On complexity and convergence of high-order coordinate descent algorithms*

V. S. Amaral† R. Andreani† E. G. Birgin‡ D. S. Marcondes‡
J. M. Martínez†

December 16, 2020

Abstract

Coordinate descent methods with high-order regularized models for box-constrained minimization are introduced. High-order stationarity asymptotic convergence and first-order stationarity worst-case evaluation complexity bounds are established. The computer work that is necessary for obtaining first-order ε-stationarity with respect to the variables of each coordinate-descent block is $O(\varepsilon^{-(p+1)/p})$ whereas the computer work for getting first-order ε-stationarity with respect to all the variables simultaneously is $O(\varepsilon^{-(p+1)})$. Numerical examples involving multidimensional scaling problems are presented. The numerical performance of the methods is enhanced by means of coordinate-descent strategies for choosing initial points.

Key words: Coordinate descent methods, bound-constrained minimization, worst-case evaluation complexity.

AMS subject classifications: 90C30, 65K05, 49M37, 90C60, 68Q25.

1 Introduction

In order to minimize a multivariate function it is natural to keep fixed some of the variables and to modify the remaining ones trying to decrease the objective function value. Coordinate descent (CD) methods proceed systematically in this way and, many times, obtain nice approximations to minimizers of practical optimization problems. Wright [55] surveyed traditional approaches and modern advances on the introduction and analysis of CD methods. Although the CD idea is perhaps the most natural one to optimize functions, it received little attention from researchers

*This work was supported by FAPESP (grants 2013/07375-0, 2016/01860-1, and 2018/24293-0) and CNPq (grants 302538/2019-4 and 302682/2019-8).

†Dept. of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, University of Campinas, 13083-859, Campinas, SP, Brazil. email: vitalianoamaral@hotmail.com, andreani@ime.unicamp.br, martinez@ime.unicamp.br

‡Dept. of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. email: egbirgin@ime.usp.br, diaulas@ime.usp.br

due to poor performance in many cases and lack of challenges in terms of convergence theory [52]. The situation changed dramatically in the last decades. CD methods proved to be useful for solving machine learning, deep learning and statistical learning problems in which the number of variables is big and the accuracy required at the solution is moderate [15, 49]. Many applications arose and, in present days, efficient implementations and insightful theory for understanding the CD properties are the subject of intense research. See, for example, [2, 3, 13, 14, 17, 27, 33, 44, 57] among many others.

In this paper we are concerned with complexity issues of CD methods that employ high-order models to approximate the subproblems that arise at each iteration. The use of high-order models for unconstrained optimization was defined and analyzed from the point of view of worst-case complexity in [6] and other subsequent papers [5, 21, 36, 37, 45, 58]. In [5] numerical implementations with quartic regularization were introduced. In [21], [36], [37], and [45], new high-order regularization methods were introduced with Hölder, instead of Lipschitz, conditions on the highest-order derivatives employed. In [43], high-order methods were studied as discretizations of ordinary differential equations. These methods generalize the methods based on third-order models introduced in [40] and later developed in [19, 20, 30, 32, 50] among many others. Griewank [40] introduced third-order regularization having in mind affine scaling properties. Nesterov and Polyak [50] introduced the first cubic regularized Newton methods with better complexity results than the ones that were known for gradient-like algorithms [38]. In [18], a multilevel strategy that exploits a hierarchy of problems of decreasing dimension was introduced in order to reduce the global cost of the step computation. However, high-order methods remain difficult to implement in the many-variables case due to the necessity of computing high-order derivatives and solving nontrivial model-based subproblems. Nevertheless, if the number of variables is small, high-order model-based methods are reliable alternatives to classical methods. This feature can be obviously exploited in the CD framework.

This paper is organized as follows. In Section 2 we survey a high-order algorithmic framework that provides a basis for the development of CD algorithms. In Section 3, we present block CD methods that, for each approximate minimization on a group of variables, employ high-order regularized subproblems and we prove asymptotic convergence. In Section 4 we prove worst-case complexity results. In Section 5 the obtained theoretical results are discussed. In Section 6, we study a family of problems for which CD is suitable and we include a CD-strategy that improves convergence to global solutions. Conclusions are given in Section 7.

Notation. The symbol $\|\cdot\|$ denotes the Euclidean norm.

2 Preliminaries

In this section, we consider the problem

$$\text{Minimize } \underline{f}(x) \text{ subject to } x \in \underline{\Omega}, \tag{1}$$

where $\underline{\Omega} \subset \mathbb{R}^{\underline{n}}$ is given by

$$\underline{\Omega} = \{x \in \mathbb{R}^{\underline{n}} \mid \underline{\ell} \leq x \leq \underline{u}\} \tag{2}$$

and $\underline{\ell} < \underline{u}$. We assume that $\underline{f}$ has continuous first derivatives into $\underline{\Omega}$. We denote $\underline{g}(x) = \nabla \underline{f}(x)$ and $\underline{g}_P(x) = P_{\underline{\Omega}}(x - \underline{g}(x)) - x$, for all $x \in \underline{\Omega}$, where $P_{\underline{\Omega}}$ is the Euclidean projection operator onto $\underline{\Omega}$. In the remaining of this section, the results from [8] that are relevant to the present work are surveyed and a natural extension of the main algorithm in [8], that makes it possible a wider class of models to be considered, is introduced.

