using, for every pair of indices $i, j \in \{1, \dots, n\}$, a positive constant $\bar{L}_{i,j} \geq L_{i,j}$, where it is sufficient that $L_{i,j}$ satisfies (39). In particular, a case of interest is when we have a (possibly non-convex) separable objective function $f(x) = \sum_{i=1}^{n} f_i(x_i)$ where each $f_i : \mathbb{R} \to \mathbb{R}$ has a Lipschitz continuous derivative with constant L_i. In this case, Proposition 3 holds even with $\bar{L}_{i,j}$ replaced by $\bar{L}_i + \bar{L}_j$ in (38), where $\bar{L}_i$ and $\bar{L}_j$ are two positive overestimates of L_i and L_j, respectively. This follows from the fact that, in this case, (39) holds even with $L_{i,j}$ replaced by $L_i + L_j$, since, by Lipschitz continuity of $\dot{f}_1, \dots, \dot{f}_n$, we have

$$\begin{aligned} f\big(z + t(e_i - e_j)\big) &= f_i(z_i + t) + f_j(z_j - t) + \sum_{h \neq i,j} f_h(z_h) \\ &\leq f(z) + t[\dot{f}_i(z_i) - \dot{f}_j(z_j)] + \frac{L_i + L_j}{2} t^2 \\ &= f(z) + t[\nabla_i f(z) - \nabla_j f(z)] + \frac{L_i + L_j}{2} t^2. \end{aligned}$$

Now, let us analyze the case where the objective function is quadratic of the following form: $f(x) = \frac{1}{2} x^T Q x - q^T x$, with $Q \in \mathbb{R}^{n \times n}$ symmetric and $q \in \mathbb{R}^n$. Denoting by $Q_{i,j}$ the element of Q in position (i, j), we have (again from [1])

$$L_{i,j} = |Q_{i,i} + Q_{j,j} - 2Q_{i,j}|, \quad i, j \in \{1, \dots, n\}.$$

So, at every inner iteration (k, i), we can easily obtain (a positive overestimate of) $L_{p_i^k, j(k)}$ in order to compute the stepsize $\alpha^{k,i}$ as in (38).

Moreover, if $(d^{k,i})^T Q d^{k,i} > 0$ for a given direction $d^{k,i}$, we can write

$$0 < \frac{1}{Q_{p_i^k, p_i^k} + Q_{j(k), j(k)} - 2Q_{p_i^k, j(k)}} = -\frac{\nabla f(z^{k,i})^T d^{k,i}}{(d^{k,i})^T Q d^{k,i}} \in \operatorname*{Argmin}_{\alpha \in \mathbb{R}} f(z^{k,i} + \alpha d^{k,i}).$$

It follows that, when $(d^{k,i})^T Q d^{k,i} > 0$, we can set $\gamma = 1/2$, $\bar{L}_{p_i^k, j(k)} = L_{p_i^k, j(k)}$ and the stepsize given in (38) is the exact stepsize, i.e., it is the feasible minimizer of $f(z^{k,i} + \alpha d^{k,i})$ with respect to α.

Vice versa, if $(d^{k,i})^T Q d^{k,i} \leq 0$ for a given direction $d^{k,i}$, we can exploit the fact that the greatest objective decrease along $d^{k,i}$ is achieved by setting $\alpha^{k,i}$ as large as possible, since

$$f(z^{k,i} + \alpha d^{k,i}) = f(z^{k,i}) + \alpha \nabla f(z^{k,i})^T d^{k,i} + \frac{1}{2} \alpha^2 (d^{k,i})^T Q d^{k,i}, \quad \forall \alpha \in \mathbb{R}. \tag{41}$$

So, we can set $\alpha^{k,i} = \min\{\bar{\alpha}^{k,i}, A_u\}$, with $0 < A_u < \infty$ being a (large) fixed parameter. Using (41), it is easy to see that also this stepsize is a particular case of the Armijo stepsize defined in Proposition 2 (indeed, for every nonnegative value of $\Delta^{k,i}$, the Armijo condition (21) is satisfied by $c = 0$).

4.3 Strictly convex objective functions

Let us consider a strictly convex objective function. In this case, we can satisfy SC 1 by computing, at every inner iteration (k, i), the stepsize $\alpha^{k,i}$ by an exact line search, that is,

$$\alpha^{k,i} \in \operatorname{Argmin}\{f(z^{k,i} + \alpha d^{k,i}) \colon \alpha \in [0, \bar{\alpha}^{k,i}]\}. \tag{42}$$

To ensure that the above minimization is well defined at every inner iteration, we assume that $\mathcal{L}^0$ is compact.

Proposition 4 *Let us assume that f is strictly convex and $\mathcal{L}^0$ is compact. Then, SC 1 is satisfied by computing, at every inner iteration (k, i), the stepsize $\alpha^{k,i}$ as in (42).*

Proof Point (i) of SC 1 immediately follows from the definition of $\alpha^{k,i}$. Since $f(x^{k+1}) = f(z^{k,n+1}) \le f(z^{k,n}) \le \cdots \le f(z^{k,1}) = f(x^k)$ and each $z^{k,i}$ lies in the compact set $\mathcal{L}^0$, it follows that $\{f(x^k)\}$ converges. So, to prove point (ii) of SC 1 we have to show that $\lim\limits_{k\to\infty} \|z^{k,i+1} - z^{k,i}\| = 0$, $i = 1, \dots, n$. Arguing by contradiction, assume that this is not true. Then, there exist a real number $\rho > 0$, an index $\hat{\imath} \in \{1, \dots, n\}$ and an infinite subsequence $\{z^{k,\hat{\imath}}\}_{K \subseteq \{0,1,\dots\}}$ such that $\|z^{k,\hat{\imath}+1} - z^{k,\hat{\imath}}\| \ge \rho$ for all $k \in K$. Since every point $z^{k,i}$ lies in the compact set $\mathcal{L}^0$, there also exist a further infinite subsequence, that we still denote by $\{z^{k,\hat{\imath}}\}_K$ without loss of generality, and two distinct points $z', z'' \in \mathbb{R}^n$ such that

$$\lim_{\substack{k\to\infty\\k\in K}} z^{k,\hat{\imath}} = z' \quad \text{and} \quad \lim_{\substack{k\to\infty\\k\in K}} z^{k,\hat{\imath}+1} = z''. \tag{43}$$

As $\alpha^{k,i}$ is obtained by an exact line search, we can write

$$\begin{aligned} f(z^{k,\hat{\imath}+1}) &\le f\Big(z^{k,\hat{\imath}} + \frac{1}{2}\alpha^{k,\hat{\imath}} d^{k,\hat{\imath}}\Big) = f\Big(\frac{z^{k,\hat{\imath}} + z^{k,\hat{\imath}+1}}{2}\Big) \\ &\le \frac{1}{2} f(z^{k,\hat{\imath}}) + \frac{1}{2} f(z^{k,\hat{\imath}+1}) \le f(z^{k,\hat{\imath}}), \end{aligned} \tag{44}$$

where the second inequality follows from the convexity of f and the last inequality follows from the fact that $f(z^{k,\hat{\imath}+1}) \le f(z^{k,\hat{\imath}})$. Moreover, since $\{f(x^k)\}$ converges, a real number $\bar{f}$ exists such that

$$\lim_{k\to\infty} f(z^{k,i}) = \bar{f}, \quad i = 1, \dots, n. \tag{45}$$

By continuity of the objective function, we can write

$$\lim_{\substack{k\to\infty\\k\in K}} f(z^{k,\hat{\imath}}) = f(z') = \bar{f} \quad \text{and} \quad \lim_{\substack{k\to\infty\\k\in K}} f(z^{k,\hat{\imath}+1}) = f(z'') = \bar{f}.$$

So, passing to the limit in (44), we obtain $\bar{f} \leq f\Big(\dfrac{z'+z''}{2}\Big) \leq \bar{f}$, that is, $f\Big(\dfrac{z'+z''}{2}\Big) = \bar{f}$. Adding to the left-hand side of this equality the two quantities $\Big(\dfrac{1}{2}\bar{f} - \dfrac{1}{2}f(z')\Big)$ and $\Big(\dfrac{1}{2}\bar{f} - \dfrac{1}{2}f(z'')\Big)$, that are both equal to zero, we get

$$f\Big(\frac{z'+z''}{2}\Big) = \frac{1}{2}f(z') + \frac{1}{2}f(z''),$$

contradicting the fact that f is strictly convex and $z' \neq z''$. Then, point (ii) of SC 1 holds.

Finally, point (iii) of SC 1 can be proved by the same arguments used in the proof of Proposition 2 for the Armijo stepsize, just observing that the objective decrease achieved by the exact line search is greater than or equal to the one achieved by the Armijo line search. □

5 Convergence rate analysis

In this section, we show that the convergence rate of AC2CD is linear under standard additional assumptions.

The key to prove linear convergence rate of AC2CD is to show a relation between the points produced by AC2CD and the points produced by the classical coordinate descent method applied to an equivalent transformed problem. Then, linear convergence rate follows from the well known properties of the classical coordinate descent method proved by Luo and Tseng [17] and by Beck and Tetruashvili [2].

In particular, here we give two results. The first one is more general and states that, eventually, $\{f(x^k)\}$ converges linearly to the optimal value of problem (1), that is, $C \in [0, 1)$ exists such that $f(x^{k+1}) - f(x^*) \leq C[f(x^k) - f(x^*)]$ for all sufficiently large k, with x^* being the optimal solution of problem (1). The second result is for the case where there are no bounds on the variables and establishes a non-asymptotic linear convergence rate, that is, the above inequality holds for every $k \geq 0$.

