THE CYCLIC BLOCK CONDITIONAL GRADIENT METHOD FOR CONVEX OPTIMIZATION PROBLEMS[*]

AMIR BECK[†], EDOUARD PAUWELS[†], AND SHOHAM SABACH[†]

Abstract. In this paper we study the convex problem of optimizing the sum of a smooth function and a compactly supported nonsmooth term with a specific separable form. We analyze the block version of the generalized conditional gradient method when the blocks are chosen in a cyclic order. A global sublinear rate of convergence is established for two different stepsize strategies commonly used in this class of methods. Numerical comparisons of the proposed method to both the classical conditional gradient algorithm and its random block version demonstrate the effectiveness of the cyclic block update rule.

Key words. conditional gradient, cyclic block decomposition, iteration complexity, linear oracle, nonsmooth convex minimization, support vector machine

AMS subject classifications. 90C25, 65K05

DOI. 10.1137/15M1008397

1. Introduction. With the growth of size of problems commonly encountered in many applied fields, there is a strong demand for numerical methods featuring low computational cost iterative schemes. By low computational cost, we mean algorithms which require at most matrix by vector multiplication (inversion of matrices are, for example, too expensive). In this context, it is necessary to propose and analyze numerical schemes that

- are based on computationally efficient steps;
- exploit problem structure and data information;
- enjoy global convergence properties and iteration complexity estimates.

We consider structured convex problems consisting of minimizing the sum of two terms: a smooth term, which is a composition of a smooth function and a linear mapping and a nonsmooth separable term. We focus on programs for which the geometry of the nonsmooth part exhibits such a degree of complexity that proximal-based methods [5, 6, 24] do not constitute a viable alternative. Indeed, the efficiency of these methods considerably deteriorates in situations that do not fit in "favorable geometric settings"; see [11, 16] for a more detailed description of this concept. Algorithms based on linear oracles, such as the conditional gradient method (also known as the Frank–Wolfe algorithm) [14, 21, 12, 13, 17] and their extensions to structured composite problems [8, 9, 1], or block separable problems [19], are based on the principle of iteratively solving linearized subproblems. In settings where computing the proximal operator is too expensive, this approach has proven to be competitive. Successful examples of applications which benefit from this approach include trace-norm constrained or penalized problems [18, 11] and structured multiclass classification with an extremely large number of classes [19]. On the theoretical side, for most of the algorithms mentioned in the references above, a sublinear rate of $O(1/k)$, both in function values and duality gap, is available.

Continuously increasing problem dimensions have motivated the principle of taking advantage of available block structure of the problem at hand. This led to the development of variable decomposition methods which break down the original large-scale problem into several much smaller subproblems that could be solved efficiently.

In recent works, [23] and [27] analyzed the average case iteration complexity of block versions of gradient, projected gradient, and forward-backward methods where at each iteration the block to be updated is selected at random. We refer to such a block selection rule as the *random update rule*. In the context of linear oracles, [19] applied this approach to the conditional gradient method focusing on implementing it for the structured Support Vector Machine (SVM) training problem (see more details in section 6.2). The analysis of algorithms involving random update rules typically provides average case complexity results. Different kinds of works consider updating the blocks in a cyclic order. We refer to such a deterministic update rule as the *cyclic update rule*. In this context, [22] provides an asymptotic analysis of exact coordinate minimization for composite strongly convex problems. In [6], a global rate of convergence result was established for the cyclic block coordinate gradient projection method for convex problems over feasible sets with a separable structure. In [28], an explicit rate is given for the block proximal gradient method for ℓ_1 regularized convex problems. For this line of work, the estimates given for the cyclic update rule deterministically hold for the sequence of function values. As we already mentioned, this is not the case for the random update rule for which only average case estimates are available.

Another relevant feature of block decomposition methods is the fact that they usually allow one to take a substantially larger step at each iteration (with respect to each block) compared to their classical nonblock counterparts. This fact potentially gives a numerical advantage to block decomposition algorithms compared to classical variants; see [6].

In this work, we propose a cyclic block version of the generalized conditional gradient method [8, 9, 1] which for ease of reference is called the Cyclic Block Conditional Gradient (CBCG). The word "generalized" means that we consider a more general nonsmooth part, which is not necessarily an indicator function. The word "cyclic" means here that each block is updated once at each iteration. The order in which the blocks are updated may vary arbitrarily between iterations, and our analysis therefore includes the random permutation approach. We provide deterministic and global convergence rate estimates for the predefined stepsize strategy [13, 17] for an adaptive stepsize strategy [21] including a backtracking version of it. These rates hold independently of the order of updating the blocks at each iteration. We also establish rate estimates for an optimality measure in the spirit of [17, 1], which constitutes an additional novelty compared to the results presented in [6]. All the rates proved below have the form of $O(1/k)$ where k is the number of complete iterations (over all blocks). Interestingly, this analysis leads to new convergence results for methods that do not relate directly to linear oracles at first sight. For example, our results lead to explicit deterministic rate estimates in terms of the duality gap for the cyclic and random permutation variants of the Stochastic Dual Coordinate Ascent (SDCA) of [29].

We numerically compare the proposed CBCG method (using both the cyclic and random permutation updating rules) to its random update rule counterpart, that is, the Random Block Conditional Gradient (RBCG), which was proposed and analyzed in [19]. Extensive simulations on a large number of synthetic examples suggest that CBCG is competitive with both RBCG and the classical conditional gradient algo-

rithm (CG). Finally, we also compare CBCG and RBCG on the problem of training the structured SVM [30, 31] for the optical character recognition (OCR) task originally proposed in [30]. In this setting, we observe that the random permutation updating rule has an advantage over the other updating rules.

The next section is dedicated to the presentation of the model and main assumptions (see section 2.1) and to the description of the CBCG algorithm (see section 2.2). Section 3 presents a few auxiliary results for an optimality measure which is commonly encountered when using methods which are based on linear oracles. In section 4 we present our theoretical findings about the rate of convergence results of the CBCG method. We split this section into two subsections which deal with the two different stepsize strategies that we analyze in this paper (section 4.1 for the predefined stepsize and section 4.2 for the adaptive stepsize). We conclude section 4 with a discussion on a backtracking version of the CBCG method (see section 4.3). Section 5 presents an extension of the analysis given in section 4.2 for the specific case where the stepsize is chosen using exact line search for problems in which the smooth part of the objective function is quadratic. This leads to a discussion about the implications for block coordinate descent methods. Numerical experiments on synthetic data and on the structured SVM training problem are presented in section 6.

Conventions. Throughout this paper the underlying vector space is the n-dimensional Euclidean space $\mathbb{R}^n$ with the l_2-norm which is denoted by $\|\cdot\|$. This notation is also used for the matrix norm, which is assumed to be the spectral norm. We will consider the partition of an arbitrary vector $\mathbf{x} \in \mathbb{R}^n$ into N blocks where each block consists of a subset of the n coordinates. The size of each block (the number of coordinates) is given by the integer n_i for $i = 1, 2, \ldots, N$, such that $\sum_{i=1}^{N} n_i = n$. The ith block of a vector $\mathbf{x} \in \mathbb{R}^n$ is denoted by $\mathbf{x}_i$. We assume that $\mathbf{x} \in \mathbb{R}^n$ can be written as follows:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}.$$

For any $i = 1, 2, \ldots, N$ we define the matrix $\mathbf{U}_i \in \mathbb{R}^{n \times n_i}$ as the submatrix of the $n \times n$ identity matrix consisting of the columns corresponding to the ith block. Thus, in particular,

$$(\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N) = \mathbf{I}_n.$$

It is clear that, using these notations, we have for any $\mathbf{x} \in \mathbb{R}^n$ that $\mathbf{x}_i = \mathbf{U}_i^T \mathbf{x}$ and $\mathbf{x} = \sum_{i=1}^{N} \mathbf{U}_i \mathbf{x}_i$. Finally, for any subset S of $\mathbb{R}^n$, δ_S denotes the indicator function of S which takes the value 0 on S and $+\infty$ otherwise.

2. The optimization model and algorithm.

2.1. Problem formulation and assumptions. We consider the optimization model

$$(2.1) \qquad \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ H(\mathbf{x}) \equiv F(\mathbf{A}\mathbf{x}) + \sum_{i=1}^{N} g_i(\mathbf{x}_i) \right\},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$. We make the following standing assumption on model (2.1).

ASSUMPTION 1.

(i) $g_i : \mathbb{R}^{n_i} \to (-\infty, \infty]$, $i = 1, 2, \ldots, N$, *is a proper, closed, and convex function which satisfies the following:*

  − $X_i \equiv \operatorname{dom} g_i \subseteq \mathbb{R}^{n_i}$ *is a compact set with diameter* D_i, *that is,*

$$\sup_{\mathbf{x}_i, \mathbf{y}_i \in X_i} \|\mathbf{x}_i - \mathbf{y}_i\| = D_i.$$

  − g_i *is globally Lipschitz on* X_i *with constant* l_i, *that is,*

$$(2.2) \qquad |g_i(\mathbf{x}_i) - g_i(\mathbf{y}_i)| \leq l_i \|\mathbf{x}_i - \mathbf{y}_i\| \quad \forall\, \mathbf{x}_i, \mathbf{y}_i \in X_i.$$

(ii) $F : \mathbb{R}^m \to \mathbb{R}$ *is convex and continuously differentiable*[1] *over*

$$\mathbf{A}(X_1 \times X_2 \times \cdots \times X_N) \subseteq \mathbb{R}^m$$

and has Lipschitz continuous gradient with constant L_F, *that is,*

$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L_F \|\mathbf{x} - \mathbf{y}\| \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbf{A}(X_1 \times X_2 \times \cdots \times X_N).$$

Since g_i is assumed to be convex, it immediately implies that the domain X_i is a convex subset of $\mathbb{R}^{n_i}$ for $i = 1, 2, \ldots, N$. We set $g(\mathbf{x}) \equiv \sum_{i=1}^N g_i(\mathbf{x}_i)$ and $f(\mathbf{x}) \equiv F(\mathbf{Ax})$. The domain of g is denoted by $\operatorname{dom} g \equiv X$ and its diameter is denoted by D. Using these simplified notations, problem (2.1) actually consists of minimizing the sum $f + g$. It holds that $X \equiv X_1 \times X_2 \times \cdots \times X_N$ and, therefore,

$$(2.3) \qquad D^2 = \sum_{i=1}^N D_i^2.$$

Remark 2.1. By setting $g(\cdot) = \delta_X(\cdot)$, we recover the constrained optimization model that motivated the development of the traditional conditional gradient method (see [17] and the references therein). This is also the case when we add a linear term to the indicator.

Under Assumption 1, problem (2.1) is guaranteed to attain its optimal value, and therefore the optimal set, which is denoted by X^*, is nonempty and the corresponding optimal value is denoted by $H^* \in \mathbb{R}$. For each block $i \in \{1, 2, \ldots, N\}$, we employ the following notation:

- $\mathbf{A}_i \equiv \mathbf{A}\mathbf{U}_i$, so that $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N)$;
- $\nabla_i f(\mathbf{x}) \equiv \mathbf{U}_i^T \nabla f(\mathbf{x})$ denotes the partial gradient of f.

