Block cubic Newton with greedy selection

Andrea Cristofari[*]

[*]Department of Civil Engineering and Computer Science Engineering
University of Rome “Tor Vergata”
Via del Politecnico, 1, 00133 Rome, Italy
E-mail: andrea.cristofari@uniroma2.it

Abstract. A second-order block coordinate descent method is proposed for the unconstrained minimization of an objective function with Lipschitz continuous Hessian. At each iteration, a block of variables is selected by means of a greedy (Gauss-Southwell) rule which considers the amount of first-order stationarity violation, then an approximate minimizer of a cubic model is computed for the block update. In the proposed scheme, blocks are not required to have a prefixed structure and their size is allowed to change during the iterations. For non-convex objective functions, global convergence to stationary points is proved and a worst-case iteration complexity analysis is provided. In particular, given a tolerance ε, we show that at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations are needed to drive the stationarity violation with respect to the selected block of variables below ε, while at most $\mathcal{O}(\varepsilon^{-2})$ iterations are needed to drive the stationarity violation with respect to all variables below ε. Numerical results are finally provided.

Keywords. Block coordinate descent. Cubic Newton methods. Second-order methods. Worst-case iteration complexity.

1 Introduction

Many challenging problems require the minimization of an objective function with several variables. In this respect, block coordinate descent methods often represent an advantageous approach, especially when the objective function has a nice structure, since these methods update a block of variables at each iteration and might have a low per-iteration cost. In the literature, block coordinate descent methods have been extensively analyzed in several forms, employing different rules to choose and update the blocks (see, e.g., [26, 32]).

Most block coordinate descent methods use first-order information and gained great popularity as they guarantee high efficiency in several applications. When the objective function is twice continuously differentiable, second-order information can be conveniently used as well, in order to speed up the convergence of the algorithm and overcome some drawbacks connected with first-oder methods, such as the performance deterioration in ill-conditioned or highly non-separable problems [17]. Of course, second-order information should be used judiciously in a block coordinate descent scheme, so as not to increase the per-iteration cost excessively.

Among second-order methods, a common approach in the literature is represented by *cubic Newton methods* [7, 8, 14, 15, 19, 20, 24], where, at each iteration, the next point is obtained by minimizing a cubic model, that is, a second-order model with cubic regularization. This class of algorithms requires $\mathcal{O}(\varepsilon^{-3/2})$ iterations to drive the norm of the gradient of the objective function below a given threshold ε, thus improving the bounds obtained for the steepest descent method [24]. Extensions to higher order models have also been provided when the objective function is several times continuously differentiable [5, 9].

In recent years, block coordinate descent versions of cubic Newton methods have been proposed in the literature using different block selection rules. In particular, cyclic block selection was considered in [1] for high order models that include cubic models as a special case, whereas random block selection was analyzed in [12, 21] and [33] for convex and non-convex objective functions, respectively.

Here, still considering a block coordinate descent version of cubic Newton methods, we focus on the use of a *greedy selection rule*. Under Lipschitz continuity of the Hessian of the objective function, we provide the following worst-case iteration complexity bounds for non-convex objective functions:

- at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations are needed to drive the stationarity violation with respect to *the selected block of variables* below ε,
- at most $\mathcal{O}(\varepsilon^{-2})$ iterations are needed to drive the stationarity violation with respect to *all variables* below ε.

Our results are appealing if compared with those given in [1] for cyclic block selection when using cubic models. Specifically, the former complexity bound of $\mathcal{O}(\varepsilon^{-3/2})$ was obtained in [1] as well, but note that the latter complexity bound of $\mathcal{O}(\varepsilon^{-2})$ improves over the one given in [1], which is of $\mathcal{O}(\varepsilon^{-3})$. So, according to current results established in the literature, the proposed greedy selection seems to be able to provide better complexity bounds than a cyclic selection when using cubic models.

Let us remark that, for the proposed method, we do not need to know the Lipschitz constant of the Hessian of the objective function. Moreover, we use inexact minimizers of the cubic model whose computation does not require additional evaluations of the objective function or its derivatives. Due to the block structure and the use of inexact information, we name the proposed algorithm *Inexact Block Cubic Newton* (IBCN) method.

The rest of the paper is organized as follows. In Section 2, we introduce the problem and give preliminary results. In Section 3, we describe the proposed method. In Section 4, we carry out the convergence analysis and give worst-case iteration complexity bounds. In Section 5, we show some numerical results. Finally, we draw some conclusions in Section 6.

2 Preliminaries and notations

We consider the following unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a (possibly non-convex) objective function. We assume that the Hessian matrix $\nabla^2 f(x)$ is Lipschitz continuous over $\mathbb{R}^n$ with constant $L > 0$, that is,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le L\|x - y\| \quad \forall x, y \in \mathbb{R}^n,$$

where, here and in the rest of the paper, $\|v\|$ is the Euclidean norm for any vector v, whereas $\|A\|$ is the norm induced by the vector Euclidean norm for any matrix A. The sup-norm of a vector v is indicated by $\|v\|_\infty$.

Given $\mathcal{I} \subseteq \{1, \dots, n\}$, we denote by $U_{\mathcal{I}} \in \mathbb{R}^{n\times|\mathcal{I}|}$ the submatrix of the n-dimensional identity matrix obtained by removing all columns with indices not belonging to $\mathcal{I}$. Then, given $x \in \mathbb{R}^n$ and $\mathcal{I} \subseteq \{1, \dots, n\}$, we use the following notation:

- $x_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ is the subvector of x with elements in $\mathcal{I}$, that is,
$$x_{\mathcal{I}} = U_{\mathcal{I}}^T x;$$
- $\nabla_{\mathcal{I}} f(x) \in \mathbb{R}^{|\mathcal{I}|}$ is the vector of first-order partial derivatives of f with respect to x_i, $i \in \mathcal{I}$, that is,
$$\nabla_{\mathcal{I}} f(x) = U_{\mathcal{I}}^T \nabla f(x); \tag{2}$$
- $\nabla^2_{\mathcal{I}} f(x) \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ is the matrix of second-order partial derivatives of f with respect to x_i, $i \in \mathcal{I}$, that is,
$$\nabla^2_{\mathcal{I}} f(x) = U_{\mathcal{I}}^T \nabla^2 f(x) U_{\mathcal{I}}. \tag{3}$$

For example, if $n = 5$ and

$$x = \begin{bmatrix} 3 \\ 1 \\ 4 \\ -2 \\ 0 \end{bmatrix}, \quad \nabla f(x) = \begin{bmatrix} 2 \\ -1 \\ 0 \\ -3 \\ 4 \end{bmatrix}, \quad \nabla^2 f(x) = \begin{bmatrix} -2 & 3 & -6 & 0 & -7 \\ 3 & 1 & -5 & 4 & 2 \\ -6 & -5 & 7 & -3 & -1 \\ 0 & 4 & -3 & 5 & -4 \\ -7 & 2 & -1 & -4 & 6 \end{bmatrix},$$

using $\mathcal{I} = \{1, 3, 4\}$ we get

$$U_{\mathcal{I}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad x_{\mathcal{I}} = \begin{bmatrix} 3 \\ 4 \\ -2 \end{bmatrix}, \quad \nabla_{\mathcal{I}} f(x) = \begin{bmatrix} 2 \\ 0 \\ -3 \end{bmatrix}, \quad \nabla^2_{\mathcal{I}} f(x) = \begin{bmatrix} -2 & -6 & 0 \\ -6 & 7 & -3 \\ 0 & -3 & 5 \end{bmatrix}.$$

Note that, for any choice of $\mathcal{I} \subseteq \{1, \dots, n\}$, we have

$$\|U_{\mathcal{I}}\| = 1, \tag{4}$$
$$\|U_{\mathcal{I}} v\| = \|v\| \quad \forall v \in \mathbb{R}^{|\mathcal{I}|}. \tag{5}$$

Moreover, for any choice of $\mathcal{I} \subseteq \{1, \dots, n\}$, we define the block Lipschitz constant $L_{\mathcal{I}}$ such that

$$\|\nabla^2_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla^2_{\mathcal{I}} f(x)\| \le L_{\mathcal{I}}\|s\| \quad \forall x \in \mathbb{R}^n,\ \forall s \in \mathbb{R}^{|\mathcal{I}|}. \tag{6}$$

Note that

$$L_{\mathcal{I}} \in (0, L] \tag{7}$$

since, recalling (3), we have

$$\begin{aligned}\|\nabla^2_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla^2_{\mathcal{I}} f(x)\| &= \|U_{\mathcal{I}}^T (\nabla^2_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla^2_{\mathcal{I}} f(x)) U_{\mathcal{I}}\| \\ &\le \|\nabla^2 f(x + U_{\mathcal{I}} s) - \nabla^2 f(x)\| \|U_{\mathcal{I}}\|^2 \\ &= \|\nabla^2 f(x + U_{\mathcal{I}} s) - \nabla^2 f(x)\| \le L\|U_{\mathcal{I}} s\| = L\|s\|,\end{aligned}$$

where (4) has been used in the second equality and (5) has been used in the last equality.

Let us also define

$$L^{\min} = \min_{\mathcal{I} \subseteq \{1,\dots,n\}} L_{\mathcal{I}}.$$

From (7), it follows that

$$0 < L^{\min} \le L_{\mathcal{I}} \le L \quad \forall \mathcal{I} \subseteq \{1, \dots, n\}. \tag{8}$$

Extending known results on functions with Lipschitz continuous Hessian [11, 24], we can give the following proposition whose proof is reported in Appendix A.

Proposition 1. *Given a point $x \in \mathbb{R}^n$ and a block of variable indices $\mathcal{I} \subseteq \{1, \dots, n\}$, for all $s \in \mathbb{R}^{|\mathcal{I}|}$ we have that*

$$\|\nabla_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla_{\mathcal{I}} f(x) - \nabla^2_{\mathcal{I}} f(x) s\| \le \frac{L_{\mathcal{I}}}{2}\|s\|^2, \tag{9}$$

$$\left| f(x + U_{\mathcal{I}} s) - f(x) - \nabla_{\mathcal{I}} f(x)^T s - \frac{1}{2} s^T \nabla^2_{\mathcal{I}} f(x) s \right| \le \frac{L_{\mathcal{I}}}{6}\|s\|^3. \tag{10}$$

3 The Inexact Block Cubic Newton (IBCN) method

In this section, we describe the proposed algorithm, named *Inexact Block Cubic Newton* (IBCN) method.