Let us start by showing the general result. First, we need a specific rule to compute the index $j(k)$ in order to ensure that it remains constant from a certain outer iteration $\hat{k}$. In particular, we initialize AC2CD with $\tau \in (0, 1)$ (step 0 of Algorithm 1) and, from a certain $k \geq 1$, we adopt the following rule: $j(k) = j(k-1)$ if this choice satisfies (4), otherwise $j(k)$ is set equal to any index h such that $D_h(x^k) = D^k$, where D_h and D^k are the operators given in (3) and (5), respectively. Namely,

$$j(k) \begin{cases} = j(k-1), & \text{if this choice satisfies (4)}, \\ \in \operatorname*{Argmax}_{h=1,\ldots,n} D_h(x^k), & \text{otherwise}. \end{cases} \tag{46}$$

We can state the following intermediate lemma.

Lemma 3 *Let Assumption 1 hold and let $\tau \in (0,1)$. Let $\{x^k\}$ be a sequence of points produced by AC2CD, where $j(k)$ is computed as in (46) from a certain $k \geq 1$. Let us also assume that $\lim_{k\to\infty}\{x^k\} = x^* \in \mathbb{R}^n$.*

Then, there exist a variable index $\bar{j}$ and an outer iteration $\hat{k}$ such that $j(k) = \bar{j}$ for all $k \geq \hat{k}$. Moreover, $x^*_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$.

Proof Let $j^* \in \text{Argmax}_{h=1,\dots,n} D_h(x^*)$. From Assumption 1, we have $D_{j^*}(x^*) > 0$.

First, we prove that there exist a variable index $\bar{j}$ and an outer iteration $\hat{k}$ such that $j(k) = \bar{j}$ for all $k \geq \hat{k}$. Arguing by contradiction, assume that this is not true. Since the set of indices $\{1,\dots,n\}$ is finite, there exist two indices $j_1, j_2 \in \{1,\dots,n\}$ and two infinite subsequences $\{x^k\}_{K_1}$ and $\{x^k\}_{K_2}$ such that

$$\begin{cases} j(k-1) \neq j_1, \\ j(k) = j_1, \end{cases} \quad \forall k \in K_1, \qquad \text{and} \qquad \begin{cases} j(k-1) = j_1, \\ j(k) = j_2, \end{cases} \quad \forall k \in K_2.$$

Since $j(k)$ is computed as in (46) from a certain $k \geq 1$, it follows that

$$\begin{aligned} &D_{j_1}(x^k) = D^k \geq D_{j^*}(x^k), && \forall \text{ sufficiently large } k \in K_1, \\ &D_{j_1}(x^k) < \tau D^k = \tau D_{j_2}(x^k), && \forall \text{ sufficiently large } k \in K_2. \end{aligned}$$

By continuity of the operator D_h, we can write

$$\begin{aligned} D_{j_1}(x^*) &= \lim_{\substack{k\to\infty \\ k\in K_1}} D_{j_1}(x^k) \geq \lim_{\substack{k\to\infty \\ k\in K_1}} D_{j^*}(x^k) = D_{j^*}(x^*), \\ D_{j_1}(x^*) &= \lim_{\substack{k\to\infty \\ k\in K_2}} D_{j_1}(x^k) \leq \tau \lim_{\substack{k\to\infty \\ k\in K_2}} D_{j_2}(x^k) = \tau D_{j_2}(x^*). \end{aligned}$$

Combining these two inequalities, and recalling that $D_{j^*}(x^*) > 0$, we obtain

$$0 < D_{j^*}(x^*) \leq \tau D_{j_2}(x^*).$$

This is contradiction, since $D_{j^*}(x^*) \geq D_{j_2}(x^*)$ and $\tau \in (0,1)$. So, a variable index $\bar{j}$ and an outer iteration $\hat{k}$ exist such that $j(k) = \bar{j}$ for all $k \geq \hat{k}$.

To prove that $x^*_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$, assume by contradiction that $D_{\bar{j}}(x^*) = 0$. Since $j(k)$ is computed as in (46) from a certain $k \geq 1$ and $j(k) = \bar{j}$ for all $k \geq \hat{k}$, for all sufficiently large k we have $D_{\bar{j}}(x^k) \geq \tau D^k \geq \tau D_{j^*}(x^k)$. By continuity of the operator D_h, we obtain

$$0 = D_{\bar{j}}(x^*) = \lim_{k\to\infty} D_{\bar{j}}(x^k) \geq \tau \lim_{k\to\infty} D_{j^*}(x^k) = \tau D_{j^*}(x^*),$$

which leads to a contradiction, since $D_{j^*}(x^*) > 0$ and $\tau \in (0,1)$. Therefore, $x^*_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$. □

Now, we are ready to show that, eventually, $\{f(x^k)\}$ converges linearly to $f(x^*)$ under the following assumption.

Assumption 2 It holds that

- f is strictly convex twice continuously differentiable over $\mathbb{R}^n$;
- the optimal solution of problem (1), denoted by x^*, exists;
- $\nabla^2 f(x^*) \succ 0$.

Note that Assumption 2 implies that $\mathcal{L}^0$ is compact (see Lemma 9.1 in [29]). Therefore, if also Assumption 1 holds and SC 1 is satisfied, then $\{x^k\}$ converges to the optimal solution x^*. As a further consequence, under Assumption 2 we can compute the stepsize by an exact line search to satisfy SC 1 (see Proposition 4).

Theorem 2 *Let Assumption 1 and 2 hold, and let $\tau \in (0, 1)$. Let $\{x^k\}$ be a sequence of points produced by AC2CD, where $j(k)$ is computed as in (46) from a certain $k \geq 1$ and the stepsize is computed as indicated in Proposition 4.*

Then, a real number $C \in [0, 1)$ and an outer iteration $\bar{k}$ exist such that

$$f(x^{k+1}) - f(x^*) \leq C[f(x^k) - f(x^*)], \quad \forall\, k \geq \bar{k}.$$

Proof Without loss of generality, we assume that $n > 1$ (otherwise the feasible set is either empty or a singleton). Let $\bar{j}$ and $\hat{k}$ be the variable index and the outer iteration defined in Lemma 3, respectively. Namely, $j(k) = \bar{j}$ for all $k \geq \hat{k}$. To simplify the notation, without loss of generality we assume that

$$p_n^k = \bar{j}, \quad \forall\, k \geq \hat{k}, \qquad \text{and} \qquad \bar{j} = n. \tag{47}$$

Let us consider the following variable transformation:

$$x_i = \begin{cases} y_i, & i \in \{1, \dots, n\} \setminus \{\bar{j}\}, \\ b - \displaystyle\sum_{h \neq \bar{j}} y_h, & i = \bar{j}. \end{cases} \tag{48}$$

Equivalently, we can write

$$x = My + w, \tag{49}$$

where, recalling that $\bar{j} = n$,

$$M = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -1 & -1 & \dots & -1 \end{bmatrix} \in \mathbb{R}^{n \times (n-1)} \qquad \text{and} \qquad w = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix} \in \mathbb{R}^n. \tag{50}$$

Note that the columns of M are linearly independent and, for every point $x \in \mathbb{R}^n$ such that $\sum_{i=1}^n x_i = b$, the linear system $x = My + w$ has a unique solution given by $y_i = x_i$, $i = 1, \dots, n-1$.

Now, since $x_n^* \in (l_n, u_n)$ (by Lemma 3) and f is strictly convex, in problem (1) we can remove the bound constraints on the variable x_n and x^* is still the unique optimal solution. Consequently, by defining the function $\psi \colon \mathbb{R}^{n-1} \to \mathbb{R}$ as $\psi(y) := f\big(My + w\big)$, we can recast problem (1) as follows:

$$\begin{aligned} &\min\ \psi(y) \\ &l_i \le y_i \le u_i, \quad i = 1, \ldots, n-1. \end{aligned} \tag{51}$$

We observe that, for every feasible point y of problem (51), the corresponding x obtained by (49) satisfies both the equality constraint $\sum_{i=1}^n x_i = b$ and the bound constraints $l_i \le x_i \le u_i$, $i = 1, \ldots, n-1$ (the bound constraints on x_n are ignored for the reasons explained above).

An optimal solution y^* of problem (51) exists, is unique and is given by $y_i^* = x_i^*$, $i = 1, \ldots, n-1$. Indeed, this y^* clearly satisfies $x^* = My^* + w$ and $\psi(y^*) = f(x^*)$. So, if a feasible point $\tilde{y} \neq y^*$ of problem (51) existed such that $\psi(\tilde{y}) \le \psi(y^*)$, it would mean that $M\tilde{y} + w \neq My^* + w$ (since the columns of M are linearly independent) and $f(M\tilde{y} + w) \le f(My^* + w) = f(x^*)$. But this is not possible, since $M\tilde{y}$ would be feasible for problem (1) after removing the bound constraints on x_n, and we said above that x^* is still the unique optimal solution of problem (1) even if we remove the bound constraints on x_n.