Using the previous notation, we have that $\nabla_i f(\mathbf{x}) = \mathbf{A}_i^T \nabla F(\mathbf{Ax})$. We will use a refined notion of Lipschitz continuity that fits our block separable composite setting. This is expressed by the following standing assumption.

ASSUMPTION 2. *For each* $i = 1, 2, \ldots, N$, *there exists a constant* $\beta_i > 0$, *such that for any* $\mathbf{x} \in X$ *and any* $\mathbf{h}_i \in \mathbb{R}^{n_i}$ *satisfying* $\mathbf{x} + \mathbf{U}_i \mathbf{h}_i \in X$, *it holds that*

$$\|\nabla F(\mathbf{Ax} + \mathbf{A}_i \mathbf{h}_i) - \nabla F(\mathbf{Ax})\| \leq \beta_i \|\mathbf{A}_i \mathbf{h}_i\|.$$

Assumption 2 can be seen as a consequence of Assumption 1. Indeed, it is always possible to set $\beta_i = L_F$, $i = 1, 2, \ldots, N$. However, adopting this more refined

---

[1] A function is continuously differentiable over a given set D if it is continuously differentiable over an open set containing D.

convention provides additional algorithmic flexibility which allows us to take advantage of conditioning disparities between different blocks by using stepsizes which are functions of the Lipschitz constants of the blocks, rather than the global Lipschitz constant. See, for example, section 5.1, where it is shown how this approach allows the usage of exact line search for quadratic problems. We also note that when $\mathbf{A} = \mathbf{I}$, then β_i can be chosen to be the ith block Lipschitz constant of the gradient of F (see, e.g., [6, 2]), which is always a quantity smaller than L_F. We define the following quantity:

$$(2.4) \qquad \beta_{\min} \equiv \min\{\beta_1, \beta_2, \ldots, \beta_N\} > 0.$$

The following important result will play a central role in the forthcoming analysis. The proof is almost identical to the well-known proof of the descent lemma (see [7]) and is thus given in the appendix.

LEMMA 2.2 (composite block descent lemma). *Let $i \in \{1, 2, \ldots, N\}$; then for any $\mathbf{x} \in X$ and $\mathbf{h}_i \in \mathbb{R}^{n_i}$ such that $\mathbf{x} + \mathbf{U}_i\mathbf{h}_i \in X$, we have*

$$(2.5) \qquad f(\mathbf{x} + \mathbf{U}_i\mathbf{h}_i) \le f(\mathbf{x}) + \langle \nabla_i f(\mathbf{x}), \mathbf{h}_i \rangle + \frac{\beta_i}{2} \|\mathbf{A}_i\mathbf{h}_i\|^2.$$

2.2. The cyclic block conditional gradient (CBCG) method. The generalized conditional gradient method [8, 9, 1] can be applied to problem (2.1) when the corresponding linear oracle is available. Therefore, we assume that for any $\mathbf{x} \in X$, the solution of the following problem can be easily computed:

$$\min_{\mathbf{v} \in X} \{\langle \nabla f(\mathbf{x}), \mathbf{v} \rangle + g(\mathbf{v})\}.$$

We exploit here the separability of the function g (see section 2.1) and propose a block decomposition extension of the generalized conditional gradient method which we call the Cyclic Block Conditional Gradient (CBCG) method. Before stating the algorithm, we will need the following additional notation. Let $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ be a given sequence; then for any $i = 1, 2, \ldots, N$, we define

$$(2.6) \qquad \mathbf{x}^{k,i} = \begin{pmatrix} \mathbf{x}_1^{k+1} \\ \vdots \\ \mathbf{x}_i^{k+1} \\ \mathbf{x}_{i+1}^k \\ \vdots \\ \mathbf{x}_N^k \end{pmatrix}.$$

That is, the first i blocks in $\mathbf{x}^{k,i}$ are those of $\mathbf{x}^{k+1}$ and the remaining $N - i$ blocks are those of $\mathbf{x}^k$. It is clear that using this notation we have $\mathbf{x}^{k,0} = \mathbf{x}^k$ and $\mathbf{x}^{k+1} = \mathbf{x}^{k,N}$. The algorithm is given now.

CBCG: Cyclic Block Conditional Gradient
Initialization. $\mathbf{x}^0 \in X$ and $\alpha_i^k \in [0, 1]$ for all $k \in \mathbb{N}$ and $i = 1, 2, \dots, N$.
General step. For $k = 1, 2, \dots$,
   (i) For any $i = 1, 2, \dots, N$, compute

$$(2.7) \qquad \mathbf{p}_i^k \in \operatorname{argmin}_{\mathbf{p}_i \in X_i} \left\{ \langle \nabla_i f(\mathbf{x}^{k,i-1}), \mathbf{p}_i \rangle + g_i(\mathbf{p}_i) \right\},$$

   and then

$$(2.8) \qquad \mathbf{x}^{k,i} = \mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1}).$$

  (ii) Update $\mathbf{x}^{k+1} = \mathbf{x}^{k,N}$.

We will first analyze, in section 4.1, the convergence rate of the CBCG method using a predefined stepsize [13]. Here, the predefined stepsize that we use is given, for any $i = 1, 2, \dots, N$, by

$$\alpha_i^k = \alpha^k \equiv \frac{2}{k+2}.$$

In section 4.2, we will consider an adaptive stepsize rule [21], which is determined by the minimization of the quadratic upper bound of H related to (2.5). The expression of this stepsize will be made precise below (see (4.12)). The backtracking variant of the CBCG with adaptive stepsize rule is presented and analyzed in section 4.3.

3. The optimality measure. In this section, we describe a few properties of an optimality measure including its block counterparts. This measure is typical when discussing methods which are based on linear oracles and usually plays a crucial role in the convergence analysis; see [4] for an overview and [19, 1] for a link with Fenchel duality. For any $\mathbf{x} \in X$, we define the following quantity:

$$(3.1) \qquad p(\mathbf{x}) \in \operatorname{argmin}_{\mathbf{p} \in X} \left\{ \langle \nabla f(\mathbf{x}), \mathbf{p} \rangle + g(\mathbf{p}) \right\},$$

as well as the optimality measure
(3.2)
$$S(\mathbf{x}) \equiv \max_{\mathbf{p} \in X} \left\{ \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{p} \rangle + g(\mathbf{x}) - g(\mathbf{p}) \right\} = \langle \nabla f(\mathbf{x}), \mathbf{x} - p(\mathbf{x}) \rangle + g(\mathbf{x}) - g(p(\mathbf{x})),$$

where the last equality follows from (3.1). The function S is an optimality measure in the sense that it is nonnegative on X, and it is zero only on X^*. Furthermore, for any $\mathbf{x} \in X$, the quantity $S(\mathbf{x})$ is an upper bound on $H(\mathbf{x}) - H^*$ as stated in the following lemma, whose short proof is given for the sake of completeness.

LEMMA 3.1. $S(\mathbf{x}) \geq H(\mathbf{x}) - H^*$.
Proof. For any $\mathbf{x}^* \in X^*$ we have

$$\begin{aligned}
S(\mathbf{x}) &= \langle \nabla f(\mathbf{x}), \mathbf{x} - p(\mathbf{x}) \rangle + g(\mathbf{x}) - g(p(\mathbf{x})) \\
&= \langle \nabla f(\mathbf{x}), \mathbf{x} \rangle + g(\mathbf{x}) - [\langle \nabla f(\mathbf{x}), p(\mathbf{x}) \rangle + g(p(\mathbf{x}))] \\
&\geq \langle \nabla f(\mathbf{x}), \mathbf{x} \rangle + g(\mathbf{x}) - [\langle \nabla f(\mathbf{x}), \mathbf{x}^* \rangle + g(\mathbf{x}^*)] \\
&= \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^* \rangle + g(\mathbf{x}) - g(\mathbf{x}^*),
\end{aligned}$$

where the inequality follows from (3.1). Using the convexity of f, we obtain that

$$S(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^* \rangle + g(\mathbf{x}) - g(\mathbf{x}^*) \geq f(\mathbf{x}) - f(\mathbf{x}^*) + g(\mathbf{x}) - g(\mathbf{x}^*) = H(\mathbf{x}) - H^*.$$

This proves the desired result. □

We refine the notation introduced in (3.1) and (3.2) in order to fit them to our block structured setting. For any $\mathbf{x} \in X$ and any $i = 1, 2, \ldots, N$, we set

$$(3.3) \qquad p_i(\mathbf{x}) \in \operatorname{argmin}_{\mathbf{p}_i \in X_i} \left\{ \langle \nabla_i f(\mathbf{x}), \mathbf{p}_i \rangle + g_i(\mathbf{p}_i) \right\},$$

and define the block optimality measure, which was already introduced in [19] when $g_i \equiv 0$,

$$(3.4) \qquad S_i(\mathbf{x}) \equiv \max_{\mathbf{p}_i \in X_i} \left\{ \langle \nabla f_i(\mathbf{x}), \mathbf{x}_i - \mathbf{p}_i \rangle + g_i(\mathbf{x}_i) - g_i(\mathbf{p}_i) \right\}.$$

It is clear that in this case it is also true that

$$(3.5) \qquad S_i(\mathbf{x}) = \langle \nabla f_i(\mathbf{x}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle + g_i(\mathbf{x}_i) - g_i(p_i(\mathbf{x})).$$

Using the separability of both g and X, we have for any $\mathbf{x} \in X$ that

$$(3.6) \qquad S(\mathbf{x}) = \sum_{i=1}^{N} S_i(\mathbf{x}).$$

There might be multiple optimal solutions for problem (3.1) and also for problem (3.3). Our only assumption is that the choices of $p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_N(\mathbf{x})$ and $p(\mathbf{x})$ are made under the restriction that

$$p(\mathbf{x}) = \begin{pmatrix} p_1(\mathbf{x}) \\ p_2(\mathbf{x}) \\ \vdots \\ p_N(\mathbf{x}) \end{pmatrix}.$$

The following Lipschitz-type property of the block optimality measure S_i, $i = 1, 2, \ldots, N$, will be crucial in the forthcoming analysis.