At the beginning of each iteration k, we choose a block of variable indices $\mathcal{I}_k \subseteq \{1, \dots, n\}$. In order to update the variables in $\mathcal{I}_k$, we search for a suitable $s_k \in \mathbb{R}^{|\mathcal{I}_k|}$ to move from x_k along $U_{\mathcal{I}_k} s_k$. To this aim, we define the cubic model $m_k(s)$ as follows:

$$m_k(s) = q_k(s) + \frac{M_k}{6}\|s\|^3, \tag{11}$$

where M_k is a positive scalar which should overestimate $L_{\mathcal{I}_k}$, while $q_k(s)$ is the following quadratic model:

$$q_k(s) = f(x_k) + \nabla_{\mathcal{I}_k} f(x_k)^T s + \frac{1}{2} s^T \nabla^2_{\mathcal{I}_k} f(x_k) s. \tag{12}$$

For the sake of convenience, let us also report the gradient of $m_k(s)$ as follows:

$$\nabla m_k(s) = \nabla_{\mathcal{I}_k} f(x_k) + \nabla^2_{\mathcal{I}_k} f(x_k) s + \frac{M_k}{2}\|s\| s. \tag{13}$$

We then compute s_k as an approximate minimizer of the cubic model (11). To decide whether or not to accept s_k, we compute

$$\rho_k = \frac{f(x_k) - f(x_k + U_{\mathcal{I}_k} s_k)}{q_k(0) - q_k(s_k)} \tag{14}$$

(we will show in Lemma 5 below that the denominator is positive whenever $\nabla_{\mathcal{I}_k} f(x_k) \neq 0$) and we check if

$$\rho_k \geq \eta,$$

with $\eta \in (0, 1)$. If this is the case, then we accept s_k, i.e., we set $x_{k+1} = x_k + U_{\mathcal{I}_k} s_k$, referring to k as a *successful* iteration. Otherwise, we do not accept s_k, i.e., we set $x_{k+1} = x_k$, referring to k as an *unsuccessful* iteration. Let us also denote by $\mathcal{S}$ and $\mathcal{U}$ the sets of successful and unsuccessful, respectively, that is,

$$\mathcal{S} = \{k \text{ such that } \rho_k \geq \eta\} \quad \text{and} \quad \mathcal{U} = \{0, 1, \ldots\} \setminus \mathcal{S}. \tag{15}$$

In the next subsections, first we describe a greedy selection rule to chose $\mathcal{I}_k$, then we describe how we compute M_k and s_k, finally giving the algorithmic scheme.

3.1 Block selection

In block coordinate descent methods, blocks of variables can be selected by means of different rules [26, 32] such as *cyclic* (Gauss-Seidel) rules, *greedy* (Gauss-Southwell) rules and *random* rules. For block coordinate descent methods using cubic (or higher order) models, cyclic selection rules have been analyzed in [1], whereas random selection rules have been investigated in [12, 21, 33].

Here, we focus on the use of a greedy selection rule. In particular, we consider a classical Gauss-Southwell strategy [26] where, at each iteration k, the block $\mathcal{I}_k$ includes variables providing a sufficiently large amount of first-order stationarity violation.

Greedy selection rule: There exists a real number $\theta \in (0, 1]$ such that

$$\|\nabla_{\mathcal{I}_k} f(x_k)\| \geq \theta \|\nabla f(x_k)\| \quad \forall k \geq 0. \tag{16}$$

Note that the above greedy selection rule does not require the variables to be a priori partitioned into a fixed number of blocks, so that even the size of the blocks might change during the iterations.

In the following two propositions, we describe two simple procedures to satisfy the greedy selection rule (16). For any iteration k, the first one requires $\mathcal{I}_k$ to include the variable corresponding to the largest component in absolute value of $\nabla f(x_k)$, while the second one, given an arbitrary number of (possibly overlapping) blocks of variables covering $\{1, \ldots, n\}$, requires to compute the norm of the subvectors of $\nabla f(x_k)$ with respect to each block in order to choose $\mathcal{I}_k$ as the one yielding the largest norm.

Proposition 2. *For every iteration k, let $\hat{\imath}_k \in \operatorname{Argmax}_{i=1,\ldots,n} |\nabla_i f(x_k)|$ and assume that $\hat{\imath}_k \in \mathcal{I}_k$. Then,*

$$\|\nabla_{\mathcal{I}_k} f(x_k)\| \geq (n + 1 - |\mathcal{I}_k|)^{-1/2} \|\nabla f(x_k)\| \quad \forall k \geq 0.$$

It follows that (16) *is satisfied with*

$$\theta \geq \Big(n + 1 - \min_{k \geq 0} |\mathcal{I}_k|\Big)^{-1/2}.$$

Proof. Fix any iteration k and let

$$\tilde{\mathcal{I}}_k = (\{1, \dots, n\} \setminus \mathcal{I}_k) \cup \{\hat{\imath}_k\}.$$

Recalling the definition of $\hat{\imath}_k$, we have that

$$\|\nabla_{\mathcal{I}_k} f(x_k)\|_\infty = \|\nabla_{\tilde{\mathcal{I}}_k} f(x_k)\|_\infty = \|\nabla f(x_k)\|_\infty = |\nabla_{\hat{\imath}} f(x_k)|.$$

Then, we can write

$$\begin{aligned}
\|\nabla_{\mathcal{I}_k} f(x_k)\|^2 &= (\nabla_{\hat{\imath}_k} f(x_k))^2 + \sum_{i \in \mathcal{I}_k \setminus \{\hat{\imath}_k\}} \nabla_i f(x_k)^2 \\
&= \|\nabla_{\tilde{\mathcal{I}}_k} f(x_k)\|_\infty^2 + \sum_{i \in \{1,\dots,n\} \setminus \{\tilde{\mathcal{I}}_k\}} \nabla_i f(x_k)^2 \\
&\geq |\tilde{\mathcal{I}}_k|^{-1} \left(\|\nabla_{\tilde{\mathcal{I}}_k} f(x_k)\|^2 + \sum_{i \in \{1,\dots,n\} \setminus \{\tilde{\mathcal{I}}_k\}} \nabla_i f(x_k)^2 \right) \\
&= |\tilde{\mathcal{I}}_k|^{-1} \|\nabla f(x_k)\|^2.
\end{aligned}$$

Since $|\tilde{\mathcal{I}}_k| = n + 1 - |\mathcal{I}_k|$, then the desired result follows. □

Proposition 3. *For every iteration k, let $\mathcal{J}_k^1, \dots, \mathcal{J}_k^{N_k}$ be subsets of $\{1, \dots, n\}$ such that $\bigcup_{j=1}^{N_k} \mathcal{J}_k^j = \{1, \dots, n\}$ and assume that $\mathcal{I}_k \in \operatorname{Argmax}_{\mathcal{I} = \mathcal{J}_k^1, \dots, \mathcal{J}_k^{N_k}} \|\nabla_{\mathcal{I}} f(x_k)\|$. Then,*

$$\|\nabla_{\mathcal{I}_k} f(x_k)\| \geq N_k^{-1/2} \|\nabla f(x_k)\| \quad \forall k \geq 0.$$

It follows that (16) *is satisfied with*

$$\theta \geq \max_{k \geq 0} N_k^{-1/2}.$$

Proof. Fix any iteration k. Since $\bigcup_{j=1}^{N_k} \mathcal{J}_k^j = \{1, \dots, n\}$, we can write

$$\|\nabla f(x_k)\|^2 \leq \sum_{j=1}^{N_k} \|\nabla_{\mathcal{J}_k^j} f(x_k)\|^2 \leq N_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2,$$

where the last inequality follows from how $\mathcal{I}_k$ is selected, thus leading to the desired result. □

3.2 Computation of M_k

In the cubic model (11), the scalar M_k should overestimate $L_{\mathcal{I}_k}$. To account for the fact that $L_{\mathcal{I}_k}$ might be unknown, we give two different strategies to compute M_k at any iteration k, that is,

$$M_k = \begin{cases} \dfrac{L_{\mathcal{I}_k}}{1-\eta} & \text{if } L_{\mathcal{I}_k} \text{ is known,} \\ \sigma_k & \text{otherwise,} \end{cases} \tag{17}$$

where $\eta \in (0,1)$ and σ_k is updated during the iterations as follows:

$$\sigma_{k+1} = \begin{cases} \sigma_k & \text{if } k \in \mathcal{S}, \\ \gamma\sigma_k & \text{otherwise (i.e., if } k \in \mathcal{U}\text{),} \end{cases} \tag{18}$$

for a given $\gamma > 1$, with $\mathcal{S}$ and $\mathcal{U}$ defined as in (15). In particular, we will prove below that k might be unsuccessful only when M_k does not overestimates $L_{\mathcal{I}_k}$ adequately, that is, only when we use the second option in (17) (see Remark 2 below).

We see that (18) is just a simplification of the classical updating rule inherited from trust-region methods (see, e.g., [7]), differing in that, in our case, σ_{k+1} cannot be decreased from σ_k. Essentially, when the Lipschitz constant is unknown, such a choice makes σ_k increase a finite number of times until M_k provides a suitable overestimate of $L_{\mathcal{I}_k}$ (see Propositions 9–10 below).

3.3 Computation of s_k

Assuming that $\|\nabla_{\mathcal{I}_k} f(x_k)\| \neq 0$, the inexact minimizer s_k of the cubic model (11) must satisfy two conditions. The first one is that the first-order stationarity violation must be sufficiently small compared to $\|s_k\|^2$, that is,

$$\|\nabla m_k(s_k)\| \le \tau \|s_k\|^2, \tag{19}$$

with a given $\tau \in [0,\infty)$. The second requirement is that $m_k(s_k)$ must be sufficiently low, that is,

$$\begin{aligned} & m_k(s_k) \le m_k(\hat{s}_k), \quad \text{where} \\ & \hat{s}_k = -\hat{\alpha}_k \nabla_{\mathcal{I}_k} f(x_k) \quad \text{and} \quad \hat{\alpha}_k = \min\left\{\frac{\beta}{\|\nabla^2_{\mathcal{I}_k} f(x_k)\|}, \sqrt{\frac{3\beta}{M_k \|\nabla_{\mathcal{I}_k} f(x_k)\|}}\right\}, \end{aligned} \tag{20}$$

with a given $\beta \in (0,1)$, letting the first argument within the above minimum to be $+\infty$ when $\|\nabla^2_{\mathcal{I}_k} f(x_k)\| = 0$.