Now, for any $k \ge \hat{k}$ and for any $i = 1, \ldots, n$, let us define $y^{k,i}$ as the unique vector [feasible for problem (51)] satisfying

$$z^{k,i} = My^{k,i} + w, \tag{52}$$

which is given by

$$y_h^{k,i} = z_h^{k,i}, \quad h = 1, \ldots, n-1. \tag{53}$$

For all $k \ge \hat{k}$, without loss of generality we can consider only the first $n-1$ inner iterations in AC2CD. Indeed, by (47) we are assuming that $p_n^k = \bar{j}$ for all $k \ge \hat{k}$, and then, no pair of coordinates is updated in the inner iteration (k, n) for all $k \ge \hat{k}$ (see Remark 1). Let us consider any inner iteration (k, i), with $k \ge \hat{k}$ and $i \in \{1, \ldots, n-1\}$. We want to show that

$$z^{k,i} + \alpha d^{k,i} = My \Bigg|_{\substack{y_{p_i^k} = y^{k,i}_{p_i^k} - \alpha \nabla_{p_i^k} \psi(y^{k,i}) \\ y_h = y_h^{k,i},\, h=1,\ldots,n-1,\, h \neq p_i^k}} + w, \qquad \forall \alpha \in \mathbb{R}. \tag{54}$$

In other words, (54) says that moving $z^{k,i}$ along $d^{k,i}$ with a certain stepsize α corresponds, in the y space, to moving the p_i^kth coordinate of $y^{k,i}$ along $-\nabla_{p_i^k} \psi(y^{k,i})$ with the same stepsize α, keeping all the other coordinates of $y^{k,i}$ fixed. To prove that (54) holds, we use again the fact that, for all $\alpha \in \mathbb{R}$, the linear system $z^{k,i} + \alpha d^{k,i} = My + w$

has a unique solution y given by $y_h = z^{k,i}_h + \alpha d^{k,i}_h$, $h = 1, \dots, n-1$. Then, (54) is equivalent to writing

$$z^{k,i}_{p^k_i} + \alpha d^{k,i}_{p^k_i} = y^{k,i}_{p^k_i} - \alpha \nabla_{p^k_i} \psi(y^{k,i}), \tag{55a}$$
$$z^{k,i}_h + \alpha d^{k,i}_h = y^{k,i}_h, \quad h = 1, \dots, n-1, \quad h \neq p^k_i. \tag{55b}$$

Since $d^{k,i}_h = 0$ for all $h \notin \{p^k_i, \bar{j}\}$, and we are assuming that $\bar{j} = n$, (55b) immediately follows from (53). To obtain (55a), we use (53) again to write $y^{k,i}_{p^k_i} = z^{k,i}_{p^k_i}$. Thus, we only need to show that $d^{k,i}_{p^k_i} = -\nabla_{p^k_i} \psi(y^{k,i})$. This follows from the relation $\nabla \psi(y) = M^T \nabla f(My + w)$, that, combined with (52) and the definition of M given in (50), yields to

$$\begin{aligned}\nabla_{p^k_i} \psi(y^{k,i}) &= \nabla_{p^k_i} f(My^{k,i} + w) - \nabla_{\bar{j}} f(My^{k,i} + w) \\ &= \nabla_{p^k_i} f(z^{k,i}) - \nabla_{\bar{j}} f(z^{k,i}) = -d^{k,i}_{p^k_i}.\end{aligned}$$

Therefore, (55) holds, implying that (54) holds too. Using $\alpha = \alpha^{k,i}$ in (55), for all $k \geq \hat{k}$ we can write

$$\begin{aligned} &z^{k,i+1}_{p^k_i} = y^{k,i}_{p^k_i} - \alpha^{k,i} \nabla_{p^k_i} \psi(y^{k,i}), && i = 1, \dots, n-1, \\ &z^{k,i+1}_h = y^{k,i}_h, \quad h = 1, \dots, n-1, \quad h \neq p^k_i, && i = 1, \dots, n-1. \end{aligned}$$

Using (53) (with i replaced by $i+1$) in the above relations, for all $k \geq \hat{k}$ we obtain

$$y^{k,i+1}_{p^k_i} = y^{k,i}_{p^k_i} - \alpha^{k,i} \nabla_{p^k_i} \psi(y^{k,i}), \qquad i = 1, \dots, n-1, \tag{56a}$$
$$y^{k,i+1}_h = y^{k,i}_h, \quad h = 1, \dots, n-1, \quad h \neq p^k_i, \qquad i = 1, \dots, n-1. \tag{56b}$$

We see that, for all $k \geq \hat{k}$, the vectors $y^{k,1}, \dots, y^{k,n}$ are the same that would be generated by a coordinate descent algorithm applied to problem (51). In particular, for all $k \geq \hat{k}$, every $y^{k+1,1}$ is obtained from $y^{k,1}$ by selecting one coordinate at a time by a Gauss-Seidel (or cyclic) rule and moving it with a certain stepsize.

Now we want to show that, for all sufficiently large k, each stepsize $\alpha^{k,i}$ is the same that would be obtained by performing an exact line search in the updating scheme (56), applied to problem (51). First observe that there exists $\tilde{k} \geq \hat{k}$ such that $z^{k,i}_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$, $i = 1, \dots, n$, for all $k \geq \tilde{k}$, as $x^*_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$ and $\{z^{k,i}\} \to x^*$, $i = 1, \dots, n$. Therefore, since an exact line search is used in AC2CD, for all $i = 1, \dots, n$ we can write

$$\alpha^{k,i} \in \operatorname*{Argmin}_{\alpha \in \mathbb{R}} \{f(z^{k,i} + \alpha d^{k,i}) : \; z^{k,i}_{p^k_i} + \alpha d^{k,i}_{p^k_i} \in [l_{p^k_i}, u_{p^k_i}]\}, \quad \forall k \geq \tilde{k}. \tag{57}$$

In other words, for all $k \geq \tilde{k}$, the constraints $z^{k,i}_{\bar{j}} + \alpha d^{k,i}_{\bar{j}} \in [l_{\bar{j}}, u_{\bar{j}}]$ with respect to α are not necessary for the computation of the stepsize in AC2CD. Combining (57) with (54), we get that, for all $k \geq \tilde{k}$, each $\alpha^{k,i}$ is the optimal solution of the following one-dimensional problem:

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}} \ \psi(y) \\ &y_{p_i^k} = y_{p_i^k}^{k,i} - \alpha \nabla_{p_i^k} \psi(y^{k,i}) \\ &y_h = y_h^{k,i}, \quad h = 1, \ldots, n-1, \quad h \neq p_i^k, \\ &l_{p_i^k} \leq y_{p_i^k}^{k,i} - \alpha \nabla_{p_i^k} \psi(y^{k,i}) \leq u_{p_i^k}. \end{aligned} \tag{58}$$

In particular, the last bound constraints in (58) follow from those in (57), using (55a). Therefore, for all $k \geq \tilde{k}$, each $\alpha^{k,i}$ is the stepsize that would be obtained by performing an exact line search in the coordinate descent scheme (56), applied to problem (51).

So, according to the results established by Luo and Tseng [17], eventually $\{\psi(y^{k,1})\}$ converges linearly to the optimal value of problem (51) if the following three conditions hold: (i) ψ is strictly convex twice continuously differentiable over $\mathbb{R}^{n-1}$, (ii) the optimal solution y^* of problem (51) exists, (iii) $\nabla^2 \psi(y^*) \succ 0$. The second point has already been proved before, while the other two points follow by combining Assumption 2 with the fact that the columns of M are linearly independent and $x^* = My^* + w$. We conclude that a real number $C \in [0, 1)$ and an outer iteration $\bar{k} \geq \tilde{k}$ exist such that

$$\psi(y^{k+1,1}) - \psi(y^*) \leq C[\psi(y^{k,1}) - \psi(y^*)], \quad \forall k \geq \bar{k}.$$

Since $\psi(y^*) = f(x^*)$, $\psi(y^{k,1}) = f(z^{k,1}) = f(x^k)$ and $\psi(y^{k+1,1}) = f(z^{k+1,1}) = f(x^{k+1})$, we get the result. □

Now, we focus on the case where there are no bounds on the variables, i.e., $l_i = -\infty$ and $u_i = \infty$ for all $i = 1, \ldots, n$. In this case, a non-asymptotic linear convergence rate can be achieved by using the stepsize defined in Proposition 3 (with $\gamma = 1/2$). To obtain this result, we need the following assumption.

Assumption 3 It holds that

- f is strongly convex over $\mathbb{R}^n$ with constant μ;
- ∇f is Lipschitz continuous over $\mathbb{R}^n$ with constant L;
- the optimal solution of problem (1), denoted by x^*, exists.

Also in this case, we need to maintain the index $j(k)$ fixed throughout the algorithm. In particular, now we require $j(k)$ to be equal to any index $\bar{j} \in \{1, \ldots, n\}$ for all $k \geq 0$ (note that any choice of $\bar{j}$ is acceptable, since there are no bounds on the variables).

Theorem 3 *Let us assume that there are no bounds on the variables in problem* (1), *i.e.,* $l_i = -\infty$ *and* $u_i = \infty$ *for all* $i = 1, \ldots, n$, *and let Assumption* 3 *hold. Given any index* $\bar{j} \in \{1, \ldots, n\}$, *let* $\{x^k\}$ *be a sequence of points produced by AC2CD, where* $j(k) = \bar{j}$ *for all* $k \geq 0$ *and the stepsize is computed as indicated in Proposition* 3, *with* $\gamma = 1/2$.

Then, a real number $C \in [0, 1)$ exists such that

$$f(x^{k+1}) - f(x^*) \leq C[f(x^k) - f(x^*)], \quad \forall k \geq 0. \tag{59}$$

In particular, considering the constants $L_{i,j}$ defined in Lemma 2 and the constants $\bar{L}_{i,j}$ used for the stepsize computation, we have

$$C = 1 - \frac{\mu}{2\bar{L}_{\bar{j}}^{max}\Big[1 + (n-1)(\sum_{i \neq \bar{j}} L_{i,\bar{j}})^2/(\bar{L}_{\bar{j}}^{min})^2\Big]}, \tag{60}$$

where $\bar{L}_{\bar{j}}^{min} := \min\limits_{i \neq \bar{j}} \bar{L}_{i,\bar{j}}$ and $\bar{L}_{\bar{j}}^{max} := \max\limits_{i \neq \bar{j}} \bar{L}_{i,\bar{j}}$.

Proof Following the same line of arguments as in the proof of Theorem 2, without loss of generality we assume that $n > 1$ and that (47) holds for all $k \geq 0$. Then, we consider the variable transformation $x = My + w$ given in (48), (49) and (50) to define the function $\psi \colon \mathbb{R}^{n-1} \to \mathbb{R}$ as $\psi(y) := f\big(My + w\big)$. We obtain that problem (1) is equivalent to the following problem (which is now unconstrained since there are no bounds on the variables):

$$\min_{y \in \mathbb{R}^{n-1}} \; \psi(y). \tag{61}$$

An optimal solution y^* of problem (61) exists, is unique and is given by $y_i^* = x_i^*$, $i = 1, \ldots, n-1$, satisfying $x^* = My^* + w$ and $\psi(y^*) = f(x^*)$. Moreover, for any $k \geq 0$ and for any $i = 1, \ldots, n$, let us define $y^{k,i}$ as in (52)–(53). We have that (56) holds for all $k \geq 0$. Namely, for all $k \geq 0$, every $y^{k+1,1}$ is obtained from $y^{k,1}$ by selecting one coordinate at a time by a Gauss-Seidel (or cyclic) rule and moving it with a certain stepsize.