LEMMA 3.2. *Let* $\mathbf{x}, \mathbf{y} \in X$ *be two vectors which satisfy* $\mathbf{x}_i = \mathbf{y}_i$ *for some* $i = 1, 2, \ldots, N$. *Then the following inequality holds:*

$$|S_i(\mathbf{x}) - S_i(\mathbf{y})| \le L_F D_i \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|.$$

Proof. From (3.5) we have

(3.7)
$$\begin{aligned} S_i(\mathbf{x}) &= \langle \nabla f_i(\mathbf{x}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle + g_i(\mathbf{x}_i) - g_i(p_i(\mathbf{x})) \\ &= \langle \nabla f_i(\mathbf{y}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle + g_i(\mathbf{x}_i) - g_i(p_i(\mathbf{x})) + \langle \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle. \end{aligned}$$

Now, using the fact that $f(\mathbf{x}) \equiv F(A\mathbf{x})$ and Assumption 1(ii), we obtain

$$\begin{aligned} \langle \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle &= \langle \mathbf{A}_i^T(\nabla F(\mathbf{A}\mathbf{x}) - \nabla F(\mathbf{A}\mathbf{y})), \mathbf{x}_i - p_i(\mathbf{x}) \rangle \\ &= \langle \nabla F(\mathbf{A}\mathbf{x}) - \nabla F(\mathbf{A}\mathbf{y}), \mathbf{A}_i(\mathbf{x}_i - p_i(\mathbf{x})) \rangle \\ &\le \|\nabla F(\mathbf{A}\mathbf{x}) - \nabla F(\mathbf{A}\mathbf{y})\| \cdot \|\mathbf{A}_i(\mathbf{x}_i - p_i(\mathbf{x}))\| \\ &\le L_F \|\mathbf{A}(\mathbf{x} - \mathbf{y})\| \cdot \|\mathbf{A}_i(\mathbf{x}_i - p_i(\mathbf{x}))\| \\ (3.8) &\le L_F D_i \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|, \end{aligned}$$

where the first inequality follows from the Cauchy–Schwarz inequality and the last inequality follows from the fact that both $\mathbf{x}_i$ and $p_i(\mathbf{x})$ belong to X_i (see Assumption 1(i)). Finally, by combining (3.7) with (3.8), and using the fact that $\mathbf{x}_i = \mathbf{y}_i$, we obtain that

$$
\begin{aligned}
S_i(\mathbf{x}) &\le \langle \nabla f_i(\mathbf{y}), \mathbf{x}_i - p_i(\mathbf{x}) \rangle + g_i(\mathbf{x}_i) - g_i(p_i(\mathbf{x})) + L_F D_i \, \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\| \\
&= \langle \nabla f_i(\mathbf{y}), \mathbf{y}_i - p_i(\mathbf{x}) \rangle + g_i(\mathbf{y}_i) - g_i(p_i(\mathbf{x})) + L_F D_i \, \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|
\end{aligned}
$$
$$
(3.9) \qquad \le S_i(\mathbf{y}) + L_F D_i \, \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|,
$$

where the last inequality follows from the definition of S_i (see (3.4)). Changing the roles of $\mathbf{x}$ and $\mathbf{y}$, we also obtain that

$$
S_i(\mathbf{y}) \le S_i(\mathbf{x}) + L_F D_i \, \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|,
$$

which along with (3.9) yields the desired result. $\quad\square$

4. Convergence analysis of the CBCG method. This section is devoted to the convergence analysis of the CBCG algorithm. We will first prove, in section 4.1, a sublinear convergence rate for the variant with the predefined stepsize rule. A similar rate of convergence will be then established, in section 4.2, for the variant with the adaptive stepsize rule. Finally, we describe a backtracking procedure in section 4.3 which allows us to use the CBCG method when the constants β_i, $i = 1, 2, \ldots, N$, given in Assumption 2 are unknown in advance. We begin with an extension of Lemma 2.2 that holds for any choice of stepsize.

LEMMA 4.1. *Let* $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ *be the sequence generated by the CBCG method. Then, for any* $k \ge 0$ *and* $i \in \{1, 2, \ldots, N\}$*, we have*

$$
H(\mathbf{x}^{k,i}) \le H(\mathbf{x}^{k,i-1}) - \alpha_i^k S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2.
$$

Proof. First, by the definition of the main step of the CBCG method (see (2.8)), we have

$$
\begin{aligned}
H(\mathbf{x}^{k,i}) &= f(\mathbf{x}^{k,i}) + g(\mathbf{x}^{k,i}) \\
&= f(\mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})) + g(\mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})).
\end{aligned}
$$

We can now use Lemma 2.2 to obtain

$$
\begin{aligned}
H(\mathbf{x}^{k,i}) &\le f(\mathbf{x}^{k,i-1}) + \alpha_i^k \langle \nabla_i f(\mathbf{x}^{k,i-1}), \mathbf{p}_i^k - \mathbf{x}_i^{k,i-1} \rangle \\
(4.1) &\qquad + \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\|^2 + g(\mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})).
\end{aligned}
$$

The last term can be bounded from above as follows:

$$
\begin{aligned}
g(\mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})) &= \sum_{j=1, j \ne i}^{N} g_j(\mathbf{x}_j^{k,i-1}) + g_i((1 - \alpha_i^k)\mathbf{x}_i^{k,i-1} + \alpha_i^k \mathbf{p}_i^k) \\
&\le \sum_{j=1, j \ne i}^{N} g_j(\mathbf{x}_j^{k,i-1}) + (1 - \alpha_i^k)g_i(\mathbf{x}_i^{k,i-1}) + \alpha_i^k g_i(\mathbf{p}_i^k) \\
(4.2) &= g(\mathbf{x}^{k,i-1}) + \alpha_i^k(g_i(\mathbf{p}_i^k) - g_i(\mathbf{x}_i^{k,i-1})),
\end{aligned}
$$

where the inequality follows from the convexity of each g_i, $i = 1, 2, \ldots, N$. Now, combining (4.1) with (4.2) and using the definition of S_i (see (3.5)), we obtain

$$
\begin{aligned}
H(\mathbf{x}^{k,i}) \leq {} & f(\mathbf{x}^{k,i-1}) + g(\mathbf{x}^{k,i-1}) - \alpha_i^k(\langle \nabla_i f(\mathbf{x}^{k,i-1}), \mathbf{x}_i^{k,i-1} - \mathbf{p}_i^k \rangle \\
& + g_i(\mathbf{x}_i^{k,i-1}) - g_i(\mathbf{p}_i^k)) \\
& + \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\|^2 \\
= {} & H(\mathbf{x}^{k,i-1}) - \alpha_i^k S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\|^2 \\
= {} & H(\mathbf{x}^{k,i-1}) - \alpha_i^k S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2,
\end{aligned}
$$

where the last equality follows from the fact that $\mathbf{x}_i^k = \mathbf{x}_i^{k,i-1}$ (see (2.6)). This proves the desired result. $\quad\Box$

We also need an explicit expression of the distance between two consecutive iterates generated by the CBCG method. The following lemma also holds for any choice of stepsize.

LEMMA 4.2. *Let $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the CBCG method. Then, for any $k \in \mathbb{N}$, we have*

$$
\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 = \sum_{j=1}^{N} (\alpha_j^k)^2 \|\mathbf{p}_j^k - \mathbf{x}_j^{k,j-1}\|^2.
$$

Furthermore, for any $i = 0, 1, \ldots, N$, we have

$$
\|\mathbf{x}^{k,i} - \mathbf{x}^k\|^2 \leq \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2.
$$

Proof. For any fixed $k \in \mathbb{N}$, we have from the definition of the iterates of the CBCG algorithm (see (2.8)) and (2.6) that

$$
\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 = \sum_{j=1}^{N} \|\mathbf{x}_j^{k,N} - \mathbf{x}_j^k\|^2 = \sum_{j=1}^{N} (\alpha_j^k)^2 \|\mathbf{p}_j^k - \mathbf{x}_j^{k,j-1}\|^2,
$$

which proves the first statement. Furthermore, since for any $i = 0, 1, \ldots, N$, $\mathbf{x}_j^{k,i} = \mathbf{x}_j^k$ for all $j > i$ and $\mathbf{x}_j^{k,i} = \mathbf{x}^{k,N}$ for all $j \leq i$ we have

$$
\begin{aligned}
\|\mathbf{x}^{k,i} - \mathbf{x}^k\|^2 &= \sum_{j=1}^{N} \|\mathbf{x}_j^{k,i} - \mathbf{x}_j^k\|^2 = \sum_{j=1}^{i} \|\mathbf{x}_j^{k,i} - \mathbf{x}_j^k\|^2 = \sum_{j=1}^{i} \|\mathbf{x}_j^{k,N} - \mathbf{x}_j^k\|^2 \\
&\leq \sum_{j=1}^{N} \|\mathbf{x}_j^{k,N} - \mathbf{x}_j^k\|^2 = \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2,
\end{aligned}
$$

which proves the second statement, and the proof is complete. $\quad\Box$

4.1. Analysis for the predefined stepsize strategy. In this section, we analyze the CBCG algorithm with the predefined stepsize given by

$$
(4.3) \qquad\qquad \alpha_i^k = \alpha^k \equiv \frac{2}{k+2}.
$$

Note that this stepsize rule is completely blind to disparities between blocks and therefore does not allow one to counterbalance them in a way that will increase the convergence speed. The main step towards the proof of convergence rate in this case is recorded in the following lemma.

LEMMA 4.3. *Let $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ be the sequence generated by the CBCG method with the stepsize given in (4.3). Then, for any $k \geq 0$, we have*

$$H(\mathbf{x}^{k+1}) \leq H(\mathbf{x}^k) - \alpha^k S(\mathbf{x}^k) + \frac{C_1}{2}(\alpha^k)^2,$$

where

$$(4.4) \qquad C_1 = \sum_{i=1}^{N} \beta_i \|\mathbf{A}_i\|^2 D_i^2 + 2L_F D \|\mathbf{A}\| \sum_{i=1}^{N} D_i \|\mathbf{A}_i\|.$$

Proof. For any $k \geq 0$ and each $i \in \{1, 2, \ldots, N\}$, we have from Lemma 4.1 that

$$H(\mathbf{x}^{k,i}) \leq H(\mathbf{x}^{k,i-1}) - \alpha^k S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2$$

$$(4.5) \qquad \leq H(\mathbf{x}^{k,i-1}) - \alpha^k S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha^k)^2 \beta_i \|\mathbf{A}_i\|^2 D_i^2}{2}.$$

Summing (4.5) for $i = 1, 2, \ldots, N$ yields

$$H(\mathbf{x}^{k+1}) = H(\mathbf{x}^{k,N})$$

$$\leq H(\mathbf{x}^{k,0}) - \alpha^k \sum_{i=1}^{N} S_i(\mathbf{x}^{k,i-1}) + \frac{(\alpha^k)^2}{2} \sum_{i=1}^{N} (\beta_i \|\mathbf{A}_i\|^2 D_i^2)$$

$$= H(\mathbf{x}^k) - \alpha^k \sum_{i=1}^{N} S_i(\mathbf{x}^k) + \frac{(\alpha^k)^2}{2} \sum_{i=1}^{N} (\beta_i \|\mathbf{A}_i\|^2 D_i^2)$$

$$+ \alpha^k \sum_{i=1}^{N} (S_i(\mathbf{x}^k) - S_i(\mathbf{x}^{k,i-1}))$$

$$= H(\mathbf{x}^k) - \alpha^k S(\mathbf{x}^k) + \frac{(\alpha^k)^2}{2} \sum_{i=1}^{N} (\beta_i \|\mathbf{A}_i\|^2 D_i^2)$$

$$(4.6) \qquad + \alpha^k \sum_{i=1}^{N} (S_i(\mathbf{x}^k) - S_i(\mathbf{x}^{k,i-1})),$$

where the last equality follows from (3.6). Using Lemma 3.2, and the fact that $\mathbf{x}_i^{k,i-1} = \mathbf{x}_i^k$ (see (2.6)), gives, for any $i = 1, 2, \ldots, N$,

$$(4.7) \qquad S_i(\mathbf{x}^k) - S_i(\mathbf{x}^{k,i-1}) \leq L_F D_i \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|.$$