We see that condition (19) is a straightforward adaptation of those used in [1, 5, 9], while condition (20) is inspired by the classical Cauchy condition [7] which requires $m_k(s_k) \le \min_{\alpha \ge 0} m_k(-\alpha \nabla_{\mathcal{I}_k} f(x_k))$. In our case, $m_k(s_k)$ is compared to $m_k(\hat{s}_k)$, hence (20) is weaker than the Cauchy condition.

As to be shown, for every successful iteration k, using (19)–(20) will allow us to suitably lower bound $(f(x_k) - f(x_{k+1}))$ (see Theorem 6 below) and upper bound $\|\nabla_{\mathcal{I}_k} f(x_{k+1})\|$ (see Theorem 7 below), thus leading to the desired complexity bounds.

Note that we can compute a vector s_k satisfying (19)–(20) in finite time without the need of additional evaluations of f or its derivatives in other points. In particular, we can apply an algorithm to approximately minimize the cubic model (11) (using, e.g., the methods analyzed in [4, 6, 18, 23]). In our experiments, for the inexact minimization of the cubic model (11), we use a Barzilai-Borwein gradient method [28], which was observed to be effective in practice [4].

Finally, observe that (19)–(20) are clearly satisfied if s_k is a global minimizer of the cubic model (11) (details on how to compute global minimizers of a cubic model can be found in [7, 10, 24]).

3.4 The scheme

The proposed method, named *Inexact Block Cubic Newton* (IBCN) method, is reported in Algorithm 1.

Algorithm 1 Inexact Block Cubic Newton (IBCN) method

```
Given $x_0 \in \mathbb{R}^n$, $\sigma_0 \in (0,\infty)$, $\eta \in (0,1)$, $\gamma \in (1,\infty)$, $\tau \in [0,\infty)$ and $\beta \in (0,1)$
while $\nabla f(x_k) \neq 0$ do
    compute $\mathcal{I}_k \subseteq \{1,\dots,n\}$ such that $\|\nabla_{\mathcal{I}_k} f(x_k)\| \geq \theta \|\nabla f(x_k)\|$
    compute $M_k = \begin{cases} \dfrac{L_{\mathcal{I}_k}}{1-\eta} & \text{if } L_{\mathcal{I}_k} \text{ is known} \\ \sigma_k & \text{otherwise,} \end{cases}$
    compute $s_k$ such that
        $\|\nabla m_k(s_k)\| \leq \tau \|s_k\|^2 \quad \text{and} \quad m_k(s_k) \leq m_k(\hat{s}_k)$
        where
        $\hat{s}_k = -\hat{\alpha}_k \nabla_{\mathcal{I}_k} f(x_k)$
        $\hat{\alpha}_k = \min\left\{\dfrac{\beta}{\|\nabla^2_{\mathcal{I}_k} f(x_k)\|}, \sqrt{\dfrac{3\beta}{M_k \|\nabla_{\mathcal{I}_k} f(x_k)\|}}\right\}$
    compute $\rho_k = \dfrac{f(x_k) - f(x_k + U_{\mathcal{I}_k} s_k)}{q_k(0) - q_k(s_k)}$
    if $\rho_k \geq \eta$ then
        set $x_{k+1} = x_k + U_{\mathcal{I}_k} s_k$ and $\sigma_{k+1} = \sigma_k$        ▷ $k \in \mathcal{S}$
    else
        set $x_{k+1} = x_k$ and $\sigma_{k+1} = \gamma \sigma_k$        ▷ $k \in \mathcal{U}$
    end if
end while
```

4 Convergence analysis

We start the convergence analysis of the proposed IBCN method by bounding, for every iteration, the decrease of the cubic model similarly as when using the Cauchy condition [7].

Proposition 4. *For every iteration k, we have*

$$m_k(0) - m_k(s_k) \ge m_k(0) - m_k(\hat{s}_k) \ge (1-\beta)\hat{\alpha}_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2,$$

where $\hat{s}_k$ is defined as in (20)*.*

Proof. The first inequality of the thesis follows from the first inequality of (20), so we only have to show the second inequality. To this aim, recalling the definitions of m_k and ∇m_k from (11), (12) and (13), we can write

$$\begin{aligned} m_k(0) - m_k(\hat{s}_k) &= f(x_k) - m_k(-\hat{\alpha}_k \nabla_{\mathcal{I}_k} f(x_k)) \\ &= \hat{\alpha}_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2 - \frac{\hat{\alpha}_k^2}{2} \nabla_{\mathcal{I}_k} f(x_k)^T \nabla^2_{\mathcal{I}_k} f(x_k) \nabla_{\mathcal{I}_k} f(x_k) + \\ &\quad - \frac{\hat{\alpha}_k^3 M_k}{6} \|\nabla_{\mathcal{I}_k} f(x_k)\|^3 \\ &\ge \hat{\alpha}_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2 \Bigl(1 - \frac{\hat{\alpha}_k \|\nabla^2_{\mathcal{I}_k} f(x_k)\|}{2} - \frac{\hat{\alpha}_k^2 M_k \|\nabla_{\mathcal{I}_k} f(x_k)\|}{6}\Bigr). \end{aligned}$$

From the definition of $\hat{\alpha}_k$ given in (20), it follows that

$$1 - \frac{\hat{\alpha}_k \|\nabla^2_{\mathcal{I}_k} f(x_k)\|}{2} - \frac{\hat{\alpha}_k^2 M_k \|\nabla_{\mathcal{I}_k} f(x_k)\|}{6} \ge 1 - \frac{\beta}{2} - \frac{\beta}{2} = 1 - \beta.$$

Then, the desired result follows. □

Using the above proposition, we can easily lower bound the decrease of the quadratic model at every iteration as follows.

Lemma 5. *For every iteration k, we have*

$$q_k(0) - q_k(s_k) \ge (1-\beta)\hat{\alpha}_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2 + \frac{M_k}{6}\|s_k\|^3.$$

Proof. For any iteration k, from (12) and (11) it follows that

$$q_k(0) - q_k(s_k) = m_k(0) - m_k(s_k) + \frac{M_k}{6}\|s_k\|^3.$$

Hence, the desired result is obtained by using Proposition 4. □

In the following two theorems we show how, for every successful iteration k, we can lower bound $(f(x_k) - f(x_{k+1}))$ and upper bound $\|\nabla_{\mathcal{I}_k} f(x_{k+1})\|$.

Theorem 6. *For every $k \in \mathcal{S}$, we have*

$$f(x_k) - f(x_{k+1}) \ge \eta\Bigl((1-\beta)\hat{\alpha}_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2 + \frac{M_k}{6}\|s_k\|^3\Bigr).$$

Proof. Take any $k \in \mathcal{S}$. From the instructions of the algorithm, we have that $x_{k+1} = x_k + U_{\mathcal{I}_k} s_k$. Recalling the definition of ρ_k given in (14), then the desired result follows from the definition of $\mathcal{S}$ given in (15) and from Lemma 5. □

Remark 1. *The sequence $\{f(x_k)\}$ is monotonically non-increasing since, according to Theorem 6 and the instructions of the algorithm,*

$$f(x_{k+1}) \begin{cases} \le f(x_k) & \text{if } k \in \mathcal{S}, \\ = f(x_k) & \text{otherwise (i.e., if } k \in \mathcal{U}). \end{cases}$$

Theorem 7. *For every $k \in \mathcal{S}$, we have*

$$\|\nabla_{\mathcal{I}_k} f(x_{k+1})\| \le \Big(\tau + \frac{M_k + L_{\mathcal{I}_k}}{2}\Big)\|s_k\|^2.$$

Proof. Take any $k \in \mathcal{S}$. First, we can write

$$\|\nabla_{\mathcal{I}_k} f(x_{k+1})\| \le \|\nabla_{\mathcal{I}_k} f(x_k) + \nabla^2_{\mathcal{I}_k} f(x_k)s_k\| + \|\nabla_{\mathcal{I}_k} f(x_{k+1}) - \nabla_{\mathcal{I}_k} f(x_k) - \nabla^2_{\mathcal{I}_k} f(x_k)s_k\|. \tag{21}$$

Using (13), we can upper bound the first norm in the right-hand side of (21) as follows:

$$\begin{aligned} \|\nabla_{\mathcal{I}_k} f(x_k) + \nabla^2_{\mathcal{I}_k} f(x_k)s_k\| &= \Big\|\nabla m_k(s_k) - \frac{M_k}{2}\|s_k\|s_k\Big\| \\ &\le \|\nabla m_k(s_k)\| + \frac{M_k}{2}\|s_k\|^2 \\ &\le \Big(\tau + \frac{M_k}{2}\Big)\|s_k\|^2, \end{aligned} \tag{22}$$

where the last inequality follows from (19). Using (9) and the fact that, from the instructions of the algorithm, $x_{k+1} = x_k + U_{\mathcal{I}_k} s_k$ since $k \in \mathcal{S}$, we can also upper bound the second norm in the right-hand side of (21) as follows:

$$\|\nabla_{\mathcal{I}_k} f(x_{k+1}) - \nabla_{\mathcal{I}_k} f(x_k) - \nabla^2_{\mathcal{I}_k} f(x_k)s_k\| \le \frac{L_{\mathcal{I}_k}}{2}\|s_k\|^2. \tag{23}$$

Then, the desired result follows from (21), (22) and (23). □

To establish convergence of the algorithm, we have to upper bound the number of unsuccessful iterations, which will be obtained in Proposition 11 below. To get such a result, we have to pass through a few intermediate steps. First we show that, if M_k is a suitable overestimate of $L_{\mathcal{I}_k}$, then k is a successful iteration.

Theorem 8. *Assume that, for an iteration k, we have*

$$M_k \ge \frac{L_{\mathcal{I}_k}}{1-\eta}.$$

Then, $k \in \mathcal{S}$.