Now, let us consider the stepsize $\alpha^{k,i}$. By hypothesis, it is computed as described in Proposition 3, with $\gamma = 1/2$, using the constants $L_{i,j}$ and $\bar{L}_{i,j}$ as they are defined in Lemma 2 and in (37), respectively. Since there are no bounds on the variables, for every $d^{k,i} \neq 0$ we have $\bar{\alpha}^{k,i} = \infty$. Then,

$$\alpha^{k,i} = \begin{cases} 1/\bar{L}_{p_i^k,\bar{j}} & \text{if } d^{k,i} \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

We want to show that $L_{1,\bar{j}}, \ldots, L_{n-1,\bar{j}}$ are the coordinatewise Lipschitz constants of $\nabla\psi$ over $\mathbb{R}^{n-1}$. Namely, for every $i = 1, \ldots, n-1$ and for all $y \in \mathbb{R}^{n-1}$, we want to show that

$$\big|\nabla_i \psi\big(y(\alpha)\big) - \nabla_i \psi(y)\big| \leq L_{i,\bar{j}}|\alpha|, \quad \forall \alpha \in \mathbb{R}, \tag{62}$$

where $y(\alpha) \in \mathbb{R}^{n-1}$ is defined as follows:

$$\begin{aligned} y(\alpha)_i &= y_i + \alpha, \\ y(\alpha)_h &= y_h, \quad h = 1, \ldots, n-1, \quad h \neq i. \end{aligned}$$

Recalling that $\bar{j} = n$ and the definition of M given in (50), first observe that $\nabla_i \psi(y) = \nabla f\big(My + w\big)^T (e_i - e_{\bar{j}})$ and $\nabla_i \psi\big(y(\alpha)\big) = \nabla f\big(My(\alpha) + w\big)^T (e_i - e_{\bar{j}}) = \nabla f\big(My + w + \alpha(e_i - e_{\bar{j}})\big)^T (e_i - e_{\bar{j}})$. Thus,

$$\begin{aligned}&\big|\nabla_i \psi\big(y(\alpha)\big) - \nabla_i \psi(y)\big| \\ &\quad = \big|\nabla f\big(My + w + \alpha(e_i - e_{\bar{j}})\big)^T (e_i - e_{\bar{j}}) - \nabla f(My + w)^T (e_i - e_{\bar{j}})\big|.\end{aligned}$$

Considering the functions $\phi_{i,j,z}$ defined in Lemma 2, we have that $\dot{\phi}_{i,j,z}(\alpha) = \nabla f\big(z + \alpha(e_i - e_j)\big)^T (e_i - e_j)$. Therefore, from Lemma 2 we get

$$\big|\nabla_i \psi\big(y(\alpha)\big) - \nabla_i \psi(y)\big| = |\dot{\phi}_{i,\bar{j},My+w}(\alpha) - \dot{\phi}_{i,\bar{j},My+w}(0)| \le L_{i,\bar{j}} |\alpha|$$

and then (62) holds.

So, for all $k \ge 0$ and all $i = 1, \dots, n-1$ such that $d^{k,i} \ne 0$, each stepsize $\alpha^{k,i}$ is the reciprocal of an overestimate of $L_{p_i^k,\bar{j}}$, where $L_{p_i^k,\bar{j}}$ is the p_i^kth coordinatewise Lipschitz constant of $\nabla \psi$. According to the results stated by Beck and Tetruashvili [2] (see Theorem 3.9 in [2]), $\{\psi(y^{k,1})\}$ has a non-asymptotic linear convergence rate if the following three conditions hold: (i) ψ is strongly convex over $\mathbb{R}^{n-1}$, (ii) $\nabla \psi$ is Lipschitz continuous over $\mathbb{R}^{n-1}$, (iii) the optimal solution y^* of problem (61) exists. The third point has already been proved before. The first point follows from Assumption 2 and the fact that the columns of M are linearly independent. The second point follows from the fact that $L_{1,\bar{j}}, \dots, L_{n-1,\bar{j}}$ are the coordinatewise Lipschitz constants of $\nabla \psi$ over $\mathbb{R}^{n-1}$. In particular, using known results (see Lemma 2 in [22]) and the fact that $\bar{j} = n$, we can write

$$\begin{aligned}\|\nabla \psi(y') - \nabla \psi(y'')\| &\le \sum_{i=1}^{n-1} L_{i,\bar{j}} \, \|y' - y''\| \\ &= \sum_{i \ne \bar{j}} L_{i,\bar{j}} \, \|y' - y''\|, \quad \forall \, y', y'' \in \mathbb{R}^{n-1}.\end{aligned} \tag{63}$$

Therefore, a real number $C \in [0, 1)$ exists such that

$$\psi(y^{k+1,1}) - \psi(y^*) \le C[\psi(y^{k,1}) - \psi(y^*)], \quad \forall \, k \ge 0. \tag{64}$$

Then, we get (59) by using the fact that $\psi(y^*) = f(x^*)$, $\psi(y^{k,1}) = f(z^{k,1}) = f(x^k)$ and $\psi(y^{k+1,1}) = f(z^{k+1,1}) = f(x^{k+1})$.

Now, we prove (60). First observe that, by Theorem 3.9 in [2], the constant C appearing in (64) is given by

$$C = 1 - \frac{\mu^{\psi}}{2\bar{L}_{\bar{j}}^{\max}\Big[1 + (n-1)(L^{\psi})^2/(\bar{L}_{\bar{j}}^{\min})^2\Big]},$$

where μ^{ψ} and L^{ψ} are the strong convexity constant of ψ over $\mathbb{R}^{n-1}$ and the Lipschitz constant of $\nabla\psi$ over $\mathbb{R}^{n-1}$, respectively, while $\bar{L}_{\bar{j}}^{\min}$ and $\bar{L}_{\bar{j}}^{\max}$ are defined as in the assertion of the theorem. Using (63), we can upper bound L^{ψ} with $\sum_{i\neq\bar{j}} L_{i,\bar{j}}$. Therefore, to obtain (60), we only have to show that

$$\mu^{\psi} = \mu, \tag{65}$$

i.e., we have to show that ψ is strongly convex over $\mathbb{R}^{n-1}$ with constant μ. Using the fact that f is strongly convex over $\mathbb{R}^n$ with constant μ, for any $y', y'' \in \mathbb{R}^{n-1}$ and for all $\theta \in [0,1]$ we can write

$$\begin{aligned}\psi(\theta y' + (1-\theta)y'') &= f\big(M(\theta y' + (1-\theta)y'') + w\big)\\ &= f\big(\theta(My' + w) + (1-\theta)(My'' + w)\big)\\ &\le \theta f(My' + w) + (1-\theta) f(My'' + w) +\\ &\quad -\frac{\mu}{2}\theta(1-\theta)\|My' + w - My'' - w\|^2\\ &= \theta\psi(y') + (1-\theta)\psi(y'') - \frac{\mu}{2}\theta(1-\theta)\|M(y'-y'')\|^2.\end{aligned}$$

Denoting by $\lambda_{\min}(M^TM)$ the smallest eigenvalue of M^TM, we also have $\|M(y'-y'')\|^2 = (y'-y'')^T M^T M(y'-y'') \ge \lambda_{\min}(M^TM)\|y'-y''\|^2$, and then,

$$\begin{aligned}\psi(\theta y' + (1-\theta)y'') &\le \theta\psi(y') + (1-\theta)\psi(y'')\\ &-\frac{\mu}{2}\lambda_{\min}(M^TM)\theta(1-\theta)\|y'-y''\|^2.\end{aligned} \tag{66}$$

Note that M^TM has all entries equal to 1, except for those on the diagonal that are equal to 2. Namely,

$$M^TM = I_{(n-1)\times(n-1)} + 1_{(n-1)\times(n-1)},$$

where $I_{(n-1)\times(n-1)}$ is the $(n-1)$ dimensional identity matrix and $1_{(n-1)\times(n-1)}$ is the $(n-1)\times(n-1)$ matrix made of all ones. It follows that $\lambda_{\min}(M^TM) = 1 + 0 = 1$, that, combined with (66), implies that ψ is strongly convex over $\mathbb{R}^{n-1}$ with constant μ. Then, (65) holds and the result is obtained. □

Let us conclude this section by discussing the relation between AC2CD and the classical coordinate descent method. As appears from the proofs of Theorem 2 and Theorem 3, this relation is crucial to obtain linear convergence rate of AC2CD and it also provides further insight into the proposed method.

Indeed, for every $k \ge 0$, a cycle of inner iterations $(k,1),\dots,(k,n)$ in AC2CD can be seen as a cycle of iterations of the classical coordinate descent method, with cyclic selection rule, applied to an equivalent transformed problem. In particular, for every $k \ge 0$ we can use the variable transformation given in (48), (49) and (50), with $\bar{j}$ replaced by $j(k)$, to define the function $\psi(y) := f\big(My + w\big)$ and recast (1) as an equivalent problem. Note that, in absence of any information on $j(k)$, the resulting equivalent transformed problem

- can change from k to $k+1$, as $j(k)$ can change;
- has bound constraints $l_i \le y_i \le u_i$, $i \in \{1,\dots,n\}\setminus\{j(k)\}$, plus the constraints $l_{j(k)} \le b - \sum_{i\ne j(k)} y_i \le u_{j(k)}$, where the latter follow from the constraints $l_{j(k)} \le x_{j(k)} \le u_{j(k)}$ in (1).