Combining (4.6) and (4.7) yields

$$H(\mathbf{x}^{k+1}) \leq H(\mathbf{x}^k) - \alpha^k S(\mathbf{x}^k) + \frac{(\alpha^k)^2}{2} \sum_{i=1}^{N} (\beta_i \|\mathbf{A}_i\|^2 D_i^2)$$

$$(4.8) \qquad + \alpha^k L_F \sum_{i=1}^{N} D_i \|\mathbf{A}_i\| \cdot \|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|.$$

From Lemma 4.2, for any $i = 1, 2, \ldots, N$, we have

$$\|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|^2 \leq \|\mathbf{A}\|^2 \cdot \|\mathbf{x}^k - \mathbf{x}^{k,i-1}\|^2 \leq \|\mathbf{A}\|^2 \cdot \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2$$

$$= (\alpha^k)^2 \|\mathbf{A}\|^2 \sum_{j=1}^N \|\mathbf{p}_j^k - \mathbf{x}_j^k\|^2 \leq (\alpha^k)^2 \|\mathbf{A}\|^2 \sum_{j=1}^N D_j^2$$

$$(4.9) \qquad\qquad = (\alpha^k)^2 \|\mathbf{A}\|^2 D^2.$$

Combining (4.8) and (4.9) yields

$$H(\mathbf{x}^{k+1}) \leq H(\mathbf{x}^k) - \alpha^k S(\mathbf{x}^k) + \frac{(\alpha^k)^2}{2} \left[\sum_{i=1}^N \beta_i \|\mathbf{A}_i\|^2 D_i^2 + 2L_F D \|\mathbf{A}\| \sum_{i=1}^N D_i \|\mathbf{A}_i\| \right],$$

which proves the desired result. $\quad\square$

We now recall the following technical lemma on sequences of nonnegative numbers (cf. [1, Lemma 1]).

LEMMA 4.4. *Let* $\{a_k\}_{k \in \mathbb{N}}$ *and* $\{b_k\}_{k \in \mathbb{N}}$ *be two sequences satisfying, for any* $k \geq 0$, *that*

$$(4.10) \qquad\qquad a_{k+1} \leq a_k - t_k b_k + \frac{C}{2} t_k^2,$$

where $0 \leq a_k \leq b_k$, $t_k = 2/(k+2)$, *and* $C > 0$. *Then*
   (a) $a_k \leq 2C/(k+1)$;
   (b) *for any* $n \geq 1$, *we have*

$$\min_{k=\lfloor n/2 \rfloor, \ldots, n} b_k \leq \frac{8C}{n}.$$

We are now ready to establish the sublinear rate of convergence of the CBCG algorithm with predefined stepsize.

THEOREM 4.5 (sublinear rate for CBCG with predefined stepsize). *Let* $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ *be the sequence generated by the CBCG method with the predefined stepsize strategy given in* (4.3). *Then, for any* $k \geq 0$, *we have*

$$(4.11) \qquad\qquad H(\mathbf{x}^k) - H^* \leq \frac{2C_1}{k+1},$$

and, for any $n \geq 1$, *we have*

$$\min_{k=\lfloor n/2 \rfloor, \ldots, n} S(\mathbf{x}^k) \leq \frac{8C_1}{n},$$

where

$$C_1 = \sum_{i=1}^N \beta_i \|A_i\|^2 D_i^2 + 2L_F D \|A\| \sum_{i=1}^N D_i \|A_i\|.$$

Proof. From Lemma 4.3 we have

$$H(\mathbf{x}^{k+1}) - H^* \leq H(\mathbf{x}^k) - H^* - \alpha^k S(\mathbf{x}^k) + \frac{C_1}{2} (\alpha^k)^2.$$

In addition, by Lemma 3.1, we have that $S(\mathbf{x}^k) \geq H(\mathbf{x}^k) - H^*$. We can therefore invoke Lemma 4.4 with $a_k = H(\mathbf{x}^k) - H^*$ and $b_k = S(\mathbf{x}^k)$, and the desired result is established. $\quad\square$

4.2. Analysis for the adaptive stepsize strategy. We now turn to the analysis of CBCG with adaptive stepsize. The main insight is to choose a stepsize that minimizes the quadratic upper bound given in Lemma 4.1. In this setting, the adaptive stepsize is defined as follows:

$$\alpha_i^k = \operatorname{argmin}_{\alpha \in [0,1]} \left\{ -\alpha S_i(\mathbf{x}^{k,i-1}) + \alpha^2 \cdot \frac{\beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2 \right\}$$

$$(4.12) \qquad = \min \left\{ \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2}, 1 \right\}.$$

Note that by Lemma 4.1, this choice of stepsize leads to a decrease of objective value at each step. The analysis employed for the predefined stepsize does not seem to be easily adjusted to the adaptive stepsize rule. Thus, a different analysis is developed in this section. We begin with the following technical result.

LEMMA 4.6. *Let* $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ *be the sequence generated by the CBCG method with the adaptive stepsize strategy given in* (4.12). *Then, for any* $k \geq 0$ *and* $i \in \{1, 2, \ldots, N\}$, *we have*

$$H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{\alpha_i^k}{2} S_i(\mathbf{x}^{k,i-1}) \geq \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2.$$

Proof. We split the proof into two cases. First, if

$$(4.13) \qquad \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2} \geq 1,$$

then $\alpha_i^k = 1$, and by Lemma 4.1 we have

$$H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \alpha_i^k S_i(\mathbf{x}^{k,i-1}) - \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2$$

$$(4.14) \qquad = S_i(\mathbf{x}^{k,i-1}) - \frac{\beta_i}{2} \|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{p}_i^k)\|^2$$

$$(4.15) \qquad \geq \frac{\alpha_i^k}{2} S_i(\mathbf{x}^{k,i-1}) \geq \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{p}_i^k)\|^2,$$

where the last two inequalities follow from (4.13) and the fact that $\alpha_i^k = 1$. In the second case, when $S_i(\mathbf{x}^{k,i-1}) < \beta_i \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2$, we have that

$$\alpha_i^k = \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2}.$$

Thus,

$$H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \alpha_i^k S_i(\mathbf{x}^{k,i-1}) - \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2$$

$$= \frac{S_i(\mathbf{x}^{k,i-1})^2}{2\beta_i \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2}$$

$$(4.16) \qquad = \frac{\alpha_i^k}{2} S_i(\mathbf{x}^{k,i-1}) = \frac{(\alpha_i^k)^2 \beta_i}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2.$$

The result now follows by combining (4.15) and (4.16). ☐

We can now prove the following important result.

LEMMA 4.7. *Let $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the CBCG method with the adaptive stepsize strategy given in (4.12). Then, for any $k \geq 0$ and $i \in \{1, 2, \ldots, N\}$, we have*

$$H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{S_i(\mathbf{x}^{k,i-1})^2}{2\max\{\beta_i\|\mathbf{A}_i\|^2 D_i^2, K_i\}},$$

where, for each $i = 1, 2, \ldots, N$,

$$(4.17) \qquad\qquad\qquad M_i = \max_{\mathbf{x} \in X} \|\nabla_i f(\mathbf{x})\|,$$

$$(4.18) \qquad\qquad\qquad K_i = (M_i + l_i)D_i.$$

Proof. We again split the proof into two cases. First, if $\alpha_i^k = 1$, then by Lemma 4.6 we have

$$(4.19) \qquad H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{1}{2}S_i(\mathbf{x}^{k,i-1}) \geq \frac{S_i(\mathbf{x}^{k,i-1})^2}{2K_i},$$

where the last inequality follows from the fact that for any $\mathbf{x} \in X$

$$
\begin{aligned}
S_i(\mathbf{x}) &= \langle \nabla_i f(\mathbf{x}), \mathbf{x}_i - \mathbf{p}_i(\mathbf{x})\rangle + g_i(\mathbf{x}_i) - g_i(\mathbf{p}_i(\mathbf{x}))\\
&\leq \|\nabla_i f(\mathbf{x})\| \cdot \|\mathbf{x}_i - \mathbf{p}_i(\mathbf{x})\| + l_i \|\mathbf{x}_i - \mathbf{p}_i(\mathbf{x})\|\\
(4.20) \qquad &\leq (M_i + l_i)D_i.
\end{aligned}
$$

In the second case, when $\alpha_i^k < 1$, we have that $\alpha_i^k = \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i\|\mathbf{A}_i(\mathbf{p}_i^k-\mathbf{x}_i^k)\|^2}$, and by Lemma 4.6 we can write

$$(4.21) \qquad H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{S_i(\mathbf{x}^{k,i-1})^2}{2\beta_i\|\mathbf{A}_i(\mathbf{p}_i^k-\mathbf{x}_i^k)\|^2} \geq \frac{S_i(\mathbf{x}^{k,i-1})^2}{2\beta_i\|\mathbf{A}_i\|^2 D_i^2}.$$

The result now follows by combining (4.20) and (4.21). $\qquad\square$

We will now prove an additional technical lemma that establishes a "sufficient decrease" property of the CBCG method with adaptive stepsize.

LEMMA 4.8. *Let $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the CBCG method with the adaptive stepsize strategy given in (4.12). Then, for any $k \geq 0$ and $i \in \{1, 2, \ldots, N\}$, we have*

$$H(\mathbf{x}^k) - H(\mathbf{x}^{k+1}) \geq \frac{\beta_{min}}{2N}\|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|^2,$$

where $\beta_{\min}$ is given by (2.4).

Proof. By Lemma 4.6 we have

$$(4.22) \qquad H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{(\alpha_i^k)^2\beta_i}{2}\|\mathbf{A}_i(\mathbf{p}_i^k-\mathbf{x}_i^k)\|^2.$$

Now, for any $i \in \{1, 2, \ldots, N\}$, we can write

$$\|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|^2 = \left\|\sum_{j=1}^{N} \mathbf{A}_j(\mathbf{x}_j^k - \mathbf{x}_j^{k,i-1})\right\|^2$$

$$\leq N \sum_{j=1}^{N} \|\mathbf{A}_j(\mathbf{x}_j^k - \mathbf{x}_j^{k,i-1})\|^2$$

$$= N \sum_{j=1}^{i-1} \|\mathbf{A}_j(\mathbf{x}_j^k - \mathbf{x}_j^{k,i-1})\|^2$$

$$= N \sum_{j=1}^{i-1} (\alpha_i^k)^2 \|\mathbf{A}_j(\mathbf{x}_j^k - \mathbf{p}_j^k)\|^2$$

$$\leq \frac{2N}{\beta_{min}} \sum_{j=1}^{i-1} (H(\mathbf{x}^{k,j-1}) - H(\mathbf{x}^{k,j}))$$

$$= \frac{2N}{\beta_{min}} (H(\mathbf{x}^k) - H(\mathbf{x}^{k,i-1}))$$

$$\leq \frac{2N}{\beta_{min}} (H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})),$$

where the first inequality follows from the fact that for any N vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$, the inequality $\left\|\sum_{j=1}^{N} \mathbf{u}_j\right\|^2 \leq N \sum_{j=1}^{N} \|\mathbf{u}_j\|^2$ holds, the second inequality follows from (4.22), and the last inequality follows from Lemma 4.6, which shows that the sequence of function values is nonincreasing. □

Remark 4.9. The bound in Lemma 4.8 can be improved in some situations where additional information on the structure of $\mathbf{A}$ is available. For example, if the column space of each $\mathbf{A}_i$ span orthogonal spaces, that is, $\mathbf{A}_i^T \mathbf{A}_j = \mathbf{0}$ for any $1 \leq i < j \leq N$, then the factor N can be avoided.