Proof. From (12), we can write

$$-\nabla_{\mathcal{I}_k} f(x_k)^T s_k - \frac{1}{2}s_k^T \nabla^2_{\mathcal{I}_k} f(x_k)s_k = q_k(0) - q_k(s_k). \tag{24}$$

Using Lemma 5, it follows that

$$-\nabla_{\mathcal{I}_k} f(x_k)^T s_k - \frac{1}{2} s_k^T \nabla^2_{\mathcal{I}_k} f(x_k) s_k \geq \frac{M_k}{6} \|s_k\|^3. \tag{25}$$

Using (24) and (10), from the definition of ρ_k given in (14) we obtain

$$1 - \rho_k = \frac{-\nabla_{\mathcal{I}_k} f(x_k)^T s_k - \frac{1}{2} s_k^T \nabla^2_{\mathcal{I}_k} f(x_k) s_k - f(x_k) + f(x_k + s_k)}{-\nabla_{\mathcal{I}_k} f(x_k)^T s_k - \frac{1}{2} s_k^T \nabla^2_{\mathcal{I}_k} f(x_k) s_k} \leq \frac{L_{\mathcal{I}_k}}{M_k},$$

where, in the last inequality, we have used (10) and (25) to upper bound the numerator by $(L_{\mathcal{I}_k}/6)\|s_k\|^3$ and lower bound the numerator by $(M_k/6)\|s_k\|^3$, respectively. Since $M_k \geq L_{\mathcal{I}_k}/(1-\eta)$ by hypothesis, it follows that $1 - \rho_k \leq 1 - \eta$, that is, $\rho_k \geq \eta$. Then, the desired result follows from the definition of $\mathcal{S}$ given in (15). □

Remark 2. *From Theorem 8 and the definition of M_k given in* (17)*, it follows that k might be an unsuccessful iteration only when M_k is set to σ_k.*

In the following two propositions, we show that both σ_k and M_k have finite positive bounds.

Proposition 9. *It holds that*

$$0 < \sigma_0 \leq \sigma_k \leq \max\left\{\sigma_0, \frac{\gamma L}{1-\eta}\right\} \quad \forall k \geq 0.$$

In particular,

$$\sigma_0 \geq \frac{L}{1-\eta} \quad \Rightarrow \quad \sigma_k = \sigma_0 \quad \forall k \geq 0.$$

Proof. From Theorem 8 and the definition of M_k given in (17), we can write

$$k \in \mathcal{U} \quad \Rightarrow \quad M_k = \sigma_k < \frac{L_{\mathcal{I}_k}}{1-\eta} \leq \frac{L}{1-\eta},$$

where we have used (8) to upper bound $L_{\mathcal{I}_k}$ in the last inequality. Taking into account the updating rule of σ_k given in (18), we get

$$\sigma_k \leq \sigma_{k+1} \leq \max\left\{\sigma_k, \frac{\gamma L}{1-\eta}\right\} \quad \forall k \geq 0,$$

Proceeding by induction and recalling that $\sigma_0 > 0$ from the algorithm initialization, the desired bounds on σ_k follows. □

Proposition 10. *Two finite positive constant M^{min} and M^{max} exist such that*

$$M^{min} \leq M_k \leq M^{max} \quad \forall k \geq 0.$$

In particular,

$$M^{min} = \min\left\{\frac{L^{min}}{1-\eta}, \sigma_0\right\} \quad \text{and} \quad M^{max} = \max\left\{\sigma_0, \frac{\gamma L}{1-\eta}\right\}.$$

Proof. Using the definition of M_k given in (17), we can write

$$\min\left\{\frac{L_{\mathcal{I}_k}}{1-\eta}, \sigma_k\right\} \le M_k \le \max\left\{\frac{L_{\mathcal{I}_k}}{1-\eta}, \sigma_k\right\} \quad \forall k \ge 0.$$

Using the bounds on $L_{\mathcal{I}_k}$ given in (8) and the bounds on σ_k given in Proposition 9, then $M^{\min} \le M_k \le M^{\max}$ for all $k \ge 0$. Finally, $M^{\min}$ and $M^{\min}$ are positive as $\sigma_0 > 0$ from the algorithm initialization. □

Using the previous results, we can give a finite upper bound on the total number of unsuccessful iterations.

Proposition 11. *There exists a finite constant U^{max} such that*

$$|\mathcal{U}| \le U^{max}.$$

In particular,

$$U^{max} = \begin{cases} 0 & \text{if } M_k \ge \frac{L_{\mathcal{I}_k}}{1-\eta} \text{ for all } k \ge 0, \\ \left\lfloor \max\left\{0, \dfrac{\log(L) - \log(\sigma_0(1-\eta))}{\log(\gamma)} + 1\right\}\right\rfloor & \text{otherwise.} \end{cases}$$

Proof. If $M_k \ge L_{\mathcal{I}_k}/(1-\eta)$ for all $k \ge 0$, it follows from Theorem 8 that $k \in \mathcal{S}$ for all $k \ge 0$. Since $\mathcal{U} \cup \mathcal{S} = \{0, 1, \ldots\}$ from (15), then we conclude that $|\mathcal{U}| = 0$.

Now assume that $\sigma_0 \ge L/(1-\eta)$. Since, from Proposition 9 and (8), for all $k \ge 0$ we have $\sigma_k \ge \sigma_0$ and $L \ge L_{\mathcal{I}_k}$, respectively, then $\sigma_k \ge L_{\mathcal{I}_k}/(1-\eta)$ for all $k \ge 0$. Using the definition of M_k given in (17), we conclude that $M_k \ge L_{\mathcal{I}_k}/(1-\eta)$ for all $k \ge 0$, still obtaining $|\mathcal{U}| = 0$ reasoning as in the previous case.

The last case to analyze is when $\sigma_0 < L/(1-\eta)$. For any iteration $k \ge 1$, define

$$j_k = |\{j < k \colon j \in \mathcal{U}\}|,$$

that is, j_k is the number of unsuccessful iterations up to iteration k. From the updating rule of σ_k given in (18), we have that $\sigma_k = \gamma^{j_k}\sigma_0$ for all $k \ge 1$. So, using the upper bound on σ_k given in Proposition 9, we can write

$$\gamma^{j_k}\sigma_0 = \sigma_k \le \max\left\{\sigma_0, \frac{\gamma L}{1-\eta}\right\} \le \frac{\gamma L}{1-\eta} \quad \forall k \ge 1,$$

where the last inequality follows from the fact that we are considering the case $\sigma_0 < L/(1-\eta)$ and $\gamma > 1$. Applying the logarithm, we get

$$j_k \le \left\lfloor \max\left\{0, \frac{\log(L) - \log(\sigma_0(1-\eta))}{\log(\gamma)} + 1\right\}\right\rfloor \quad \forall k \ge 1.$$

Since $|U| = \max_{k \ge 1} j_k$, then the desired result follows. □

Remark 3. *As appears from the proof of Proposition 11, if $\sigma_0 \ge L/(1-\eta)$, then $U^{max} = 0$.*

Now, to establish convergence to stationary points, we need an assumption on the boundedness of f and $\nabla^2 f$ over the following level set:

$$\mathcal{L}^0 = \{x \in \mathbb{R}^n \colon f(x) \le f(x_0)\}. \tag{26}$$

Assumption 1. *Two finite positive constants f^{min} and B exist such that, for all $x \in \mathcal{L}^0$, we have $f(x) \ge f^{min}$ and $\|\nabla^2 f(x)\| \le B$, where $\mathcal{L}^0$ is defined as in* (26)*.*

We see that Assumption 1 is satisfied if $\mathcal{L}^0$ is compact. Note also that, since $\{f(x_k)\}$ is monotonically non-increasing from Remark 1, then $\{x_k\} \subseteq \mathcal{L}^0$. It follows that, under Assumption 1, we have

$$f(x_k) \ge f^{\min} \quad \forall k \ge 0, \tag{27}$$

$$\|\nabla^2_{\mathcal{I}_k} f(x_k)\| \le B \quad \forall k \ge 0. \tag{28}$$

In oder to show convergence of IBCN to stationary points, we first give the following result, which will also be useful in the worst-case iteration complexity analysis.

Proposition 12. *Given $\varepsilon \in [0,1]$, if Assumption 1 holds, then*

$$f(x_k) - f(x_{k+1}) \ge c_1 \varepsilon^2 \quad \forall k \in \mathcal{S} \colon \|\nabla f(x_k)\| \ge \varepsilon,$$

where

$$c_1 = \theta\eta(1-\beta) \min\biggl\{\frac{\theta\beta}{B}, \sqrt{\frac{3\theta\beta}{M^{max}}}\biggr\}.$$

Proof. Take any iteration $k \in \mathcal{S}$ such that $\|\nabla f(x_k)\| \ge \varepsilon$, with $\varepsilon \in [0,1]$. Using Theorem 6 and the greedy selection rule (16), we have that

$$f(x_k) - f(x_{k+1}) \ge \eta(1-\beta)\hat\alpha_k \|\nabla_{\mathcal{I}_k} f(x_k)\|^2 \ge \theta\eta(1-\beta)\hat\alpha_k \|\nabla f(x_k)\| \|\nabla_{\mathcal{I}_k} f(x_k)\|.$$

Since $\|\nabla f(x_k)\| \ge \varepsilon$, we obtain

$$f(x_k) - f(x_{k+1}) \ge \theta\eta(1-\beta)\varepsilon\hat\alpha_k \|\nabla_{\mathcal{I}_k} f(x_k)\|. \tag{29}$$

Now, using the definition of $\hat\alpha_k$ given in (20), we can write

$$\hat\alpha_k \|\nabla_{\mathcal{I}_k} f(x_k)\| = \min\biggl\{\frac{\beta\|\nabla_{\mathcal{I}_k} f(x_k)\|}{\|\nabla^2_{\mathcal{I}_k} f(x_k)\|}, \sqrt{\frac{3\beta\|\nabla_{\mathcal{I}_k} f(x_k)\|}{M_k}}\biggr\}.$$

Since, from (28) and Proposition 10, respectively, $\|\nabla^2_{\mathcal{I}_k} f(x_k)\| \le B$ and $M_k \le M^{\max}$, then we get

$$\hat\alpha_k \|\nabla_{\mathcal{I}_k} f(x_k)\| \ge \min\biggl\{\frac{\beta\|\nabla_{\mathcal{I}_k} f(x_k)\|}{B}, \sqrt{\frac{3\beta\|\nabla_{\mathcal{I}_k} f(x_k)\|}{M^{\max}}}\biggr\}.$$

So, using the greedy selection rule (16) and the fact that $\|\nabla f(x_k)\| \ge \varepsilon$, with $\varepsilon \in [0,1]$, we obtain

$$\hat\alpha_k \|\nabla_{\mathcal{I}_k} f(x_k)\| \ge \min\biggl\{\frac{\theta\beta\varepsilon}{B}, \sqrt{\frac{3\theta\beta\varepsilon}{M^{\max}}}\biggr\} \ge \varepsilon \min\biggl\{\frac{\theta\beta}{B}, \sqrt{\frac{3\theta\beta}{M^{\max}}}\biggr\}. \tag{30}$$

Then, combining (29) and (30), the desired result follows. □

Now, we can finally show global convergence of IBCN to stationary points.