Then, in the proofs of Theorems 2 and 3, we obtain a linear convergence rate by exploiting the rule used to compute $j(k)$ that, together with other standard assumptions, guarantees that $j(k) = \bar{j}$ for all $k \ge \hat{k}$ and $x^*_{\bar{j}} \in (l_{\bar{j}}, u_{\bar{j}})$ (for problems with no bounds on the variables, we have $\hat{k} = 0$). In this way, for all sufficiently large k, the equivalent transformed problem remains the same and the constraints $l_{j(k)} \le b - \sum_{i\ne j(k)} y_i \le u_{j(k)}$ can be ignored. It also follows that quickly identifying the index $\bar{j}$ in problems with finite bounds on the variables may benefit the algorithm. Then, it may be useful combining AC2CD with some strategies to predict which variables are at the bounds in the final solution, such es, e.g., those proposed in [6,8,11,27].

Finally, the approach of transforming (1) into an equivalent simply constrained problem has some connections with the method proposed by Bertsekas in [3] for solving problems with linear inequality constraints. In particular, at every iteration, the algorithm proposed in [3] selects a subset of n indices that comprises all those corresponding to the binding constraints. Then, assuming linear independence of these constraints, a linear variable transformation is used to obtain a new problem with a first block of box constraints, plus a second block of general linear inequality constraints which are not binding at the current point and can be ignored for the computation of the search direction, obtained by a Newton strategy. So, the main difference between AC2CD and the method proposed in [3] is that, for every $k \ge 0$, we perform a cycle of inner iterations to update one coordinate at a time in the equivalent transformed problem.

6 Numerical results

In this section, we present the numerical results of AC2CD on some structured problems where the computation of the partial derivatives of the objective function is cheap with respect to the computation of the whole gradient. The codes were written in Matlab (version R2017b) and the experiments were run on an Intel(R) Core(TM) i7-7500U with 16 GB RAM memory.

First, we considered the Chebyshev center problem and the linear support vector machine (SVM) training problem, which both can be written as

$$\begin{aligned} &\min\ f(x) = \frac{1}{2} x^T Q^T Q x - q^T x, \\ &\sum_{i=1}^n x_i = 1 \\ &l_i \le x_i \le u_i, \quad i = 1,\dots,n, \end{aligned} \tag{67}$$

for some $Q \in \mathbb{R}^{m\times n}$, $q \in \mathbb{R}^n$, $l_i \in \mathbb{R} \cup \{-\infty\}$ and $u_i \in \mathbb{R} \cup \{\infty\}$, $i = 1,\dots,n$. On these convex quadratic problems, we compared AC2CD with the following block decomposition methods:

- Random Coordinate Descent (RCD) [21], which, at every iteration, randomly selects two distinct variables from a given probability distribution and updates them by minimizing a quadratic model of the objective function;
- Maximal Violating Pair (MVP) [9,26], which, at every iteration, selects the two variables that most violate the stationarity conditions (2) and moves them by using an appropriate stepsize that, in our case, is computed by an exact line search.

Note that RCD, like AC2CD, does not need to compute the whole gradient of the objective function for choosing the working set. On the other hand, MVP exploits a Gauss-Southwell (or greedy) strategy which requires to calculate the whole vector $\nabla f(x^k)$ at each iteration k.

Then, we focused on problems with strongly convex objective function and no bounds on the variables, for which non-asymptotic linear convergence rate of AC2CD has been proved in Sect. 5. In this case, we compared our algorithm with two versions of RCD that were proposed in [20] for problems of the following form:

$$\begin{aligned} & \min\ f(x) = \sum_{i=1}^n f_i(x_i), \\ & \sum_{i=1}^n x_i = 0, \end{aligned} \tag{68}$$

where each $f_i\colon \mathbb{R} \to \mathbb{R}$ is convex and has a Lipschitz continuous derivative with constant $L_i > 0$. In our experiments, we considered the same test problems used in [20,32], which are of the form of (68), with strongly convex f_i, $i = 1,\dots,n$.

Lastly, we considered the following non-convex problems:

$$\begin{aligned} & \min\ f(x) = \frac{1}{2} x^T Q^T D Q x - q^T x, \\ & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0, \quad i = 1,\dots,n, \end{aligned} \tag{69}$$

for some $Q \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times m}$ and $q \in \mathbb{R}^n$, choosing D as a diagonal matrix so that $Q^T DQ$ is indefinite. On these problems, AC2CD is still compared with RCD (whose analysis for the non-convex case can be found in [25]) and MVP with exact line search (whose analysis for non-convex problems over the unit simplex can be found in [4], just observing that, for this class of problems, MVP coincides with the so called pairwise Frank-Wolfe method).

Finally, to have a fair comparison between AC2CD and RCD, in the following results we consider an outer iteration of RCD as made of n inner iterations, each of them involving a minimization step with respect to a pair of coordinates.

6.1 Implementation issues

In this subsection, we describe some implementation details concerning

- the computation of $\{p_1^k, \dots, p_n^k\}$ at step 3 of Algorithm 1,
- the computation of the derivatives of the objective function in each algorithm,
- the termination criterion used in each algorithm.

As regards the first point, a permutation $\{p_1^k, \dots, p_n^k\}$ of $\{1, \dots, n\}$ was randomly computed at the beginning of every outer iteration k of AC2CD (it is known that, in coordinate descent methods with cyclic selection rules, periodically shuffling the ordering of the variables may speed up convergence in practice [7,31]).

For what concerns the derivatives computation, this operation is straightforward for problem (68), given the separable structure of the objective function. More attention is needed for problems (67) and (69), since, due to the possibly excessive dimension of Q, in these problems the Hessian matrices $Q^T Q$ and $Q^T DQ$ were not stored. For what concerns problems (67), using some ideas from [10,21] we introduced the vector $r(x){:=}Qx$, updating it during the iterations of each considered algorithm, so that

$$\nabla_i f(x) = {Q_i}^T r(x) - q_i, \quad i = 1, \dots, n, \tag{70}$$

where Q_i is the ith column of Q. We see that computing any partial derivative has a cost $\mathcal{O}(m)$ and we have an extra cost $\mathcal{O}(2m)$ to update the vector $r(x)$ (since the considered algorithms can move two variables at a time). These costs can also reduce when Q is sparse. Similarly, for what concerns problems (69), we introduced the vector $r(x){:=}DQx$ and (70) still holds (in this case we have a cost $\mathcal{O}(3m)$ to update $r(x)$, due to the presence of the diagonal matrix D).

It is worth observing that, in AC2CD and RCD, both variables included in the working set at the beginning of an inner iteration may be at the lower or at the upper bound. In this case, the minimization step is not performed in practice (since the two variables cannot be moved), and then, the partial derivatives of the objective function do not need to be computed. So, by inserting a simple check in the scheme of AC2CD and RCD, we avoided to compute the partial derivatives of the objective function when not necessary.

For what concerns the termination criterion used in AC2CD, let us first rewrite the stationarity conditions (2) in an equivalent form:

$$\min_{i\,:\,x_i<u_i} \{\nabla_i f(x)\} - \max_{i\,:\,x_i>l_i} \{\nabla_i f(x)\} \geq 0.$$

Exploiting some ideas from [10], the termination criterion used in AC2CD tries to approximately satisfy the above relation without the need of computing the whole gradient of the objective function, in order to preserve efficiency. To this extent, at the beginning of every outer iteration k we first set $G^k_{\min} = \infty$ and $G^k_{\max} = -\infty$. Then, at every inner iteration (k, i) we update $G^k_{\min}$ and $G^k_{\max}$ as follows: for each variable $z_h^{k,i}$ included in the working set, if $z_h^{k,i} < u_h$ then we update $G^k_{\min} \leftarrow \min\{G^k_{\min}, \nabla_h f(z^{k,i})\}$, and if $z_h^{k,i} > l_h$ then we update $G^k_{\max} \leftarrow$

$\max\{G^k_{\max}, \nabla_h f(z^{k,i})\}$. So, AC2CD was stopped at the end of the first outer iteration k satisfying

$$G^k_{\min} - G^k_{\max} \ge -\epsilon, \tag{71}$$

where ϵ is a scalar that was set to 10^{-1}. Since $\nabla_{p^k_i} f(z^{k,i})$ and $\nabla_{j(k)} f(z^{k,i})$ are not computed when both $z^{k,i}_{p^k_i}$ and $z^{k,i}_{j(k)}$ are at the lower or at the upper bound, for the reasons explained above, we also added a final check after that (71) is satisfied, evaluating all those components of $\nabla f(x^k)$ that were skipped, if any.

For the convex problems considered in our experiments, we used the final objective value f^{AC2CD} returned by AC2CD as benchmark for termination of RCD and MVP. Namely, RCD and MVP were stopped when they produced a point x^k satisfying

$$\frac{f(x^k) - f^{\mathrm{AC2CD}}}{1 + |f^{\mathrm{AC2CD}}|} \le \nu, \tag{72}$$

where ν is a scalar that was set to 10^{-6}. Clearly, a termination criterion based on (72) cannot be used when the objective function is non-convex, since the algorithms may converge to different stationary points. So, for the non-convex problems (69), the same termination criterion used in AC2CD was also used in RCD (recall that we consider an outer iteration of RCD as made of n inner iterations), while MVP, that computes $\nabla f(x^k)$ at each iteration k, was stopped when it produced a point x^k such that $\min_{i=1,\dots,n}\{\nabla_i f(x^k)\} - \max_{i\,:\,x_i>0}\{\nabla_i f(x^k)\} \ge -10^{-1}$.

6.2 Chebyshev center

Given a finite set of vectors $v^1, \dots, v^n \in \mathbb{R}^m$, the Chebyshev center problem consists in finding the smallest ball that encloses all the given points. It arises in many fields, such as mechanical engineering, biology, environmental science and computer graphics (see [33] and the references therein for more details). The Chebyshev center problem can be formulated as the following convex standard quadratic problem:

$$\begin{aligned} &\min\ f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}(v^i)^T v^j x_i x_j - \sum_{i=1}^{n}\|v^i\|^2 x_i \\ &\sum_{i=1}^{n} x_i = 1 \\ &x_i \ge 0, \quad i = 1, \dots, n. \end{aligned}$$

Nine synthetic data sets were created by randomly generating each component of the samples $v^1, \dots, v^n$ from a standard normal distribution, using $n = 40{,}000, 60{,}000, 80{,}000$ and $m = 0.01n, 0.05n, 0.1n$ for every fixed n. The nine data sets are listed below:

(i) $n = 40{,}000$, $m = 400$;

(ii) $n = 40{,}000$, $m = 2000$;
(iii) $n = 40{,}000$, $m = 4000$;
(iv) $n = 60{,}000$, $m = 600$;
(v) $n = 60{,}000$, $m = 3000$;
(vi) $n = 60{,}000$, $m = 6000$;
(vii) $n = 80{,}000$, $m = 800$;
(viii) $n = 80{,}000$, $m = 4000$;
(ix) $n = 80{,}000$, $m = 8000$.