The next lemma constitutes the crucial step towards the establishment of a sublinear convergence rate of the CBCG method with adaptive stepsize.

LEMMA 4.10. *Let $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ be the sequence generated by the CBCG method with the adaptive stepsize strategy given in (4.12). Then, for any $k \geq 0$, we have*

$$H(\mathbf{x}^k) - H(\mathbf{x}^{k+1}) \geq \frac{1}{NC_2} S(\mathbf{x}^k)^2,$$

where
(4.23)
$$C_2 = 4\left[\max_{i=1,2,\ldots,N}\left\{\max\left\{\beta_i \|A_i\|^2 D_i^2, K_i\right\}\right\} + \frac{NL_F^2 D^2 \max_{i=1,2,\ldots,N} \|A_i\|^2}{\beta_{\min}}\right],$$

and $\beta_{\min}$ is defined in (2.4) while K_i is defined in (4.17), for $i = 1, 2, \ldots, N$.

Proof. For any $i \in \{i = 1, 2, \ldots, N\}$ we have that

$$
\begin{aligned}
S_i(\mathbf{x}^k)^2 &= (S_i(\mathbf{x}^{k,i-1}) + S_i(\mathbf{x}^k) - S_i(\mathbf{x}^{k,i-1}))^2 \\
&\leq 2S_i(\mathbf{x}^{k,i-1})^2 + 2(S_i(\mathbf{x}^k) - S_i(\mathbf{x}^{k,i-1}))^2 \\
&\leq 2S_i(\mathbf{x}^{k,i-1})^2 + 2L_F^2 D_i^2 \|\mathbf{A}_i\|^2 \cdot \|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k,i-1})\|^2 \\
&\leq 2S_i(\mathbf{x}^{k,i-1})^2 + \frac{4NL_F^2 D_i^2 \|\mathbf{A}_i\|^2}{\beta_{\min}}(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})),
\end{aligned}
\tag{4.24}
$$

where the second inequality follows from Lemma 3.2 and the last inequality follows from Lemma 4.8. Invoking Lemma 4.7, we obtain from (4.24) that

$$
\begin{aligned}
S_i(\mathbf{x}^k)^2 &\leq 2S_i(\mathbf{x}^{k,i-1})^2 + \frac{4NL_F^2 D_i^2 \|\mathbf{A}_i\|^2}{\beta_{\min}}(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})) \\
&\leq 4\max\left\{\beta_i\|\mathbf{A}_i\|^2 D_i^2, K_i\right\}(H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i})) \\
&\quad + \frac{4NL_F^2 D_i^2 \|\mathbf{A}_i\|^2}{\beta_{\min}}(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})).
\end{aligned}
\tag{4.25}
$$

Summing (4.25) for $i = 1, 2, \ldots, N$ yields

$$
\begin{aligned}
\sum_{i=1}^{N} S_i(\mathbf{x}^k)^2 &\leq 4\sum_{i=1}^{N} \max\left\{\beta_i\|\mathbf{A}_i\|^2 D_i^2, K_i\right\}(H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i})) \\
&\quad + \frac{4NL_F^2}{\beta_{\min}}\sum_{i=1}^{N} D_i^2\|\mathbf{A}_i\|^2 (H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})) \\
&\leq 4 \max_{i=1,2,\ldots,N} \max\left\{\beta_i\|\mathbf{A}_i\|^2 D_i^2, K_i\right\} \sum_{i=1}^{N} (H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i})) \\
&\quad + \frac{4NL_F^2 D^2 \max_{i=1,2,\ldots,N} \|\mathbf{A}_i\|^2}{\beta_{\min}}(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})) \\
&= C_2(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})).
\end{aligned}
$$

Finally, using (3.6) we have

$$
S(\mathbf{x}^k)^2 = \left[\sum_{i=1}^{N} S_i(\mathbf{x}^k)\right]^2 \leq N\sum_{i=1}^{N} S_i(\mathbf{x}^k)^2 \leq NC_2(H(\mathbf{x}^k) - H(\mathbf{x}^{k+1})),
$$

which proves the desired result. $\quad\Box$

By combining Lemma 3.1 with Lemma 4.10, we obtain the following corollary.

COROLLARY 4.11. *Let $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ be the sequence generated by the CBCG method with the adaptive stepsize strategy given in* (4.12). *Then, for any $k \geq 0$, we have*

$$
H(\mathbf{x}^k) - H(\mathbf{x}^{k+1}) \geq \frac{1}{NC_2}(H(\mathbf{x}^k) - H^*)^2,
$$

where C_2 is given in (4.23).

In order to get a rate of convergence result in this case we will need the following technical lemma (cf. [6, Lemma 3.5]).

LEMMA 4.12. *Let $\{a_k\}_{k \in \mathbb{N}}$ be a sequence of nonnegative real numbers satisfying, for any $k \in \mathbb{N}$, the property*

$$
a_k - a_{k+1} \geq \gamma a_k^2,
\tag{4.26}
$$

and $a_0 \leq 1/(\gamma m)$ for some positive γ and m. Then, for any $k \in \mathbb{N}$, we have that

$$a_k \leq \frac{1}{\gamma} \cdot \frac{1}{m+k}.$$

Combining Lemma 4.12 with Corollary 4.11 and Lemma 3.1 establishes the sublinear rate of convergence of the CBCG method with adaptive stepsize.

THEOREM 4.13 (sublinear rate for CBCG with adaptive stepsize). *Let $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ be the sequence generated by the CBCG method with the adaptive stepsize strategy given in* (4.12). *Then for any $k \geq 0$ we have*

$$(4.27) \qquad H(\mathbf{x}^k) - H^* \leq \frac{NC_2}{k+4},$$

and

$$\min_{n=0,1,\ldots,k} S(\mathbf{x}^n) \leq \frac{2NC_2}{k+4},$$

where

$$C_2 = 4 \left[\max_{i=1,2,\ldots,N} \left\{ \max \left\{ \beta_i \|\mathbf{A}_i\|^2 D_i^2, K_i \right\} \right\} + \frac{N L_F^2 D^2 \max_{i=1,2,\ldots,N} \|\mathbf{A}_i\|^2}{\beta_{\min}} \right],$$

$\beta_{\min}$ is defined in (2.4) *and K_i is defined in* (4.17) *for $i = 1, 2, \ldots, N$.*

Proof. Denote $a_k = H(\mathbf{x}^k) - H^*$. Thanks to Corollary 4.11 we get that (4.26) holds true with $\gamma = 1/(NC_2)$. In addition, from Lemma 3.1 and (4.20), we have

$$a_0 = H(\mathbf{x}^0) - H^* \leq S(\mathbf{x}^0) = \sum_{i=1}^N S_i(\mathbf{x}^0) \leq \sum_{i=1}^N K_i \leq N \max_{i=1,2,\ldots,N} K_i \leq \frac{NC_2}{4}.$$

Hence, $a_0 \leq 1/(4\gamma)$. Picking $m = 4$, we conclude from Lemma 4.12 that (4.27) holds. To find the bound on the optimality measure $S(\cdot)$, from Lemma 4.10, we have for any $n \geq 0$

$$H(\mathbf{x}^n) - H(\mathbf{x}^{n+1}) \geq \gamma S(\mathbf{x}^n)^2.$$

For any $k_0 \geq 0$, summing the latter inequality for $n = k_0, k_0 + 1, \ldots, 2k_0$, we obtain that

$$\gamma \sum_{n=k_0}^{2k_0} S(\mathbf{x}^n)^2 \leq H(\mathbf{x}^{k_0}) - H(\mathbf{x}^{2k_0+1}) \leq H(\mathbf{x}^{k_0}) - H(\mathbf{x}^*) \leq \frac{1}{\gamma} \cdot \frac{1}{k_0 + 4},$$

where the last inequality follows from (4.27). Hence,

$$(4.28) \qquad \min_{n=k_0,k_0+1,\ldots,2k_0} S(\mathbf{x}^n)^2 \leq \frac{1}{\gamma^2} \cdot \frac{1}{(k_0+1)(k_0+4)} \leq \frac{1}{\gamma^2} \cdot \frac{1}{(k_0+2)^2},$$

where we have used the fact that $(k_0 + 1)(k_0 + 4) \geq (k_0 + 2)^2$. Similarly, for any $k_0 \geq 0$,

$$\gamma \sum_{n=k_0}^{2k_0+1} S(\mathbf{x}^n)^2 \leq H(\mathbf{x}^{k_0}) - H(\mathbf{x}^{2k_0+2}) \leq H(\mathbf{x}^{k_0}) - H(\mathbf{x}^*) \leq \frac{1}{\gamma} \cdot \frac{1}{k_0 + 4}.$$

Hence

$$(4.29) \qquad \min_{n=k_0,k_0+1,\ldots,2k_0+1} S(\mathbf{x}^n)^2 \leq \frac{1}{\gamma^2} \cdot \frac{1}{(k_0+2)(k_0+4)} \leq \frac{1}{\gamma^2} \cdot \frac{1}{(k_0+5/2)^2},$$

where we have used the fact that $(k_0+2)(k_0+4) \geq (k_0+5/2)^2$. Notice that $k_0+5/2 = (2k_0+1)/2+2$. Therefore, by combining (4.28) when k is even and (4.29) when k is odd, we conclude that

$$\min_{n=0,1,\ldots,k} S(\mathbf{x}^n) \leq \frac{1}{\gamma} \cdot \frac{1}{k/2+2} \leq \frac{2NC_2}{k+4},$$

which concludes the proof. $\qquad \square$

4.3. Backtracking version of CBCG. Computing the adaptive stepsize requires knowing the constants β_i, $i = 1, 2, \ldots, N$. In practice, these constants may not be known in advance or their known upper-approximations may be too loose. A common approach to overcome this problem is to use a backtracking scheme in order to estimate the unknown constants that are required to ensure convergence of the algorithm. This strategy is also effective in the context of CG-type algorithms.