Theorem 13. *If Assumption 1 holds, then*

$$\lim_{k\to\infty} \nabla f(x_k) = 0.$$

Proof. Since $|\mathcal{U}|$ is bounded from Proposition 11, with $\mathcal{U} \cup \mathcal{S} = \{0, 1, \ldots\}$ from (15), then an iteration $\bar{k}$ exists such that $k \in \mathcal{S}$ for all $k \geq \bar{k}$. Now, reasoning by contradiction, assume that $\{\nabla f(x_k)\}$ does not converge to 0. It follows that there exists $\varepsilon \in (0, 1]$ such that

$$\limsup_{k\to\infty} \|\nabla f(x_k)\| \geq \varepsilon.$$

Since $k \in \mathcal{S}$ for all $k \geq \bar{k}$, from Proposition 12 it follows that $\{f(x_k)\} \to -\infty$, thus contradicting (27). □

4.1 Worst-case iteration complexity

Here, we analyze the worst-case iteration complexity of the proposed IBCN method, providing two main results.

First, in the following theorem, we show that at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations are needed to drive $\|\nabla_{\mathcal{I}_k} f(x_{k+1})\|$ below a given threshold $\varepsilon > 0$. Note that, in the proof of the following theorem, no role is played by the greedy selection rule (16), that is, the result holds for any arbitrary choice of the blocks.

Theorem 14. *Given $\varepsilon > 0$, let*

$$K^b_\varepsilon = \{k \geq 0 \colon \|\nabla_{\mathcal{I}_k} f(x_{k+1})\| \geq \varepsilon\}.$$

If Assumption 1 holds, then

$$|K^b_\varepsilon| \leq \left\lfloor \frac{f(x_0) - f^{min}}{c_2} \varepsilon^{-3/2} \right\rfloor + U^{max},$$

where

$$c_2 = \frac{M^{min}}{6}\left(\tau + \frac{M^{max} + L}{2}\right)^{-3/2}.$$

Proof. Since $\mathcal{S} \cup \mathcal{U} = \{0, 1, \ldots, \}$ from (15), then

$$|K^{\mathrm{b}}_\varepsilon| = |K^{\mathrm{b}}_\varepsilon \cap \mathcal{S}| + |K^{\mathrm{b}}_\varepsilon \cap \mathcal{U}|. \tag{31}$$

To obtain the desired result, in the following we want to upper bound $|K^{\mathrm{b}}_\varepsilon \cap \mathcal{S}|$ and $|K^{\mathrm{b}}_\varepsilon \cap \mathcal{U}|$.

Using the lower bound for $f(x_k)$ given in (27) and the fact that, from Remark 1, $\{f(x_k)\}$ is monotonically non-increasing, we can write

$$f(x_0) - f^{\min} \geq \sum_{k \in K^{\mathrm{b}}_\varepsilon \cap \mathcal{S}} (f(x_k) - f(x_{k+1})). \tag{32}$$

Now we want to lower bound the right-hand side term of (32). First, from Theorem 6, we have that

$$f(x_k) - f(x_{k+1}) \geq \frac{M_k}{6}\|s_k\|^3 \geq \frac{M^{\min}}{6}\|s_k\|^3 \quad \forall k \in \mathcal{S}, \tag{33}$$

where, in the last inequality, we have Proposition 10 to lower bound M_k. Moreover, from Theorem 7, we have that

$$\|\nabla_{\mathcal{I}_k} f(x_{k+1})\| \leq \left(\tau + \frac{M_k + L_{\mathcal{I}_k}}{2}\right)\|s_k\|^2 \leq \left(\tau + \frac{M^{\max} + L}{2}\right)\|s_k\|^2 \quad \forall k \in \mathcal{S}, \tag{34}$$

where, in the last inequality, we have used Proposition 10 and (8) to upper bound M_k and $L_{\mathcal{I}_k}$, respectively. Therefore, from (33) and (34), we obtain

$$f(x_k) - f(x_{k+1}) \geq c_2\|\nabla_{\mathcal{I}_k} f(x_{k+1})\|^{3/2} \quad \forall k \in \mathcal{S}. \tag{35}$$

It follows that

$$f(x_k) - f(x_{k+1}) \geq c_2\varepsilon^{3/2} \quad \forall k \in K_\varepsilon^{\mathrm{b}} \cap \mathcal{S}.$$

Using this inequality in the right-hand side of (32), we obtain

$$f(x_0) - f^{\min} \geq |K_\varepsilon^{\mathrm{b}} \cap \mathcal{S}| c_2\varepsilon^{3/2}.$$

Hence, we can upper bound $|K_\varepsilon^{\mathrm{b}} \cap \mathcal{S}|$ as follows:

$$|K_\varepsilon^{\mathrm{b}} \cap \mathcal{S}| \leq \left\lfloor \frac{f(x_0) - f^{\min}}{c_2}\varepsilon^{-3/2} \right\rfloor. \tag{36}$$

Now, using Proposition 11, we can also upper bound $|K_\varepsilon^{\mathrm{b}} \cap \mathcal{U}|$ as follows:

$$|K_\varepsilon^{\mathrm{b}} \cap \mathcal{U}| \leq |\mathcal{U}| \leq U^{\max}. \tag{37}$$

Then, the desired result follows from (31), (36) and (37). □

From Theorem 14 we see that, to drive the stationarity violation with respect to *the selected block of variables* below ε, we need at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations, thus matching the complexity bound given in [1] for cyclic selection.

In particular, when $\mathcal{I}_k = \{1, \dots, n\}$ for all k, we retain the complexity bound of standard cubic Newton methods, that is, at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations are needed to obtain $\|\nabla f(x_k)\| < \varepsilon$.

In a general case where $|I_k| < n$, Theorem 14 does not provide information on how many iterations are needed in the worst case to drive the stationarity violation with respect to *all variables* below ε. Such a complexity bound is given in the next theorem, ensuring that at most $\mathcal{O}(\varepsilon^{-2})$ are needed to get $\|\nabla f(x_k)\| < \varepsilon$.

Theorem 15. *Given $\varepsilon \in (0, 1]$, let*

$$K_\varepsilon = \{k \geq 0 \colon \|\nabla f(x_k)\| \geq \varepsilon\}.$$

If Assumption 1 holds, then

$$|K_\varepsilon| \leq \left\lfloor \frac{f(x_0) - f^{min}}{c_1}\varepsilon^{-2} \right\rfloor + U^{max},$$

where c_1 is defined as in Proposition 12.

Proof. Since $\mathcal{S} \cup \mathcal{U} = \{0, 1, \dots, \}$ from (15), then

$$|K_\varepsilon| = |K_\varepsilon \cap \mathcal{S}| + |K_\varepsilon \cap \mathcal{U}|. \tag{38}$$

To obtain the desired result, in the following we want to upper bound $|K_\varepsilon \cap \mathcal{S}|$ and $|K_\varepsilon \cap \mathcal{U}|$.

Using the lower bound for $f(x_k)$ given in (27) and the fact that, from Remark 1, $\{f(x_k)\}$ is monotonically non-increasing, we can write

$$f(x_0) - f^{\min} \ge \sum_{k \in K_\varepsilon \cap \mathcal{S}} (f(x_k) - f(x_{k+1})).$$

From Proposition 12, it follows that

$$f(x_0) - f^{\min} \ge |K_\varepsilon \cap \mathcal{S}| c_1 \varepsilon^2.$$

Hence, we can upper bound $|K_\varepsilon \cap \mathcal{S}|$ as follows:

$$|K_\varepsilon \cap \mathcal{S}| \le \left\lfloor \frac{f(x_0) - f^{\min}}{c_1} \varepsilon^{-2} \right\rfloor. \tag{39}$$

Now, using Proposition 11, we can also upper bound $|K_\varepsilon \cap \mathcal{U}|$ as follows:

$$|K_\varepsilon \cap \mathcal{U}| \le |\mathcal{U}| \le U^{\max}. \tag{40}$$

Then, the desired result follows from (38), (39) and (40). □

Remark 4. *Since $c_1 = \mathcal{O}(\theta^{3/2})$, with $\theta \in (0, 1]$, it follows that the larger θ the better the complexity bound of Theorem 15. Lower bounds for θ have been derived in Propositions 2–3 when using two simple strategies for the block selection.*

First-order methods usually guarantee an upper bound of $\mathcal{O}(\varepsilon^{-2})$ on the maximum number of iterations needed to obtain $\|\nabla f(x_k)\| < \varepsilon$ since, at every iteration, they satisfy $f(x_k) - f(x_{k+1}) \ge c\|\nabla f(x_k)\|^2$, with a finite positive constant c, using either a greedy [25] or a cyclic [2] block selection rule. So, for the proposed IBCN method, Theorem 15 ensures the same worst-case iteration complexity as first-order methods.

When using cyclic block selection with cubic models, an upper bound of $\mathcal{O}(\varepsilon^{-3})$ was obtained in [1] on the maximum number of iterations needed to obtain $\|\nabla f(x_k)\| < \varepsilon$, thus worse than the proposed IBCN method and than first-order methods.

5 Numerical experiments

In this section, we report some numerical results. The experiments were run in Matlab R2024a on an Apple MacBook Pro with an Apple M1 Pro Chip and 16 GB RAM.

Given a set of samples $\{a^1, \dots, a^m\} \subseteq \mathbb{R}^n$ and labels $\{b^1, \dots, b^m\} \subseteq \mathbb{R}$, let $\varphi_x \colon \mathbb{R}^n \to \mathbb{R}$ be a prediction function parameterized by a vector x. We consider optimization problems from regression and classification models where the objective function has the following form:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} \ell(b^i, \varphi_x(a^i)) + \lambda P(x),$$

with $\ell\colon \mathbb{R}\times\mathbb{R}\to[0,\infty)$ being a loss function, $P\colon \mathbb{R}^n\to[0,\infty)$ being a regularizer and $\lambda\geq 0$ being a regularization parameter. In particular, a non-convex regression model is considered in Subsection 5.1, while a convex classification model is considered in Subsection 5.2.