For each data set, a randomly chosen vertex of the unit simplex was used as starting point. In AC2CD, for all $k \geq 1$ the index $j(k)$ was computed as in (46), with $\tau = 0.9$ (for $k = 0$ we set $j(k) \in \text{Argmax}_{h=1,\dots,n} D_h(x^k)$), and the stepsize was computed as described in the last part of Sect. 4.2 for quadratic objective functions, with $\gamma = 1/2$ and $A_u = 10^{12}$. In RCD, an $\mathcal{O}(1)$ procedure was used at each inner iteration to randomly choose, from a uniform distribution, the pair of distinct variables to be updated. In particular, this procedure first randomly generates a real number r from a uniform distribution on $(0, \frac{n(n-1)}{2})$, and then sets $i = 1 + \left\lfloor (\sqrt{1 + 8\lfloor r \rfloor} + 1)/2 \right\rfloor$ and $j = 1 + \left\lfloor \lfloor r \rfloor - (i-2)(i-1)/2 \right\rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operation.

In Table 1, we report the results for each algorithm in terms of final objective value, number of (outer) iterations and CPU time in seconds. To analyze how fast the objective function decreases in the three considered algorithms, in Fig. 1 we plot the normalized optimization error E^k versus the CPU time. Coherently with the termination criterion used in the experiments, E^k is computed as the left-hand side of (72).

We see that, on all the considered data sets, AC2CD outperforms RCD both in CPU time, by a factor between 2 and 14, and in the number of outer iterations, by a factor between 60 and 247. Looking at Fig. 1 more in detail, we observe that AC2CD and RCD are comparable in the first iterations, but AC2CD is able to compute a solution with higher precision in a smaller amount of time. In comparison with MVP, we observe that AC2CD is slightly slower on the data sets with $m = 0.01n$, i.e., data sets (i), (iv) and (vii) . On all the other data sets, on average AC2CD is more than 2 times faster than MVP in CPU time.

6.3 Linear SVM

Linear Support Vector Machine [5] is a popular technique for data classification, which aims at separating a given set of samples by a hyperplane. Formally, let $v^1, \dots, v^n \in \mathbb{R}^m$ be a finite set of vectors and $a^1, \dots, a^n \in \{-1, +1\}$ be the corresponding labels. To train a linear SVM, we can solve the following convex quadratic problem:

$$\begin{aligned} &\min\ f(x) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a^i a^j (v^i)^T v^j x_i x_j - \sum_{i=1}^{n} x_i \\ &\sum_{i=1}^{n} a^i x_i = 0 \\ &0 \leq x_i \leq C, \quad i = 1, \dots, n, \end{aligned}$$

Table 1 Results of AC2CD, RCD and MVP on Chebyshev center problems

Data set	AC2CD			RCD			MVP		
	Obj	Outer iter	Time (s)	Obj	Outer iter	Time (s)	Obj	Iter	Time (s)
(i)	– 492.77	27	1.43	– 492.77	4520	15.26	– 492.77	183	0.98
(ii)	– 2191.46	26	2.65	– 2191.46	2103	11.95	– 2191.46	218	4.95
(iii)	– 4254.52	25	8.02	– 4254.52	1501	24.78	– 4254.52	281	12.62
(iv)	– 716.28	31	2.69	– 716.28	6160	31.31	– 716.28	168	1.92
(v)	– 3228.41	26	7.19	– 3228.40	2225	27.45	– 3228.40	308	15.48
(vi)	– 6314.02	24	13.59	– 6314.01	1602	40.61	– 6314.01	365	36.42
(vii)	– 933.28	30	3.86	– 933.28	7391	52.48	– 933.28	202	3.73
(viii)	– 4267.89	27	16.83	– 4267.89	2563	58.99	– 4267.89	368	32.50
(ix)	– 8370.25	25	21.00	– 8370.24	1743	58.22	– 8370.24	421	75.95

For each algorithm, the first column indicates the final objective value, the second column indicates the number of (outer) iterations and the third column indicates the CPU time in seconds

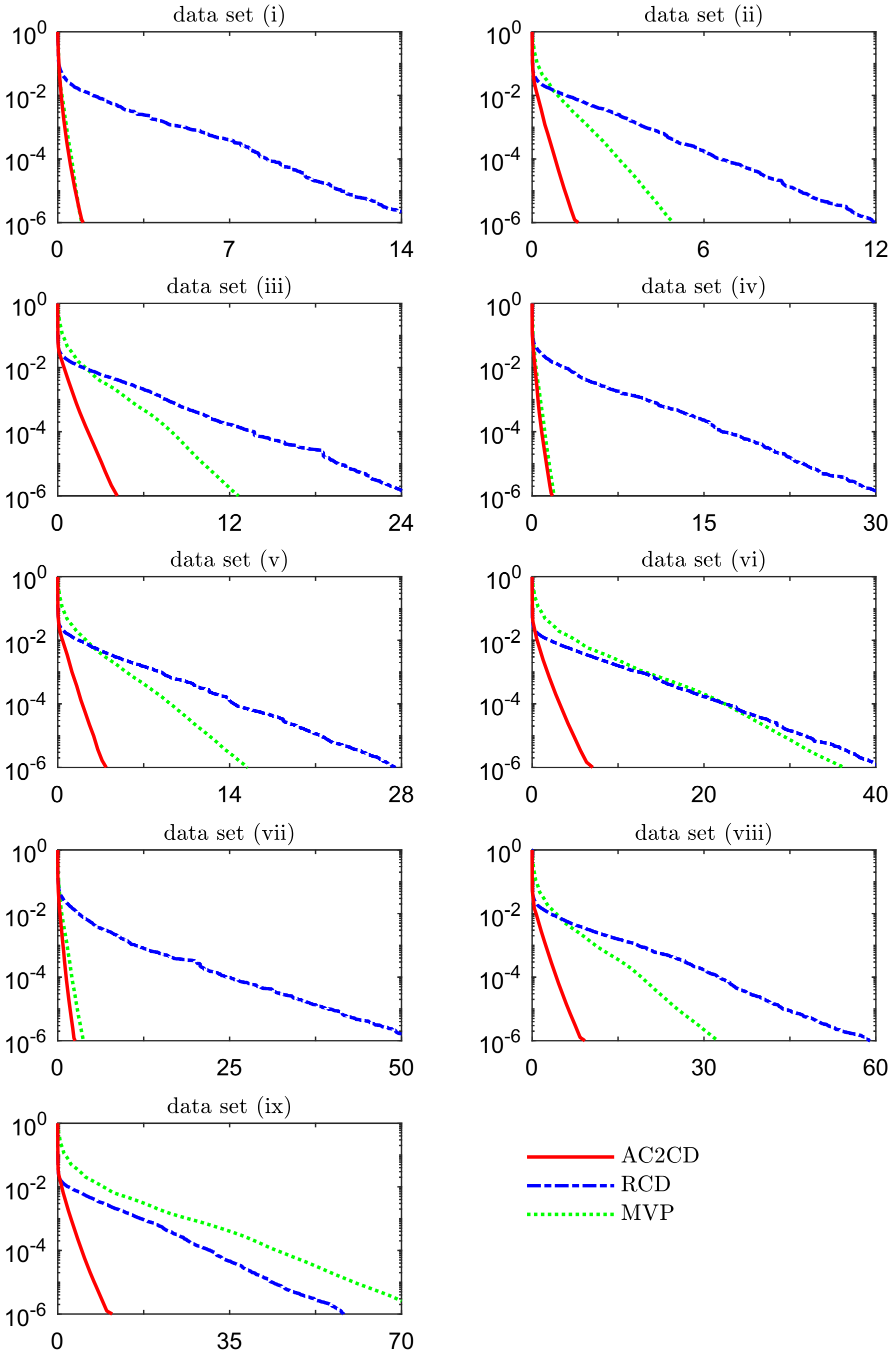

Fig. 1 Normalized optimization error (y axis) versus CPU time in seconds (x axis) for AC2CD, RCD and MVP on Chebyshev center problems. The y axis is in logarithmic scale and, for each algorithm, the normalized optimization error is computed as the left-hand side of (72)

where C is a positive parameter, set to 1 in our experiments. As mentioned in Sect. 2, the above problem can be easily rewritten as in (1).

Eight data sets were downloaded from the LIBSVM [6] webpage https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets. They are listed below:

(i) gisette ($n = 6000$, $m = 5000$);
(ii) rcv1 ($n = 20{,}242$, $m = 47{,}236$);
(iii) a9a ($n = 32{,}561$, $m = 123$);
(iv) w8a ($n = 49{,}749$, $m = 300$);
(v) ijcnn1 ($n = 49{,}990$, $m = 22$);
(vi) real sim ($n = 72{,}309$, $m = 20{,}958$);
(vii) webspam ($n = 350{,}000$, $m = 254$);
(viii) covtype ($n = 581{,}012$, $m = 54$).

For each data set, we used as starting point a vector made of all zeros except for two randomly chosen variables that were set strictly between the lower and the upper bound. In AC2CD, the index $j(k)$ and the stepsize were computed as described before for the Chebyshev center problem. In RCD, the working set was randomly chosen at each inner iteration by the same $\mathcal{O}(1)$ procedure used for the Chebyshev center problem.