The crucial property for the convergence analysis with adaptive stepsize is the block structured descent condition given in Lemma 4.6. When the constants in Lemma 4.6 are not known, a workaround is to enforce this descent condition algorithmically, which leads to the following scheme.

CBCG-B: Cyclic Block Conditional Gradient with Backtracking
Initialization. $\mathbf{x}^0 \in X$, $\beta_{\text{init}} > 0$, $\kappa > 1$, $\xi_i^{-1} = 1$, $i = 1, 2, \ldots, N$.
General step. For $k = 1, 2, \ldots$,
    (i) For any $i = 1, 2, \ldots, N$, compute

$$(4.30) \qquad \mathbf{p}_i^k \in \operatorname{argmin}_{\mathbf{p}_i \in X_i} \left\{ \langle \nabla_i f(\mathbf{x}^{k,i-1}), \mathbf{p}_i \rangle + g_i(\mathbf{p}_i) \right\}.$$

    Find $\xi \in \mathbb{N}$, which is the smallest integer $\xi \geq \xi_i^{k-1}$ such that

$$(4.31) \qquad H\left(\mathbf{x}^{k,i-1}\right) - H\left(\mathbf{x}^{k,i-1} + \alpha \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\right) \geq \frac{\alpha}{2} S_i(\mathbf{x}^{k,i-1}),$$

    where $\alpha = \min \left\{ \frac{S_i(\mathbf{x}^{k,i-1})}{\kappa^\xi \beta_{\text{init}} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\|^2}, 1 \right\}$ and set

$$(4.32) \quad \xi_i^k = \xi, \quad \beta_i^k = \kappa^\xi \beta_{\text{init}}, \quad \alpha_i^k = \min \left\{ \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i^k \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})\|^2}, 1 \right\},$$

$$(4.33) \quad \mathbf{x}^{k,i} = \mathbf{x}^{k,i-1} + \alpha_i^k \mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1}).$$

    (ii) Update: $\mathbf{x}^{k+1} = \mathbf{x}^{k,N}$.

Under Assumption 2, we have from Lemma 4.6 that (4.31) holds provided that $\beta_i^k \geq \beta_i$. We therefore have for any $k \geq 0$ and any $i = 1, 2, \ldots, N$ that

$$(4.34) \qquad \beta_{init} \leq \beta_i^k \leq \bar{\beta}_i \equiv \max\left\{ \kappa \beta_i, \beta_{init} \right\}.$$

The main insight is given in the following lemma and its proof follows the same arguments as that of Lemma 4.6 using (4.31).

LEMMA 4.14. *Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ *be the sequence generated by the CBCG-B method. Then, for any* $k \geq 0$ *and* $i \in \{1, 2, \ldots, N\}$, *we have*

$$H(\mathbf{x}^{k,i-1}) - H(\mathbf{x}^{k,i}) \geq \frac{\alpha_i^k}{2} S_i(\mathbf{x}^{k,i-1}) \geq \frac{(\alpha_i^k)^2 \beta_i^k}{2} \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2,$$

where

$$\alpha_i^k = \min\left\{ \frac{S_i(\mathbf{x}^{k,i-1})}{\beta_i^k \|\mathbf{A}_i(\mathbf{p}_i^k - \mathbf{x}_i^k)\|^2}, 1 \right\}.$$

Using the bounds in (4.34), we obtain the following rate of convergence result for the CBCG algorithm with a backtracking scheme.

THEOREM 4.15 (sublinear rate for CBCG-B). *Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ *be the sequence generated by the CBCG-B method. Then, for any* $k \geq 0$, *we have*

$$H(\mathbf{x}^k) - H^* \leq \frac{NC_3}{k+4},$$

and

$$\min_{n=0,1,\ldots,k} S(\mathbf{x}^n) \leq \frac{2NC_3}{k+4},$$

where

$$C_3 = 4\left[\max_{i=1,2,\ldots,N} \max\left\{ \bar{\beta}_i \|\mathbf{A}_i\|^2 D_i^2, K_i \right\} + \frac{NL_F^2 D^2 \max_{i=1,2,\ldots,N} \|\mathbf{A}_i\|^2}{\beta_{init}} \right],$$

and $\bar{\beta}_i$ *is given in* (4.34) *while* K_i *is defined in* (4.17), *for* $i = 1, 2, \ldots, N$

Proof. The arguments of this proof are similar to those in the proof of Theorem 4.13 where we replace Lemma 4.6 with Lemma 4.14. $\square$

5. Further discussions. In this section we begin by showing that the complexity analysis presented in section 4.2 holds for the CBCG method with exact line search when the smooth part of the objective function is quadratic. Then, we discuss two issues that are relevant for the implications of our analysis of the CBCG method.

5.1. Exact line search for quadratic problems. We recall another well-known stepsize rule—exact line search strategy (see, for example, [12]). This stepsize is defined as follows:

(5.1) $$\alpha_i^k \in \operatorname{argmin}_{\alpha\in[0,1]} H(\mathbf{x}^{k,i-1} + \alpha\mathbf{U}_i(\mathbf{p}_i^k - \mathbf{x}_i^{k,i-1})).$$

The minimization step (5.1) can incur unnecessary additional computations, unless it can be carried out efficiently. This is the case for quadratic functions for which it reduces to the minimization of a one-dimensional quadratic over a segment. It is therefore tempting to use CBCG with this stepsize rule, but our convergence analysis does not cover it explicitly. However, in the quadratic case, exact line search can be recast as an adaptive stepsize strategy. Indeed, suppose, for example, that the problem has the following form:

$$\min\left\{ \frac{1}{2}\left\|\sum_{i=1}^{N} \mathbf{A}_i\mathbf{x}_i\right\|^2 + \sum_{i=1}^{n}\langle\mathbf{b}_i, \mathbf{x}_i\rangle + \sum_{i=1}^{n}\tilde{g}_i(\mathbf{x}_i) \right\},$$

where $\mathbf{A}_i \in \mathbb{R}^{n \times n_i}$, $\mathbf{b}_i \in \mathbb{R}^{n_i}$, and $\tilde{g}_i$ satisfy Assumption 1(i). This problem fits model (2.1) with $F(\cdot) = \frac{1}{2}\|\cdot\|^2$, $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N)$, and $g_i(\cdot) = \langle \mathbf{b}_i, \cdot \rangle + \tilde{g}_i(\cdot)$. Therefore, we can choose $\beta_i = 1$, $i = 1, 2, \ldots, N$, and the exact line search strategy is equivalent to our adaptive stepsize strategy given in (4.12) since the upper bound of Lemma 4.1 holds with equality. Therefore, in the quadratic case, convergence for the exact line search strategy follows from Theorem 4.13.

5.2. Random permutations. All of the arguments presented in section 4 remain valid if the order of the blocks is not fixed for all iterations. The only important element is that all blocks are visited once at each iteration, but the order could change, in an arbitrary way, at each iteration. In particular, the arguments are valid when the order of blocks is picked as a random permutation at the beginning of each iteration. Therefore, all of the convergence rate estimates presented in section 4 still hold true when using random permutations.

The practical motivation for considering random permutations in the order of the blocks is that it may be beneficial in practical settings [29]. Since the derived theoretical efficiency estimates apply to any rule for choosing the order of the blocks at the beginning of each iteration (deterministic or random), they actually do not explain the potential performance differences between different variants, such as purely cyclic versus random permutations.

5.3. Implications for coordinate descent in machine learning. In the case of blocks of size one (i.e., $N = n$), the CBCG method bears a lot of resemblance to coordinate descent type methods. For example, in dimension one, a conditional gradient step with exact line search is the same as exact minimization. Therefore, CBCG with blocks of size one and exact line search is equivalent to coordinate descent with exact minimization. As we have seen in the previous section, our convergence analysis holds for this setting in the case of quadratic objectives. This setting has some interesting applications.

The ℓ_2 regularized empirical risk minimization is considered in [29], where the authors analyze the stochastic coordinate descent method with exact minimization in the dual—a method referred to as Stochastic Dual Coordinate Ascent (SDCA). The dual problem can be written as follows (using our notation):

$$(5.2) \qquad \min_{\mathbf{x}} \left\{ \frac{1}{n} \sum_{i=1}^{n} g_i(\mathbf{x}_i) + \frac{\lambda}{2} \left\| \frac{1}{\lambda n} \sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i \right\|^2 \right\},$$

where $\lambda > 0$ is a parameter, and for any $i = 1, 2, \ldots, n$, $\mathbf{x}_i \in \mathbb{R}$, $\mathbf{A}_i \in \mathbb{R}^n$, and g_i is a closed, convex, and proper function with a bounded domain. This is exactly the setting of section 5.1, and therefore Theorem 4.13 gives explicit deterministic rate estimates of the duality gap for the cyclic and random permutation coordinate descent variants applied to problem (5.2) when $g_1, g_2, \ldots, g_n$ satisfy Assumption 1. It can be checked that in the setting of problem (5.2) (with $\lambda = 1$ for simplicity), we get in Theorem 4.13 that as n grows, the dominant term in the rate estimate (4.27) is of the form $4D^2 \max_i \|\mathbf{A}_i\|^2/(k+4)$. The quantity D^2 remains proportional to n (see identity (2.3)) and the rate estimate of (4.27) still suffers from a multiplicative dependence in n, since each iteration requires an effective pass through the n coordinates. Therefore, these results remain mostly of theoretical interest in the context of coordinate descent in machine learning where the focus is on large n and there is space for improvement in order to theoretically grasp the practical performances of these types of methods.

6. Numerical experiments. The experiments presented in this section correspond to situations where for each i, the nonsmooth function g_i is taken to be the indicator of the set $X_i \subseteq \mathbb{R}^{n_i}$, with the eventual addition of a linear term. We therefore recover the smooth constrained optimization model of the traditional conditional gradient method (see also Remark 2.1). We begin with a modeling remark in section 5.1. Numerical results are given in section 6.1 for synthetic examples and in section 6.2 for the structured SVM training problem.

6.1. Synthetic problems. We generate artificial convex quadratic problems with box constraints in $\mathbb{R}^{100}$. We consider problems of the form

$$(6.1) \qquad \min_{\|\mathbf{x}\|_\infty \leq 1} \frac{1}{2}(\mathbf{x} - \mathbf{y})^T \mathbf{Q}(\mathbf{x} - \mathbf{y}),$$

where $\|\cdot\|_\infty$ denotes the ℓ_∞ norm, $\mathbf{y} \in \mathbb{R}^{100}$, and $\mathbf{Q} \in \mathbb{R}^{100 \times 100}$ are given. The problem has a natural coordinatewise structure which allows us to apply the CBCG algorithm where each block consists of a single coordinate. Indeed, by setting $f(\cdot) = \frac{1}{2}(\cdot - \mathbf{y})^T \mathbf{Q}(\cdot - \mathbf{y})$ and $g_i = \delta_{[-1,1]}$, $i = 1, 2, \ldots, 100$, problem (6.1) fits our model (2.1). We generate random instances of problem (6.1) as follows:

- Set $d = 100$ and $n = 200$.
- Generate $\mathbf{X} \in \mathbb{R}^{n \times d}$ with standard normal entries.
- Set $\mathbf{Q} = \frac{1}{n}\mathbf{X}^T \mathbf{D}^2 \mathbf{X}$, where $\mathbf{D} = \mathrm{diag}(1/n^2, 1/(n-1)^2, \ldots, 1)$.
- Generate $\mathbf{y}$ with standard normal entries.