We compare IBCN with two block coordinate descent methods using greedy selection rules. In particular, we consider both a first-order method and a second-order method, referred to as BCD1 and BCD2, respectively. At each iteration k of BCD1 and BCD2, given the current point x_k and a block of variables $\mathcal{I}_k$, we compute a search direction $d_k\in\mathbb{R}^{|\mathcal{I}_k|}$ as follows:

- For BCD1, we use the steepest descent direction, that is,
$$d_k = -\nabla_{\mathcal{I}_k} f(x_k);$$
- For BCD2, we use a diagonally scaled steepest descent direction [3, 30], that is,
$$d_k = -(H_k)^{-1}\nabla_{\mathcal{I}_k} f(x_k),$$
where $H_k\in\mathbb{R}^{|\mathcal{I}_k|\times|\mathcal{I}_k|}$ is symmetric and positive definite. To compute H_k, we choose a diagonal Hessian approximation as in [30, Subsection 7.2], that is,
$$H_k = \operatorname{diag}(v_k), \quad \text{with} \quad v_k = \bigl[\min\{\max\{\nabla^2_{\{j\}} f(x_k), 10^{-2}\}, 10^9\}\bigr]_{j\in\mathcal{I}_k},$$
where $\operatorname{diag}(v_k)$ denotes the diagonal matrix constructed from the vector v_k.

For both BCD1 and BCD2, once d_k is obtained, we set $x_{k+1} = x_k + \alpha_k U_{\mathcal{I}_k} d_k$, with α_k being computed by means of an Armijo line search, similarly as in [3, 30].

At each iteration k of IBCN, BCD1 and BCD2, a block of variables $\mathcal{I}_k$ is chosen as described in Proposition 2, that is, such that $\|\nabla_{\mathcal{I}_k} f(x_k)\|_\infty = \|\nabla f(x_k)\|_\infty$. More specifically, first we compute the index $\hat{\imath}_k$ corresponding to the largest component in absolute value of $\nabla f(x_k)$, then $\mathcal{I}_k$ is set to include $\hat{\imath}_k$ with the other variable indices being chosen randomly. In our experiments, we use blocks of size $q\in\{1,5,10,20,50,100\}$.

In IBCN, we set $\sigma_0 = 1$, $\eta = 0.1$, $\gamma = 2$ and $\tau = 1$, choosing M_k at each iteration k by the second option of (17). To compute s_k at each iteration k of IBCN, we set $s_k = \hat{s}_k$, with $\hat{s}_k$ defined as in (20), if this choice satisfies (19). Otherwise, we run a Barzilai-Borwein gradient method [28] to $m_k(s)$, starting from $\hat{s}_k$, until a point s is produced such that (19) holds with s_k replaced by s.

In all experiments, we run IBCN, BCD1 and BCD2 from the starting point $x_0 = 0$ for 10^4 iterations without using any other stopping condition. Then, considering a sequence $\{x_k\}$ produced by a given algorithm, we analyze the decrease of the objective error $(f(x_k)-f^*)$, with f^* being the best objective value found for a given problem, and the decrease of the stationarity violation $\|\nabla f(x_k)\|$.

5.1 Sparse least squares

The problem of recovering sparse vectors from linear measurements is central in many applications, such as compressive sensing [13] and variable selection [16]. To obtain sparse solutions, a popular approach is to use least-square with l_1-norm regularization, resulting in a convex formulation known as LASSO [29]. But, in order to overcome the bias connected to the l_1 norm, some non-convex regularizers have also been introduced in the literature [31].

Here we consider a non-convex sparsity promoting term considered in, e.g., [22, 27], given by $P(x) = \sum_{i=1}^{n}(x_i^2 + \omega^2)^{p/2}$, with small $\omega > 0$ and $p \in (0, 1)$. Using the least squares as loss, we hence obtain the following non-convex problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{m}\|Ax - b\|^2 + \lambda \sum_{i=1}^{n}(x_i^2 + \omega^2)^{p/2},$$

where $A = \begin{bmatrix} a^1 & \dots & a^m \end{bmatrix}^T \in \mathbb{R}^{m \times n}$ and $b = \begin{bmatrix} b^1 & \dots & b^m \end{bmatrix}^T \in \mathbb{R}^m$. In our experiments we set $\lambda = 10^{-3}$, $\omega = 10^{-2}$ and $p = 0.5$. After generating the elements of the matrix A randomly from a uniform distribution in $(0, 1)$, with $m = 500$ and $n = 10,000$, a vector $\hat{x} \in \mathbb{R}^n$ was created with all components equal to zero except for 5% of them, which were randomly set to ±1. Then, we set $b = A\hat{x} + \zeta$, where $\zeta \in \mathbb{R}^m$ is a noise vector with elements drawn from a normal distribution mean 0 and standard deviation 10^{-3}. We run 10 simulations and the average results with respect to the number of iterations and the CPU time are reported in Figures 1–2, respectively.

We see that, for $q = 1$, all the considered methods perform very similarly and give almost identical results, but IBCN clearly outperforms both BCD1 and BCD2 as the size of the blocks increases. In particular, for $q \geq 5$, IBCN makes both the objective function and the norm of its gradient decrease much faster. Within the given limit of 10^4 iterations, IBCN is always able to achieve a lower objective value with a smaller norm gradient.

5.2 Regularized logistic regression

To asses how IBCN works on convex problems, we consider the l_2-regularized logistic regression. In particular, assuming that $b_i \in \{\pm 1\}$, $i = 1, \dots, m$, the optimization problem can be formulated as follows:

$$\min_{(x,z) \in \mathbb{R}^{n+1}} \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + e^{-b^i(a_i^T x + z)}\Big) + \lambda\|x\|^2,$$

We use the following three datasets from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`:

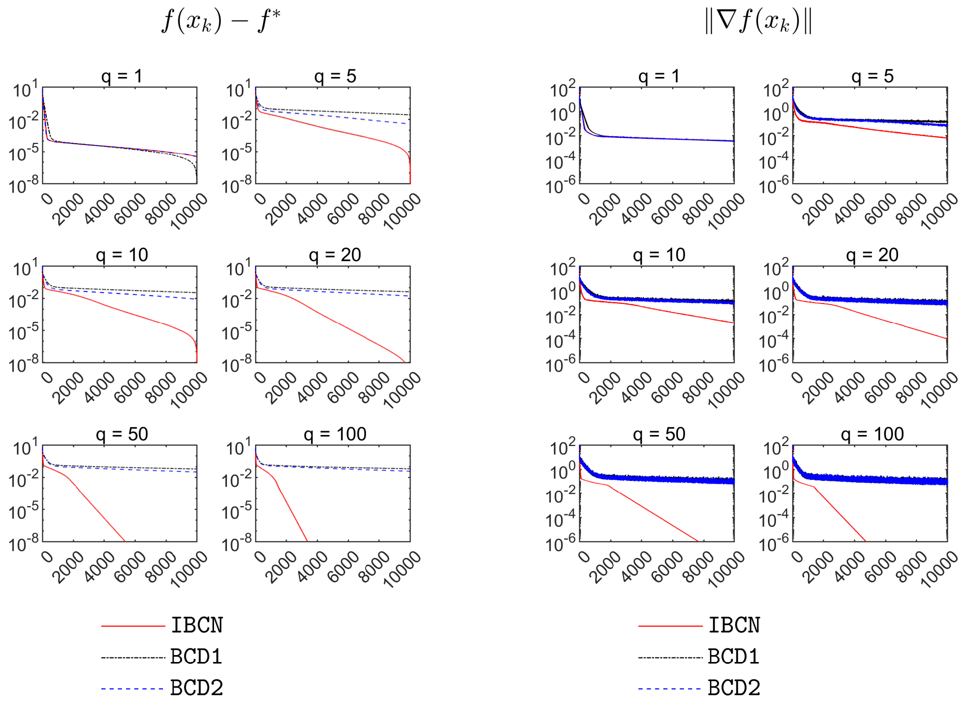
(i) gisette (train), $m = 6000$, $n = 5000$;

(ii) leu (train), $m = 38$, $n = 7129$;

(iii) madelon (train), $m = 2000$, $n = 500$;

scaling all features of the last dataset in $[-1, 1]$, while the other ones had already been scaled or normalized.

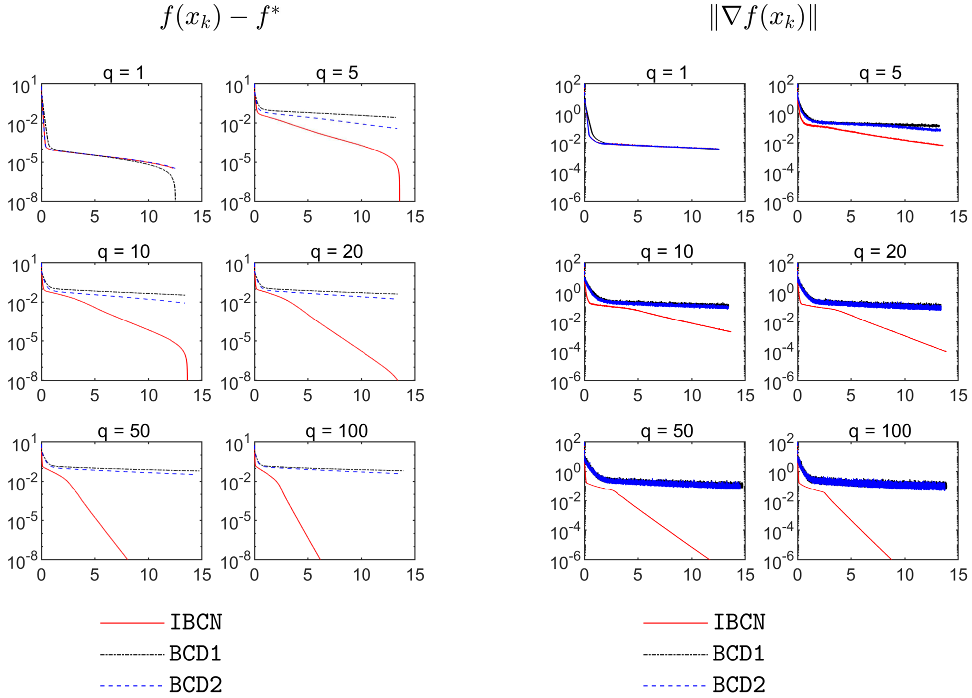
Results with respect to the number of iterations are reported in Figures 3–4. We see that, for $q = 1$, IBCN and BCD2 have similar performance and both of them give better results



(a)

(b)

Figure 1: Results on sparse least squares with respect to the number of iterations, using blocks of size q. In each plot, the y axis in logarithmic scale.