The results for each algorithm are reported in Table 2 in terms of final objective value, number of (outer) iterations and CPU time in minutes. On the first six data sets, which have less than 10^5 samples, we see that MVP has the lowest CPU time, but AC2CD still outperforms RCD. In particular, on these problems AC2CD is on average faster than RCD by a factor of almost 5 in CPU time and by a factor larger than 11 in the number of outer iterations. On the two largest data sets, having more than 10^5 samples, AC2CD achieves the best performances. In particular, considering the CPU time, on data set (vii) AC2CD is more than 42 times faster than RCD and more than 4 times faster than MVP, while on data set (viii) AC2CD is about 15 times faster than RCD and about 13 times faster than MVP.

6.4 Problems with no bounds on the variables

The last convex test problems in our experiments are of the form of (68). We considered the same class of objective functions used in [20,32], that is,

$$f_i(x) = \frac{1}{2}a_i(x_i - c_i)^2 + \log\bigl(1 + \exp(b_i(x_i - d_i))\bigr), \quad i = 1, \dots, n,$$

where $a_i > 0, i = 1, \dots, n$, and $b_i, c_i, d_i \in \mathbb{R}, i = 1, \dots, n$. It is possible to show that each f_i is strongly convex with constant a_i and has a Lipschitz continuous derivative with constant $L_i = a_i + (1/4)b_i^2$ (see [20,32]).

Six artificial problems were created. The first three have the following dimension: (i) $n = 5000$, (ii) $n = 10{,}000$, (iii) $n = 20000$, with a_i randomly generated from a uniform distribution on $(0, 15)$ and b_i, c_i, d_i randomly generated from a uniform distribution on $(-15, 15)$. The last three problems have the same dimension as the previous ones, i.e., (iv) $n = 5000$, (v) $n = 10{,}000$, (vi) $n = 20{,}000$, but with a_i

Table 2 Results of AC2CD, RCD and MVP on linear SVM training problems

Dataset	AC2CD			RCD			MVP		
	Obj	Outer iter	Time (min)	Obj	Outer iter	Time (min)	Obj	Iter	Time (min)
(i)	– 0.67	35	4.67	– 0.67	97	9.54	– 0.67	3199	1.60
(ii)	– 1745.37	29	25.79	– 1745.36	101	65.45	– 1745.36	6642	0.87
(iii)	– 11,432.18	392	3.63	– 11,432.17	4961	27.19	– 11,432.17	76,973	1.44
(iv)	– 1486.57	491	13.26	– 1486.57	7398	31.99	– 1486.57	40,957	0.92
(v)	– 8589.43	101	0.45	– 8589.42	2623	4.35	– 8589.42	17,013	0.36
(vi)	– 5344.92	39	86.01	– 5344.91	284	436.74	– 5344.91	21,505	4.63
(vii)	– 69,448.61	60	9.79	– 69,448.55	2298	414.86	– 69,448.55	62,436	41.59
(viii)	– 337,942.88	200	25.23	– 337,942.55	3029	378.62	– 337,942.55	936,104	328.81

For each algorithm, the first column indicates the final objective value, the second column indicates the number of (outer) iterations and the third column indicates the CPU time in minutes randomly generated from a uniform distribution on $(0, 2)$, b_i randomly generated from a uniform distribution on $(-2, 2)$ and c_i, d_i randomly generated from a uniform distribution on $(-10, 10)$.

In all of these problems, the zero vector was used as starting point. In AC2CD, the index $j(k)$ was maintained fixed for all $k \geq 0$ and was chosen as an element of the set $\text{Argmax}_{i=1,\dots,n} 1/L_i$, while the stepsize was computed as described in Proposition 3, with $\gamma = 1/2$ and $\bar{L}_{i,j}$ replaced by $L_i + L_j$, $i, j = 1, \dots, n$ (see Remark 2).

Our algorithm was compared with two versions of RCD proposed in [20], using blocks made of two variables and different probability distributions (studied in [20]), which are now described. Denoting by p_{ij} the probability to select a pair of distinct variable indices (i, j) at any inner iteration of RCD, we first used a uniform distribution, i.e., all p_{ij} have the same value, and then we used probabilities that depend on the Lipschitz constants, i.e., each p_{ij} is equal to $(L_i^{-1} + L_j^{-1}) / \sum_{\substack{h,t=1,\dots,n \\ h \neq t}} (L_h^{-1} + L_t^{-1})$. We name the two resulting algorithms RCD_{unif} and RCD_{Lips}, respectively. More in detail, to choose the working set at any inner iteration, in RCD_{unif} we used the same $\mathcal{O}(1)$ procedure described before for the Chebyshev center problem, while in RCD_{Lips} we used the random generator proposed by Nesterov in [22], adjusted for our purposes. It requires to randomly generate one real number from a uniform distribution on $(0, 1)$ and perform $\mathcal{O}(\ln(\frac{n(n-1)}{2}))$ operations on some vectors (whose preliminary definition has a cost that has not been included in the final statistics).

In Table 3, we report the results for each algorithm in terms of final objective value, number of outer iterations and CPU time in seconds. We see that, on the first three problems, RCD_{unif} achieves the best performances: in terms of CPU time it is faster than AC2CD by a factor of about 1.5, but AC2CD is almost 13 times faster than RCD_{Lips} on average. On the last three problems, in terms of CPU time AC2CD outperforms both RCD_{unif} and RCD_{Lips} by an average factor of about 2 and 15, respectively. Moreover, we observe that the number of outer iterations of AC2CD and RCD_{Lips} is similar on all the considered problems, but the amount of time needed to converge is remarkably different, with AC2CD being much faster. This is due to the procedure used in RCD_{Lips} to randomly generate, at each inner iteration, a pair of variable indices from a Lipschitz-dependent probability distribution.

6.5 Non-convex problems

To test how AC2CD works when the objective function is non-convex, we finally considered problems of the form of (69). Each problem was created by the following procedure: first the elements of Q and those of q were randomly generated from a standard normal distribution and from a uniform distribution on $(0, 1)$, respectively; then the diagonal elements of D were set to 1, except for a prefixed number of them that were randomly chosen and set to negative values randomly generated from a uniform distribution on $(-1, 0)$.

More in detail, we generated three problems by fixing $n = m = 7000$ and considering a number of negative diagonal elements of D equal to $0.35m$, $0.5m$ and $0.65m$, respectively. The three problems are summarized below:

Table 3 Results of AC2CD, RCD$_{\text{unif}}$ and RCD$_{\text{Lips}}$ on singly linearly constrained problems with no bounds on the variables

Problem	AC2CD			RCD$_{\text{unif}}$			RCD$_{\text{Lips}}$		
	Obj	Outer iter	Time (s)	Obj	Outer iter	Time (s)	Obj	Outer iter	Time (s)
(i)	166,966.56	40,614	87.42	166,966.73	29,625	58.64	166,966.73	40,115	1001.08
(ii)	336,590.16	7738	33.61	336,590.50	5969	23.84	336,590.50	7251	414.50
(iii)	671,944.75	46,216	387.71	671,945.42	34,181	275.61	671,945.42	43,710	5711.38
(iv)	14,510.90	96	0.19	14,510.91	180	0.37	14,510.92	101	2.57
(v)	30,792.48	484	1.94	30,792.51	1011	4.11	30,792.51	481	27.76
(vi)	58,864.11	882	6.69	58,864.16	1507	12.31	58,864.16	888	117.41

For each algorithm, the first column indicates the final objective value, the second column indicates the number of outer iterations and the third column indicates the CPU time in seconds

Table 4 Results of AC2CD, RCD and MVP on non-convex problems

Problem	AC2CD			RCD			MVP		
	Obj	Outer iter	Time (s)	Obj	Outer iter	Time (s)	Obj	Iter	Time (s)
(i) – sp 1	– 3.10	925	110.90	– 3.07	3473	135.45	– 3.12	51,768	715.26
(i) – sp 2	– 3.11	493	57.91	– 3.06	4670	172.26	– 3.00	9496	131.18
(i) – sp 3	– 3.09	582	67.56	– 3.14	6585	231.34	– 3.05	24,364	336.40
(i) – sp 4	– 2.99	278	33.01	– 3.10	2896	108.96	– 3.06	30,712	424.11
(i) – sp 5	– 3.07	507	59.29	– 3.13	3139	113.96	– 3.06	31,679	437.19
(i) – sp 6	– 3.03	411	48.16	– 3.13	3563	128.45	– 3.06	21,713	300.07
(i) – sp 7	– 2.97	472	55.40	– 3.14	4325	153.75	– 3.22	47,542	657.03
(i) – sp 8	– 3.14	650	75.42	– 3.04	1776	67.81	– 3.22	32,163	444.68
(i) – sp 9	– 3.05	444	51.85	– 3.14	2606	94.93	– 3.10	30,057	415.61
(i) – sp 10	– 3.09	474	55.45	– 3.09	4199	149.46	– 3.10	19,097	263.63
(i) – avg	– 3.06	523.60	61.49	– 3.10	3723.20	135.63	– 3.10	29,859.10	412.52
(ii) – sp 1	– 7.91	996	94.34	– 8.19	11,696	126.07	– 7.46	2597	35.35
(ii) – sp 2	– 7.80	738	69.55	– 7.61	9323	100.88	– 7.69	4206	57.38
(ii) – sp 3	– 7.37	202	19.27	– 7.44	3533	40.79	– 7.87	2698	36.58
(ii) – sp 4	– 7.69	292	27.83	– 7.56	16,032	168.96	– 7.41	5122	69.61
(ii) – sp 5	– 7.71	445	42.15	– 7.69	12,016	126.37	– 7.53	8379	114.22
(ii) – sp 6	– 7.53	512	48.44	– 7.20	5140	61.16	– 7.71	3804	51.84