We compare the standard Conditional Gradient (CG) algorithm [17], the Random Block Conditional Gradient (RBCG) algorithm [19] (with block chosen uniformly at random), the Cyclic Block Conditional Gradient where the order of the blocks is fixed for all iterations (CBCG-C), and the Cyclic Block Conditional Gradient where the order of the blocks is chosen by a random permutation at each iteration (CBCG-P). For each algorithm, we compare three different stepsize rules:

- **Predefined:** the stepsize which is given in (4.3). Note that for RBCG, there is a slight modification in the definition of the predefined stepsize. There is no notion of a "cycle" in RBCG and the stepsize has the form $2n/(\tilde{k} + 2n)$ where $\tilde{k}$ is the number of blocks updated so far; see [19].
- **Adaptive with backtracking:** set $\mathbf{A} = \mathbf{I}$ and compute the stepsize using the backtracking scheme.
- **Exact line search:** the stepsize is chosen to minimize the objective as in (5.1). Note that as explained in section 5.1, this corresponds to using the adaptive stepsize (4.12) with $\mathbf{A} = \mathbf{DX}/\sqrt{n}$ since we consider a quadratic objective function.

For problems of the form (6.1), the second strategy is questionable since a closed form expression is available for the exact line search. However, the comparison to this strategy illustrates differences in algorithmic behaviors related to the choice of $\mathbf{A}$ in the model (2.1) as well as the performance of the backtracking scheme.

Remark 6.1. For CG and RBCG, a sublinear convergence rate holds for the three stepsize strategies. For both CBCG-C and CBCG-P, in the case of the predefined and of the adaptive stepsizes, convergence is ensured by Theorems 4.5 and 4.15, respectively. Furthermore, since we consider quadratic objective functions, exact line search stepsize strategy can be seen as a particular case of our adaptive strategy and the convergence follows from Theorem 4.13 (see also section 5.1).

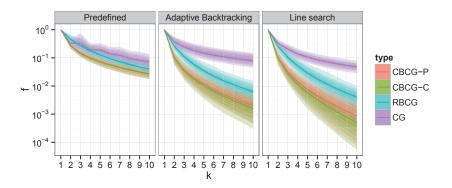We generate 1000 random instances of problem (6.1), f_w denoting the objective

FIG. 1. *Comparison of Conditional Gradient (CG), its random block version (RBDG), its cyclic block version with fixed block order (CBCG-C), and its cyclic block version with random permutation block order (CBCG-P). We compare three different stepsize strategies based on* 1000 *randomly generated instances of problem* (6.1)*. The central line is the median over the* 1000 *runs and the ribbons show* 98%*,* 90%*,* 80%*,* 60%*, and* 40% *quantiles. For all methods, k represents the number of effective passes through d coordinates.*

function of the wth randomly generated problem, $w = 1, 2, \ldots, 1000$. For each problem, we run the four different algorithms with the three different stepsize rules (the initialization is at the origin). The results for the first ten iterations are presented in Figure 1. For each algorithm, increasing k by 1 means that $N = 100$ blocks have been queried randomly for RBCG, sequentially for CBCG, and all at once for CG. Since each objective function is generated randomly, it does not make sense to directly compare performance across different problems on the same graph. To overcome this, for each $w = 1, 2, \ldots, 1000$, we "center" and "scale" the function values. That is, for each objective f_w, $w = 1, 2, \ldots, 1000$, we estimate the optimal value f_w^* of (6.1) by running CBCG-P with exact line search for 200 more iterations. For such a number of iterations, we observed on preliminary experiments that the algorithm reached machine precision on random instances of problem (6.1). The quantity plotted in Figure 1 is given by the affine transformation

$$\frac{f_w(\mathbf{x}^k) - f_w^*}{f_w(\mathbf{0}) - f_w^*},$$

so that in Figure 1, the first value is always 1 and the represented quantities are positive and asymptotically tend to 0. The main comments regarding this experiment are the following:

- For each stepsize rule, CBCG-C has an advantage.
- There is not much difference between the two variants CBCG-C and CBCG-P in this experiment.
- The predefined stepsize rule leads to much slower convergence.
- Adaptive with backtracking and exact line search rules yield improved convergence speed for both CBCG and RBCG, which is not the case with the CG.
- Exact line search rule has a slight advantage over the backtracking rule.

The last point deserves further comments. Indeed, in model (2.1), there are different possible choices of matrix $\mathbf{A}$ and function F that lead to equivalent problems. Despite

being equivalent problems, different model choices may lead to variations in algorithmic performances when numerically solving a problem. This is what we observe here. Indeed, as pointed out in section 5.1, the exact line search strategy corresponds to choosing an F that is perfectly conditioned. This means that variations of F around its minimum are isotropic. On the other hand, choosing $\mathbf{A} = \mathbf{I}$ leads to a choice of F that is less well-behaved. These numerical experiments suggest that choosing $\mathbf{A}$ that corresponds to a better conditioned F leads to better numerical performances. Finally, the gap between the two strategies remains small, highlighting the efficiency of the backtracking scheme.

6.2. Structured SVM. The main motivation for the introduction of block version of CG method with random update rule in [19] is that it leads to a new efficient algorithm for training the structured SVM [30, 31]. We refer the reader to [19] and the references therein for background on this problem and its relations to the conditional gradient method. In brief, the structured SVM solves a multiclass classification task. It is dedicated to problems for which the output classes are embedded in a combinatorial structure such as trees, graphs, or sequences. In this setting the number of classes can be enormous, which results in optimization problems with an untractable number of linear constraints. For some of these problems efficient decoding algorithms can be used as oracles to compute subgradients of the structured SVM problem. They can also be used as oracles to solve the linearized subproblem required to run the conditional gradient and block conditional gradient methods on the dual of the structured SVM. In this section, we briefly recall the mathematical formulation of the dual structured SVM problem and provide a numerical comparison between random and cyclic update rules for block conditional gradient in this context. The purpose is not to be exhaustive here and we focus solely on the aspects of the problem related to optimization. Therefore, we compare the numerical performances of the two selection rules on this real-world example based on an optimization criterion. In this section, N denotes the number of training examples, M denotes the number of output classes, and d is an integer such that $\mathbb{R}^d$ is a space of tractable size (such that elements of $\mathbb{R}^d$ can be stored in memory). The dual variable of the structured SVM is a matrix $\boldsymbol{\alpha} \in \mathbb{R}^{N \times M}$ with nonnegative entries (this could actually be refined with example dependent output classes, but we stick to this notation as a first approximation). The dual problem of the structured SVM can be written as follows:

$$(6.2) \qquad \min_{\alpha \geq \mathbf{0}} \left\{ \frac{\lambda}{2} \left\| \mathbf{A}\boldsymbol{\alpha} \right\|^2 - \mathrm{Tr}(\mathbf{b}^T \boldsymbol{\alpha}) : \boldsymbol{\alpha}1^M = 1^N \right\},$$

where $\mathbf{A} : \mathbb{R}^{N \times M} \to \mathbb{R}^d$ is a linear map, $\mathbf{b} \in \mathbb{R}^{N \times M}$ is a matrix, Tr denotes the trace operator, and 1^s denotes the vector in $\mathbb{R}^s$ in which all entries are 1. Problem (6.2) has an interesting product structure. Indeed, its feasible set can be viewed as a product of N simplices of dimension M. In the context of structured SVM it is not necessary to store $\boldsymbol{\alpha}$ explicitly, it is sufficient to store and update $\mathbf{A}\boldsymbol{\alpha} \in \mathbb{R}^d$ and $\mathrm{Tr}(\mathbf{b}^T \boldsymbol{\alpha})$. With this information, we can use the specific decoding oracles to solve simplex constrained linear subproblems and run the conditional gradient algorithm. This constitutes the main advantage of the method here, it allows us to explore a potentially very large space by using only implicit conditional gradient steps (recall that M is the size of a set of combinatorial nature; see [19] for a complete derivation and more details). We use the code provided by the authors of [19] which gives the possibility to train the structured SVM using RBCG, CBCG-C (fixed block order), or CBCG-P (random permutation block order) on the Optical Character Recognition
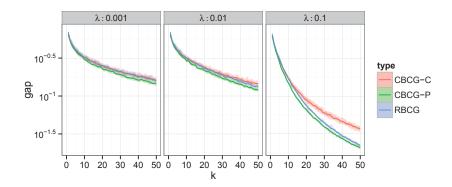
FIG. 2. *Comparison of RBCG, CBCG-C (purely cyclic), and CBCG-P (random permutation) for the structured SVM training on the OCR dataset of* [30]. *The quantity represented is the optimality measure defined by* (3.1) *and the heading represents the value of* λ *(from* (6.2)*). The central line is the median over* 20 *runs and the ribbons show* 80% *and* 50% *quantiles. For all methods, k represents the number of effective passes through all the blocks (which correspond to datapoints here).*

(OCR) originally proposed in [30]. We only consider the exact line search strategy (5.1). Note that our convergence analysis can be applied in this setting (see also section 5.1). It can be checked (see [19] for details about the structured SVM model) that the constant C_2 given by Theorem 4.13 remains bounded away from zero as N grows, in the setting of model (6.2). Therefore, the rate given in (4.27) suffers from a multiplicative dependence in N. From a theoretical point of view, in the context of large training sets (large N), this problematic dependence leaves room for improved convergence analysis. Numerical results in terms of the optimality measure S for various values of λ are given in Figure 2. The global behavior differs from what we have observed in the synthetic examples of section 6.1. The random block selection rule (RBCG) has an advantage over the purely cyclic block selection rule (CBCG-C). The random permutation approach (CBCG-P) gives the best performances here. Since our analysis is the same for both CBCG-C and CBCG-P, it does not allow us to explain this empirical difference.

7. Discussion and future work.

7.1. Comparison with existing results. We have described the CBCG algorithm and provided an explicit sublinear convergence rate estimate of the form $O(1/k)$, which is canonical for linear oracle based algorithms (up to multiplicative factor) [26]. This asymptotic rate cannot be improved in general [10]. A few words on the constants are in order. For the traditional CG method, the multiplicative constant is proportional to $L_F D^2$ [17].[2] In order to compare these rates with those obtained for block decomposition methods, we consider in this discussion that one iteration of a block decomposition method consists of updating N arbitrarily chosen blocks (which is consistent with the notation in this paper). The multiplicative constant proposed in [19] for the average case complexity of RBCG is proportional to $(L_F D^2 + H(\mathbf{x}^0) - H^*)$

---

[2]The constant presented in [17] relates to an affine invariant notion of curvature, but it can be upper bounded by the term we consider.

independently of the number of blocks.[3] The independence of the constant with respect to the number of blocks is an important property that we do not recover for the cyclic block updating rule. This behavior is consistent with previous observations comparing random and cyclic update rules [6] where multiplicative dependance in the number of blocks also appeared in the convergence rate analysis of different types of methods.