(a)

(b)

Figure 2: Results on sparse least squares with respect to the CPU time in seconds, using blocks of size q. In each plot, the y axis in logarithmic scale.

than BCD1. For larger values of q, IBCN provides a faster objective decrease and is able to produce points with a smaller norm of ∇f than the two competitive methods. Also for this class of problems, the performances of IBCN improve as the size of the blocks increases, similarly as what was obtained in the previous subsection for non-convex problems.

Finally, results with respect to the CPU time are reported in Figure 5 only for the gisette dataset since, for the other datasets, the methods take a few seconds in most cases. We see that IBCN seems to provide the best results for $q \geq 5$, confirming the above findings.

6 Conclusions

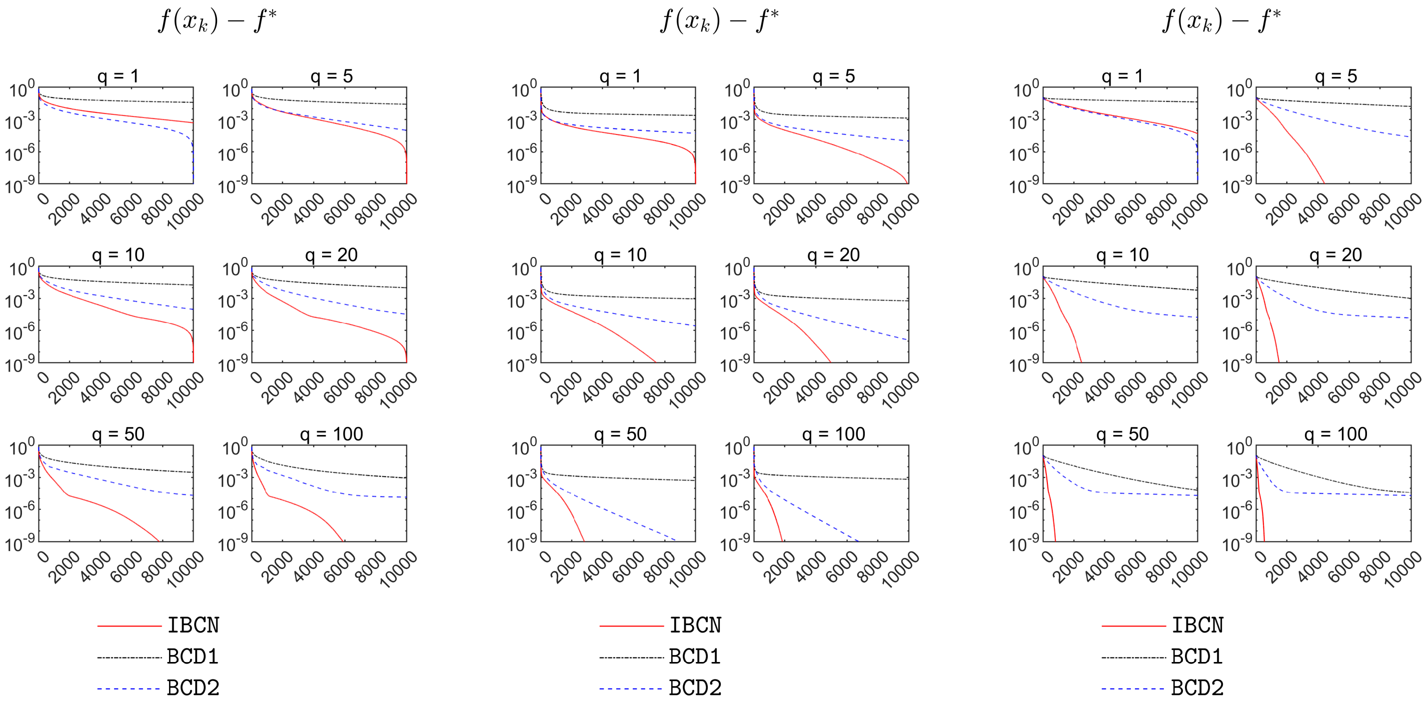
In this paper, we have considered the unconstrained minimization of an objective function with Lipschitz continuous Hessian. For this problem, we have presented a block coordinate descent version of cubic Newton methods using a greedy (Gauss-Southwell) selection rule, where blocks of variables are chosen by considering the amount of first-order stationarity violation. To update the selected block at each iteration, an inexact minimizer of a cubic model is computed. In practice, such an inexact minimization can be carried out in finite time without the need of additional evaluations of the objective function or its derivatives in other points. In the proposed scheme, blocks are not required to have a prefixed structure and their size might even change during the iterations. Moreover, the knowledge of the Lipschitz constant of the Hessian is not needed.

In a non-convex setting, we have shown global convergence to stationary points and analyzed the worst-case iteration complexity. Specifically, we have shown that at most $\mathcal{O}(\varepsilon^{-3/2})$ iterations are needed to drive the stationarity violation with respect to the selected block of variables below ε, while at most $\mathcal{O}(\varepsilon^{-2})$ iterations are needed to drive the stationarity violation with respect to all variables below ε.

Then, we have tested the proposed method on non-convex and convex problems used to build regression and classification models. Numerical results show that the proposed approach is effective and its performances improve as the size of the blocks increases.

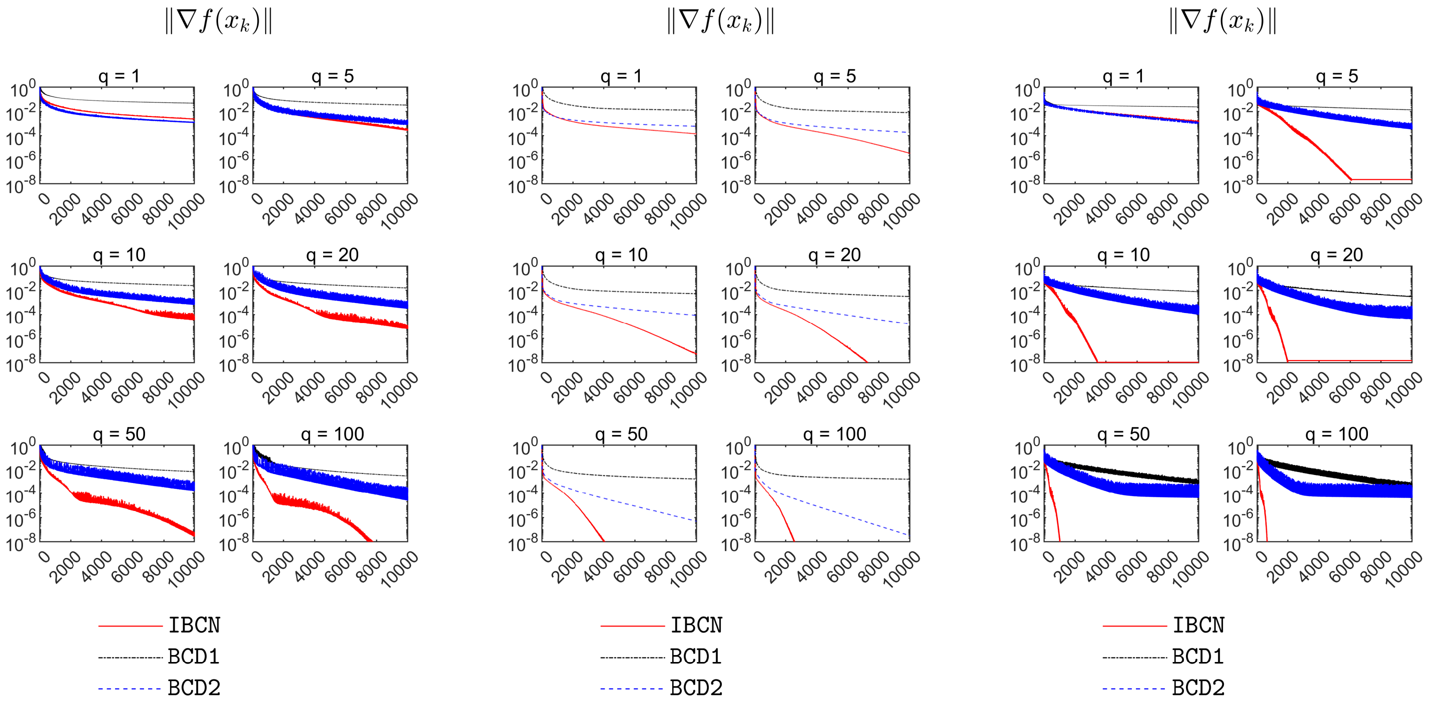
Finally, further investigation needs to be devoted to analyzing the worst-case iteration complexity in convex and strongly convex problems.

Data availability Codes are available at `https://github.com/acristofari/ibcn`, including those to generate the datasets used in Subsection 5.1, whereas the datasets used in Subsection 5.2 were downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.



(a) gisette (b) leu (c) madelon

Figure 3: Results on l_2-regularized logistic regression with respect to the number of iterations, using blocks of size q, for gisette dataset (a), leu dataset (b) and madelon dataset (c). In each plot, the y axis in logarithmic scale.

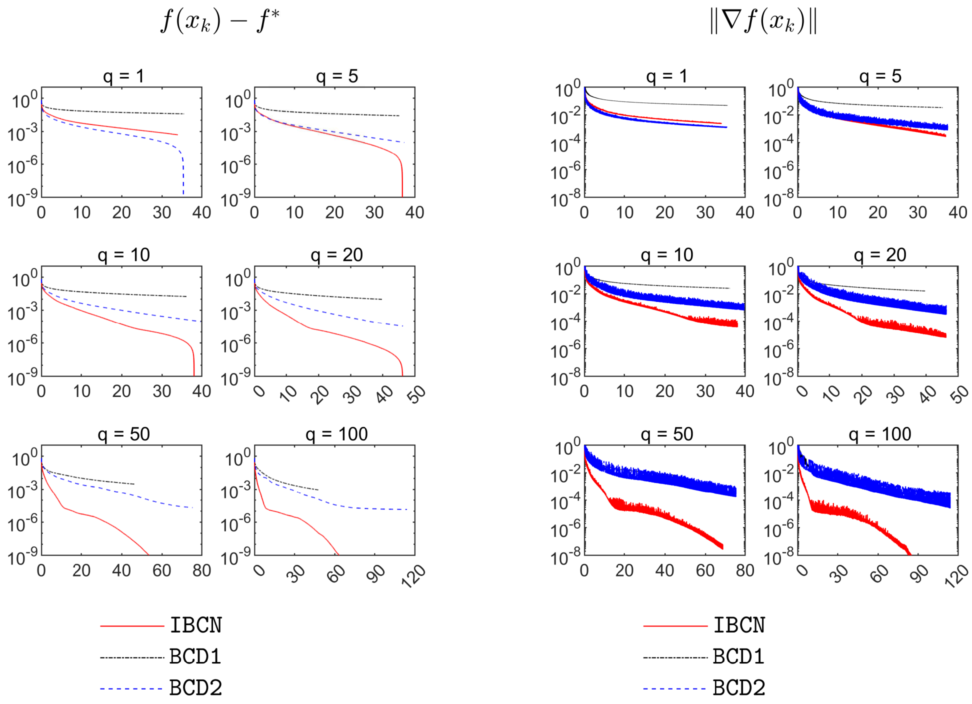


(a) gisette

(b) leu

(c) madelon

Figure 4: Results on l_2-regularized logistic regression with respect to the number of iterations, using blocks of size q, for gisette dataset (a), leu dataset (b) and madelon dataset (c). In each plot, the y axis in logarithmic scale.