Table 4 continued

Problem	AC2CD			RCD			MVP		
	Obj	Outer iter	Time (s)	Obj	Outer iter	Time (s)	Obj	Iter	Time (s)
(ii) – sp 7	– 7.87	390	36.97	– 8.07	23,649	252.05	– 7.71	3015	40.83
(ii) – sp 8	– 7.63	284	27.11	– 7.46	6617	71.65	– 7.56	11,563	157.08
(ii) – sp 9	– 8.05	389	36.78	– 7.65	16,883	187.66	– 7.53	5112	69.51
(ii) – sp 10	– 7.58	764	72.03	– 7.80	5020	59.14	– 7.61	5694	77.43
(ii) – avg	– 7.72	501.20	47.45	– 7.67	10,990.90	119.47	– 7.61	5219.00	70.98
(iii) – sp 1	– 56.61	32	2.54	– 56.77	48,600	65.35	– 44.98	44	0.59
(iii) – sp 2	– 51.07	37	3.02	– 50.38	12,031	26.98	– 54.02	19	0.25
(iii) – sp 3	– 72.15	51	4.30	– 48.97	58,935	125.35	– 55.15	59	0.78
(iii) – sp 4	– 53.08	32	2.57	– 54.34	19,085	41.03	– 48.57	27	0.37
(iii) – sp 5	– 53.35	138	11.63	– 56.94	19,784	35.11	– 44.35	29	0.39
(iii) – sp 6	– 41.55	120	10.15	– 47.77	48,053	58.37	– 58.46	100	1.33
(iii) – sp 7	– 70.33	80	6.92	– 71.63	17,600	68.96	– 71.63	13	0.18
(iii) – sp 8	– 54.30	74	6.30	– 52.52	26,160	27.18	– 54.20	20	0.27
(iii) – sp 9	– 67.51	16	1.23	– 71.77	15,153	36.87	– 51.24	73	0.98
(iii) – sp 10	– 61.52	60	4.79	– 56.74	8267	9.33	– 47.67	57	0.75
(iii) – avg	– 58.15	64.00	5.34	– 56.78	27,366.80	49.45	– 53.03	44.10	0.59

For each algorithm, the first column indicates the final objective value, the second column indicates the number of (outer) iterations and the third column indicates the CPU time in seconds

(i) $n = m = 7000$, $n_{\mathrm{neg}} = 2450$, $n_{\mathrm{pos}} = 4550$, $\lambda_{\min} = -9.19 \cdot 10^3$, $\lambda_{\max} = 2.18 \cdot 10^4$;
(ii) $n = m = 7000$, $n_{\mathrm{neg}} = 3500$, $n_{\mathrm{pos}} = 3500$, $\lambda_{\min} = -1.13 \cdot 10^4$, $\lambda_{\max} = 1.90 \cdot 10^4$;
(iii) $n = m = 7000$, $n_{\mathrm{neg}} = 4550$, $n_{\mathrm{pos}} = 2450$, $\lambda_{\min} = -1.30 \cdot 10^4$, $\lambda_{\max} = 1.60 \cdot 10^4$;

where n_{neg} denotes the number of negative eigenvalues of $Q^T DQ$, n_{pos} denotes the number of positive eigenvalues of $Q^T DQ$, $\lambda_{\min}$ denotes the smallest eigenvalue of $Q^T DQ$ and $\lambda_{\max}$ denotes the largest eigenvalue of $Q^T DQ$. Since in the non-convex case the final objective value found by an algorithm can depend on the starting point, for each problem we considered 10 different starting points, randomly chosen among the vertices of the unit simplex.

The procedures used to compute the stepsize and the index $j(k)$ in AC2CD, and that used to choose the working set in RCD, were the same described before for the Chebyshev center problem.

In Table 4, we report the results for each algorithm in terms of final objective value, number of (outer) iterations and CPU time in seconds. For each problem, we use the acronyms *sp 1, …, sp 10* to distinguish the results obtained with the 10 considered starting points, while *avg* indicates the results averaged over the 10 runs. We first observe that, for problems (ii) and (iii) , the final objective values found by AC2CD are on average smaller than those found by RCD and MVP, while we have the opposite situation for problem (i). We also see that there is a notable difference between AC2CD and RCD in both CPU time and the number of outer iterations, especially on problem (iii). Finally, in comparison with MVP, we note that AC2CD is on average faster on problems (i) and (ii) , but it is slower on problem (iii).

7 Conclusions

In this paper, a block coordinate descent method has been presented for minimizing a continuously differentiable function subject to one linear equality constraint and simple bounds on the variables. In the proposed method, the working set is chosen according to an almost cyclic strategy that does not use first-order information. So, the whole gradient of the objective function does not need to be computed during the algorithm, leading to high efficiency when the problem dimension is large and the partial derivatives of the objective function are cheap. Global convergence to stationary points has been established under an appropriate assumption on the level set, and linear convergence rate has been proved under standard additional assumptions. Promising numerical results have been obtained on different classes of test problems.

There are a number of open questions that indicate directions in which this work can be extended and that can represent challenging tasks for future research. First, it would be worth investigating if, by suitably modifying the working set selection rule or adding conditions to the stepsize, global convergence can be obtained without Assumption 1. Other interesting questions would be how to generalize the proposed method to problems with more than one linear equality constraint, and how to adjust our approach to realize a parallel algorithmic scheme (for example, by a Jacobi-type approach). We wish to report further results in the future.

References

1. Beck, A.: The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. J. Optim. Theory Appl. **162**(3), 892–919 (2014)
2. Beck, A., Tetruashvili, L.: On the convergence of block coordinate descent type methods. SIAM J. Optim. **23**(4), 2037–2060 (2013)
3. Bertsekas, D.P.: Projected Newton methods for optimization problems with simple constraints. SIAM J. Control Optim. **20**(2), 221–246 (1982)
4. Bomze, I.M., Rinaldi, F., Rota Bulò, S.: First-order methods for the impatient: support identification in finite time with convergent Frank-Wolfe variants. Optimization Online (2018). http://www.optimization-online.org/DB_HTML/2018/07/6694.html
5. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. ACM (1992)
6. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 27 (2011)
7. Chang, K.W., Hsieh, C.J., Lin, C.J.: Coordinate descent method for large-scale l2-loss linear support vector machines. J. Mach. Learn. Res. **9**, 1369–1398 (2008)
8. Cristofari, A., De Santis, M., Lucidi, S., Rinaldi, F.: An active-set algorithmic framework for non-convex optimization problems over the simplex (2018). arXiv preprint arXiv:1703.07761
9. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. J. Mach. Learn. Res. **6**, 1889–1918 (2005)
10. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: Proceedings of the 25th International Conference on Machine Learning, pp. 408–415. ACM (2008)
11. Joachims, T.: Making large-scale support vector machine learning practical. In: Schölkopf, B., Burges, C.J., Smola, A.J. (eds.) Advances in Kernel Methods—Support Vector Learning, B, pp. 169–184. MIT Press, Cambridge (1999)
12. Konnov, I.V.: Selective bi-coordinate variations for resource allocation type problems. Comput. Optim. Appl. **64**(3), 821–842 (2016)
13. Lin, C.J.: On the convergence of the decomposition method for support vector machines. IEEE Trans. Neural Netw. **12**(6), 1288–1298 (2001)
14. Lin, C.J., Lucidi, S., Palagi, L., Risi, A., Sciandrone, M.: Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. J. Optim. Theory Appl. **141**(1), 107–126 (2009)
15. Liuzzi, G., Palagi, L., Piacentini, M.: On the convergence of a Jacobi-type algorithm for singly linearly-constrained problems subject to simple bounds. Optim. Lett. **5**(2), 347–362 (2011)
16. Lucidi, S., Palagi, L., Risi, A., Sciandrone, M.: A convergent decomposition algorithm for support vector machines. Comput. Optim. Appl. **38**(2), 217–234 (2007)
17. Luo, Z.Q., Tseng, P.: On the convergence of the coordinate descent method for convex differentiable minimization. J. Optim. Theory Appl. **72**(1), 7–35 (1992)
18. Manno, A., Palagi, L., Sagratella, S.: Parallel decomposition methods for linearly constrained problems subject to simple bound with application to the SVMs training. Comput. Optim. Appl. **71**(1), 115–145 (2018)
19. Necoara, I.: Random coordinate descent algorithms for multi-agent convex optimization over networks. IEEE Trans. Autom. Control **58**(8), 2001–2012 (2013)
20. Necoara, I., Nesterov, Y., Glineur, F.: Random block coordinate descent methods for linearly constrained optimization over networks. J. Optim. Theory Appl. **173**(1), 227–254 (2017)
21. Necoara, I., Patrascu, A.: A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Comput. Optim. Appl. **57**(2), 307–337 (2014)
22. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)

23. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springer, New York (2013)
24. Palagi, L., Sciandrone, M.: On the convergence of a modified version of SVM light algorithm. Optim. Methods Softw. **20**(2–3), 317–334 (2005)
25. Patrascu, A., Necoara, I.: Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. J. Glob. Optim. **61**(1), 19–46 (2015)
26. Platt, J.C.: Sequential minimal optimization: a fast algorithm for training support vector machines. In: Schölkopf, B., Burges, C.J., Smola, A.J. (eds.) Advances in Kernel Methods—Support Vector Learning, pp. 185–208. MIT Press, Cambridge (1998)
27. Raj, A., Olbrich, J., Gärtner, B., Schölkopf, B., Jaggi, M.: Screening rules for convex problems (2016). arXiv preprint arXiv:1609.07478
28. Reddi, S., Hefny, A., Downey, C., Dubey, A., Sra, S.: Large-scale randomized-coordinate descent methods with non-separable linear constraints (2014). arXiv preprint arXiv:1409.2617
29. Tseng, P.: Descent methods for convex essentially smooth minimization. J. Optim. Theory Appl. **71**(3), 425–463 (1991)
30. Tseng, P., Yun, S.: A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. Comput. Optim. Appl. **47**(2), 179–206 (2010)
31. Wright, S.J.: Coordinate descent algorithms. Math. Program. **151**(1), 3–34 (2015)
32. Xiao, L., Boyd, S.: Optimal scaling of a gradient method for distributed resource allocation. J. Optim. Theory Appl. **129**(3), 469–488 (2006)
33. Xu, S., Freund, R.M., Sun, J.: Solution methodologies for the smallest enclosing circle problem. Comput. Optim. Appl. **25**(1–3), 283–292 (2003)