For the sake of clarity, we will discuss the results of section 4 when $\mathbf{A} = \mathbf{I}$, all D_i are equal, and $\beta_i = L_F$, $i = 1, 2, \ldots, N$. For the predefined stepsize, the quantity $C_1/L_F D^2$ (C_1 given by (4.4)) grows like $\Omega(\sqrt{N})$. Furthermore, for the adaptive stepsize, the quantity $NC_2/L_F D^2$ (C_2 given by (4.23)) grows like $\Omega(N^2)$. This is much worse than the $\sqrt{N}$ dependance of the predefined strategy although practical simulations tend to show that the adaptive rule is much faster. Therefore, there is room for improvement in the analysis, and this raises the natural question of the possibility to get multiplicative constants that do not depend on the number of blocks (which is, in general, the case for random block selection rules). As we have seen in sections 5 and 6, for SDCA and structured SVM applications, the effect is to have a multiplicative dependance of the convergence rate in the size of the training set. Therefore, although theoretically interesting, the rates we derive are limited to explain performances on large training sets (large number of blocks in the dual) which is the motivation for using block decomposition methods in machine learning applications.

For practical applications, however, this remark should be mitigated since we are comparing upper bounds. These bounds may only reflect limitations of the analysis, not of the methods, and their comparison may not shed much light on the comparative behavior of different rules on practical problems. Indeed, our numerical experiments on synthetic and real-world examples reproducibly suggest an advantage of cyclic or random permutation variants over fully random update rule. This is again something that has already been observed in other contexts [6, 29]. A question related to the discussion of the previous paragraph is to give a theoretical justification to this observation or eventually provide different iteration dependent block update rules that comply with it (see, for example, [25] for an illustration in the context of gradient descent).

7.2. Future directions. Finally, a natural question is that of the extension of specificities of the conditional gradient method in our block decomposition setting. Potential directions include the following:

- **Linear convergence.** CG is known to converge linearly when the optimum is in the relative interior of the feasible set [15, 20] or when the feasible set is strongly convex and the gradient of the smooth part of the objective is nonzero on the feasible set [21].
- **Dual interpretation of the block decomposition.** Generalized CG is known to implicitly generate subgradient sequences [1] related to the mirror descent algorithm [3] applied to a dual problem. Similarly, a dual interpretation of RBCG is in terms of stochastic subgradient [19].
- **Generalization of the results to exact line search stepsize strategies.** The analysis of such stepsize strategies is not problematic for CG or RBCG [17, 19]. However, we could not generalize it for CBCG, except in the quadratic case (see section 5.1), and thus developing an analysis for the exact

---

[3]As in the case of CG, the result is not presented with this exact constant, but it is an upper bound that we use for discussion purposes.

line search strategy is in our opinion an important task. Most of our analysis collapses without further assumptions because it is no longer possible to get sufficient decrease conditions of the type of Lemma 4.8. We therefore expect that a different path needs to be considered.

- **Generalization to inexact oracles.** In many practical applications, the search direction given by the oracle is computed by an algorithm. It is therefore relevant to consider multiplicative errors in order to use well-defined stopping criteria for the inner algorithm. This was already considered in the random block variant [19].

Appendix. Proof of Lemma 2.2. We adapt the standard proof of the descent Lemma; see, e.g., [7, Proposition A.24]. Under the assumptions of the lemma, using the fundamental theorem of calculus for line integrals on the segment $[\mathbf{x}, \mathbf{x} + \mathbf{U}_i \mathbf{h}]$, we have

$$f(\mathbf{x} + \mathbf{U}_i \mathbf{h}) = f(\mathbf{x}) + \int_0^1 \langle \nabla f(\mathbf{x} + t\mathbf{U}_i \mathbf{h}), \mathbf{U}_i \mathbf{h} \rangle \, dt$$

$$\text{(A.1)} \qquad = f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{U}_i \mathbf{h} \rangle + \int_0^1 \langle \nabla f(\mathbf{x} + t\mathbf{U}_i \mathbf{h}) - \nabla f(\mathbf{x}), \mathbf{U}_i \mathbf{h} \rangle \, dt.$$

We can bound the integrand term for any $t \in [0, 1]$ as follows:

$$\langle \nabla f(\mathbf{x} + t\mathbf{U}_i \mathbf{h}) - \nabla f(\mathbf{x}), \mathbf{U}_i \mathbf{h} \rangle = \langle \mathbf{A}^T \nabla F(\mathbf{A}(\mathbf{x} + t\mathbf{U}_i \mathbf{h})) - \mathbf{A}^T \nabla F(\mathbf{A}\mathbf{x}), \mathbf{U}_i \mathbf{h} \rangle$$

$$= \langle \nabla F(\mathbf{A}(\mathbf{x} + t\mathbf{U}_i \mathbf{h})) - \nabla F(\mathbf{A}\mathbf{x}), \mathbf{A}\mathbf{U}_i \mathbf{h} \rangle$$

$$= \langle \nabla F(\mathbf{A}\mathbf{x} + t\mathbf{A}_i \mathbf{h}) - \nabla F(\mathbf{A}\mathbf{x}), \mathbf{A}_i \mathbf{h} \rangle$$

$$\leq \|\nabla F(\mathbf{A}\mathbf{x} + t\mathbf{A}_i \mathbf{h})) - \nabla F(\mathbf{A}\mathbf{x})\| \cdot \|\mathbf{A}_i \mathbf{h}\|$$

$$\text{(A.2)} \qquad\qquad \leq t\beta_i \|\mathbf{A}_i \mathbf{h}\|^2,$$

where we have used Cauchy–Schwarz inequality and Assumption 2 to obtain the last two inequalities. Combining (A.1) and (A.2), we have

$$f(\mathbf{x} + \mathbf{U}_i \mathbf{h}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{U}_i \mathbf{h} \rangle + \beta_i \|\mathbf{A}_i \mathbf{h}\|^2 \int_0^1 t \, dt$$

$$= f(\mathbf{x}) + \langle \nabla_i f(\mathbf{x}), \mathbf{h} \rangle + \frac{\beta_i \|\mathbf{A}_i \mathbf{h}\|^2}{2},$$

which proves the desired result. $\quad\square$

REFERENCES

[1] F. BACH, *Duality between subgradient and conditional gradient methods*, SIAM J. Optim., 25 (2015), pp. 115–129.

[2] A. BECK, *On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes*, SIAM J. Optim., 25 (2015), pp. 185–209.

[3] A. BECK AND M. TEBOULLE, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Oper. Res. Lett., 31 (2003), pp. 167–175.

[4] A. BECK AND M. TEBOULLE, *A conditional gradient method with linear rate of convergence for solving convex linear systems*, Math. Methods Oper. Res., 59 (2004), pp. 235–247.

[5] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[6] A. BECK AND L. TETRUASHVILI, *On the convergence of block coordinate descent type methods*, SIAM J. Optim., 23 (2013), pp. 2037–2060.

[7] D. P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[8] K. BREDIES AND D. A. LORENZ, *Iterated hard shrinkage for minimization problems with sparsity constraints*, SIAM J. Sci. Comput., 30 (2008), pp. 657–683.

[9] K. BREDIES, D. A. LORENZ, AND P. MAASS, *A generalized conditional gradient method and its connection to an iterative shrinkage method*, Comput. Optim. Appl., 42 (2009), pp. 173–193.

[10] M. D. CANON AND C. D. CULLUM, *A tight upper bound on the rate of convergence of Frank-Wolfe algorithm*, SIAM J. Control, 6 (1968), pp. 509–516.

[11] B. COX, A. JUDITSKY, AND A. NEMIROVSKI, *Dual subgradient algorithms for large-scale nonsmooth learning problems*, Math. Program., 148 (2014), pp. 143–180.

[12] V. F. DEM'YANOV AND A. M. RUBINOV, *The minimization of a smooth convex functional on a convex set*, SIAM J. Control, 5 (1967), pp. 280–294.

[13] J. C. DUNN AND S. HARSHBARGER, *Conditional gradient algorithms with open loop step size rules*, J. Math. Anal. Appl., 62 (1978), pp. 432–444.

[14] M. FRANK AND P. WOLFE, *An algorithm for quadratic programming*, Naval Res. Logist. Quart., 3 (1956), pp. 95–110.

[15] J. GUELAT AND P. MARCOTTE, *Some comments on Wolfe's 'away step'*, Math. Programming, 35 (1986), pp. 110–119.

[16] Z. HARCHAOUI, A. JUDITSKY, AND A. NEMIROVSKI, *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Math. Program., 152 (2015), pp. 75–112.

[17] M. JAGGI, *Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 28 (2013), pp. 427–435.

[18] M. JAGGI AND M. SULOVSKÝ, *A simple algorithm for nuclear norm regularized problems*, in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 471–478.

[19] S. LACOSTE-JULIEN, M. JAGGI, M. SCHMIDT, AND P. PLETSCHER, *Block-coordinate Frank-Wolfe optimization for structural svms*, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 28 (2013), pp. 53–61.

[20] S. LACOSTE-JULIEN AND J. MARTIN, *An Affine Invariant Linear Convergence Analysis for Frank–Wolfe Algorithms*, preprint, arXiv:1312.7864, 2013.

[21] E. S. LEVITIN AND B. T. POLJAK, *Minimization methods in the presence of constraints*, USSR Comput. Math. and Math. Phys., 6 (1966), pp. 787–823.

[22] Z. Q. LUO AND P. TSENG, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[23] YU. NESTEROV, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.

[24] YU. NESTEROV, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.

[25] J. NUTINI, M. SCHMIDT, I. H. LARADJI, M. FRIEDLANDER, AND H. KOEPKE, *Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection*, preprint, arXiv:1506.00552, 2015. ICML 2015, to appear.

[26] B. T. POLYAK, *Introduction to Optimization*, Transl. Ser. Math. Engrg., Optimization Software Inc., Publications Division, New York, 1987. Translated from the Russian.

[27] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program., 144 (2014), pp. 1–38.

[28] A. SAHA AND A. TEWARI, *On the nonasymptotic convergence of cyclic coordinate descent methods*, SIAM J. Optim., 23 (2013), pp. 576–601.

[29] S. SHALEV-SHWARTZ AND T. ZHANG, *Stochastic dual coordinate ascent methods for regularized loss minimization*, J. Mach. Learn. Res., 14 (2013), pp. 567–599.

[30] B. TASKAR, C. GUESTRIN, AND D. KOLLER, *Max-margin Markov networks*, in Advances in Neural Information Processing Systems (NIPS 2003), 2004, pp. 25–32.

[31] I. TSOCHANTARIDIS, T. JOACHIMS, T. HOFMANN, AND Y. ALTUN, *Large margin methods for structured and interdependent output variables*, J. Mach. Learn. Res., 6 (2005), pp. 1453–1484.