(a)

(b)

Figure 5: Results on l_2-regularized logistic regression with respect to the CPU time in seconds, using blocks of size q, for gisette dataset. In each plot, the y axis in logarithmic scale.

Appendix A Properties from Lipschitz continuity

Proof of Proposition 1. Choose $\mathcal{I} \subseteq \{1,\dots,n\}$ and define the function $\psi \colon \mathbb{R}^n \to \mathbb{R}^{|\mathcal{I}|}$, $\psi(x) = U_{\mathcal{I}}^T \nabla f(x)$. Namely, using (2),

$$\psi(x) = \nabla_{\mathcal{I}} f(x) \quad \forall x \in \mathbb{R}^n.$$

Now, take $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^{|\mathcal{I}|}$. Applying the mean value theorem to ψ, we can write

$$\begin{aligned} \nabla_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla_{\mathcal{I}} f(x) &= \psi(x + U_{\mathcal{I}} s) - \psi(x) \\ &= \int_0^1 \nabla \psi(x + tU_{\mathcal{I}} s)^T U_{\mathcal{I}} s \, dt \\ &= \int_0^1 U_{\mathcal{I}}^T \nabla^2 f(x + tU_{\mathcal{I}} s) U_{\mathcal{I}} s \, dt \\ &= \int_0^1 \nabla_{\mathcal{I}}^2 f(x + tU_{\mathcal{I}} s) s \, dt, \end{aligned}$$

where we have used (3) in the last equality. Adding $-\nabla_{\mathcal{I}}^2 f(x) s$ to all terms, we obtain

$$\begin{aligned} \|\nabla_{\mathcal{I}} f(x + U_{\mathcal{I}} s) - \nabla_{\mathcal{I}} f(x) - \nabla_{\mathcal{I}}^2 f(x) s\| &= \left\| \int_0^1 (\nabla_{\mathcal{I}}^2 f(x + tU_{\mathcal{I}} s) - \nabla_{\mathcal{I}}^2 f(x)) s \, dt \right\| \\ &\le \int_0^1 \|(\nabla_{\mathcal{I}}^2 f(x + tU_{\mathcal{I}} s) - \nabla_{\mathcal{I}}^2 f(x)) s\| \, dt \\ &\le \|s\| \int_0^1 \|\nabla_{\mathcal{I}}^2 f(x + tU_{\mathcal{I}} s) - \nabla_{\mathcal{I}}^2 f(x)\| \, dt \\ &\le L_{\mathcal{I}} \|s\|^2 \int_0^1 t \, dt \\ &= \frac{L_{\mathcal{I}}}{2} \|s\|^2, \end{aligned} \tag{41}$$

where the last inequality follows from (6). Thus, (9) holds.

To show (10), by the mean value theorem we can write

$$f(x + U_{\mathcal{I}} s) - f(x) = \int_0^1 \nabla f(x + tU_{\mathcal{I}} s)^T U_{\mathcal{I}} s \, dt = \int_0^1 \nabla_{\mathcal{I}} f(x + tU_{\mathcal{I}} s)^T s \, dt, \tag{42}$$

where we have used (2) in the last equality. Adding $-\nabla_{\mathcal{I}} f(x)^T s - \frac{1}{2} s^T \nabla_{\mathcal{I}}^2 f(x) s$ to all terms, we obtain

$$\begin{aligned}
&\Big|f(x+U_{\mathcal{I}}s)-f(x)-\nabla_{\mathcal{I}}f(x)^T s-\frac{1}{2}s^T\nabla^2_{\mathcal{I}}f(x)s\Big| = \\
&\Big|\int_0^1 (\nabla_{\mathcal{I}}f(x+tU_{\mathcal{I}}s)-\nabla_{\mathcal{I}}f(x)-t\nabla^2_{\mathcal{I}}f(x)s)^T s\,dt\Big| \le \\
&\int_0^1 \big|(\nabla_{\mathcal{I}}f(x+tU_{\mathcal{I}}s)-\nabla_{\mathcal{I}}f(x)-t\nabla^2_{\mathcal{I}}f(x)s)^T s\big|dt \le \\
&\|s\|\int_0^1 \|\nabla_{\mathcal{I}}f(x+tU_{\mathcal{I}}s)-\nabla_{\mathcal{I}}f(x)-t\nabla^2_{\mathcal{I}}f(x)s\|dt \le \\
&\frac{L_{\mathcal{I}}}{2}\|s\|^3\int_0^1 t^2\,dt = \\
&\frac{L_{\mathcal{I}}}{6}\|s\|^3,
\end{aligned}$$

where the last inequality follows from (9). Thus, (10) holds. □

References

[1] V. Amaral, R. Andreani, E. Birgin, D. Marcondes, and J. M. Martínez. On complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization. *Journal of Global Optimization*, 84(3):527–561, 2022. doi: 10.1007/s10898-022-01168-6.

[2] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013. doi: 10.1137/120887679.

[3] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, Belmont, MA, 1999.

[4] T. Bianconcini, G. Liuzzi, B. Morini, and M. Sciandrone. On the use of iterative methods in cubic regularization for unconstrained optimization. *Computational Optimization and Applications*, 60:35–57, 2015. doi: 10.1007/s10589-014-9672-x.

[5] E. G. Birgin, J. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163:359–368, 2017. doi: 10.1007/s10107-016-1065-8.

[6] Y. Carmon and J. Duchi. Gradient Descent Finds the Cubic-Regularized Nonconvex Newton Step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019. doi: 10.1137/17M1113898.

[7] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011. doi: 10.1007/s10107-009-0286-5.

[8] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011. doi: 10.1007/s10107-009-0337-y.

[9] C. Cartis, N. I. Gould, and P. L. Toint. Universal Regularization Methods: Varying the Power, the Smoothness and the Accuracy. *SIAM Journal on Optimization*, 29(1): 595–615, 2019. doi: 10.1137/16M1106316.

[10] A. Cristofari, T. Dehghan Niri, and S. Lucidi. On global minimizers of quadratic functions with cubic regularization. *Optimization Letters*, 13:1269–1283, 2019. doi: 10.1007/s11590-018-1316-0.

[11] J. E. Dennis Jr and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996. doi: 10.1137/1.9781611971200.

[12] N. Doikov and P. Richtárik. Randomized Block Cubic Newton Method. In *International Conference on Machine Learning*, pages 1290–1298. PMLR, 2018.

[13] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4): 1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

[14] J.-P. Dussault. $\mathrm{ARC_q}$: a new adaptive regularization by cubics. *Optimization Methods and Software*, 33(2):322–335, 2018. doi: 10.1080/10556788.2017.1322080.

[15] J.-P. Dussault, T. Migot, and D. Orban. Scalable adaptive cubic regularization methods. *Mathematical Programming*, pages 1–35, 2023. doi: 10.1007/s10107-023-02007-6.

[16] J. Fan and R. Li. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273.

[17] K. Fountoulakis and R. Tappenden. A flexible coordinate descent method. *Computational Optimization and Applications*, 70(2):351–394, 2018. doi: 10.1007/s10589-018-9984-3.

[18] N. I. Gould and V. Simoncini. Error estimates for iterative algorithms for minimizing regularized quadratic subproblems. *Optimization Methods and Software*, 35(2):304–328, 2020. doi: 10.1080/10556788.2019.1670177.

[19] G. N. Grapiglia and Y. Nesterov. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM Journal on Optimization*, 27(1):478–506, 2017. doi: 10.1137/16M1087801.

[20] A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical Report NA/12, 1981.

[21] F. Hanzely, N. Doikov, Y. Nesterov, and P. Richtarik. Stochastic Subspace Cubic Newton method. In *International Conference on Machine Learning*, pages 4027–4038. PMLR, 2020.

[22] M.-J. Lai, Y. Xu, and W. Yin. Improved Iteratively Reweighted Least Squares for Unconstrained Smoothed ℓ_q Minimization. *SIAM Journal on Numerical Analysis*, 51 (2):927–957, 2013. doi: 10.1137/110840364.

[23] Y. Nesterov. Inexact basic tensor methods for some classes of convex optimization problems. *Optimization Methods and Software*, 37(3):878–906, 2022. doi: 10.1080/10556788.2020.1854252.

[24] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006. doi: 10.1007/s10107-006-0706-8.

[25] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR, 2015.

[26] J. Nutini, I. Laradji, and M. Schmidt. Let's Make Block Coordinate Descent Converge Faster: Faster Greedy Rules, Message-Passing, Active-Set Complexity, and Superlinear

Convergence. *Journal of Machine Learning Research*, 23(131):1–74, 2022.

[27] J. K. Pant, W.-S. Lu, and A. Antoniou. New Improved Algorithms for Compressive Sensing Based on ℓ_p Norm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(3):198–202, 2014. doi: 10.1109/TCSII.2013.2296133.

[28] M. Raydan. The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997. doi: 10.1137/S1052623494266365.

[29] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

[30] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009. doi: 10.1007/s10107-007-0170-0.

[31] F. Wen, L. Chu, P. Liu, and R. C. Qiu. A Survey on Nonconvex Regularization-Based Sparse and Low-Rank Recovery in Signal Processing, Statistics, and Machine Learning. *IEEE Access*, 6:69883–69906, 2018. doi: 10.1109/ACCESS.2018.2880454.

[32] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015. doi: 10.1007/s10107-015-0892-3.

[33] J. Zhao, A. Lucchi, and N. Doikov. Cubic regularized subspace Newton for non-convex optimization. *arXiv preprint arXiv:2406.16666*, 2024.