Each iteration k of Algorithm 2.1 introduced in [8] computes a new iterate x^{k+1} satisfying $(p+1)$th-order descent with respect to $\underline{f}(x^k)$ through the approximate minimization of a $(p+1)$th-regularized pth-order model of the function $\underline{f}$ around the iterate x^k. For all $\bar{x} \in \mathbb{R}^n$, let $\underline{M}_{\bar{x}} : \mathbb{R}^n \to \mathbb{R}$ be a "model" of $\underline{f}(x)$ around $\bar{x}$; and assume that $\nabla \underline{M}_{\bar{x}}(x)$ exists for all $x \in \Omega$. We now present an algorithm that corresponds to a single iteration of the algorithm introduced in [8].

Algorithm 2.1. Assume that $p \in \{1, 2, 3, \dots\}$, $\alpha > 0$, $\sigma_{\min} > 0$, $\tau_2 \geq \tau_1 > 1$, $\theta > 0$, and $\bar{x} \in \underline{\Omega}$ are given.

Step 1. Set $\sigma \leftarrow 0$.

Step 2. Compute $x^{\text{trial}} \in \underline{\Omega}$ such that

$$\underline{M}_{\bar{x}}(x^{\text{trial}}) + \sigma \|x^{\text{trial}} - \bar{x}\|^{p+1} \leq \underline{M}_{\bar{x}}(\bar{x}) \tag{3}$$

and

$$\left\| P_{\underline{\Omega}} \left[x^{\text{trial}} - \nabla \left(\underline{M}_{\bar{x}}(x) + \sigma \|x - \bar{x}\|^{p+1} \right)\big|_{x = x^{\text{trial}}} \right] - x^{\text{trial}} \right\| \leq \theta \|x^{\text{trial}} - \bar{x}\|^p. \tag{4}$$

Step 3. If

$$\underline{f}(x^{\text{trial}}) \leq \underline{f}(\bar{x}) - \alpha \|x^{\text{trial}} - \bar{x}\|^{p+1}, \tag{5}$$

then define $x^+ = x^{\text{trial}}$ and stop. Otherwise, update $\sigma \leftarrow \max\{\sigma_{\min}, \tau\sigma\}$ with $\tau \in [\tau_1, \tau_2]$ and go to Step 2.

Remark. The trial point x^{trial} computed at Step 2 is intended to be an approximate solution to the subproblem

$$\text{Minimize } \underline{M}_{\bar{x}}(x) + \sigma \|x - \bar{x}\|^{p+1} \text{ subject to } x \in \underline{\Omega}. \tag{6}$$

Note that conditions (3) and (4) can always be achieved. In fact, by the compactness of $\underline{\Omega}$, if x^{trial} is a global minimizer of (6), then it satisfies the condition

$$\left\| P_{\underline{\Omega}}[x^{\text{trial}} - \nabla(\underline{M}_{\bar{x}}(x) + \sigma \|x - \bar{x}\|^{p+1})\big|_{x = x^{\text{trial}}}] - x^{\text{trial}} \right\| = 0;$$

and so (4) takes place. In addition, if x^{trial} is a global minimizer, since $\bar{x}$ is a feasible point, (3) must hold as well.

Assumption A1 *There exists $L > 0$ such that, for all x^{trial} computed by Algorithm 2.1, $x = x^{\text{trial}}$ satisfies*

$$\left\| \underline{g}(x) - \nabla \underline{M}_{\bar{x}}(x) \right\| \leq L \|x - \bar{x}\|^p, \tag{7}$$

$$\underline{M}_{\bar{x}}(\bar{x}) = \underline{f}(\bar{x}) \text{ and } \underline{f}(x) \leq \underline{M}_{\bar{x}}(x) + L \|x - \bar{x}\|^{p+1}. \tag{8}$$

If $\underline{f} : \mathbb{R}^{\underline{n}} \to \mathbb{R}$ admits continuous derivatives up to order $p \in \{1, 2, 3, \dots\}$, then the Taylor polynomial of order p of $\underline{f}$ around $\bar{x}$ can be written in the form

$$\underline{T}_p(\bar{x}, x) = \underline{f}(\bar{x}) + \sum_{j=1}^{p} \underline{P}_j(\bar{x}, x), \tag{9}$$

where $\underline{P}_j(\bar{x}, x)$ is an homogeneous polynomial of degree j given by

$$\underline{P}_j(\bar{x}, x) = \frac{1}{j!} \left((x_1 - \bar{x}_1) \frac{\partial}{\partial x_1} + \cdots + (x_n - \bar{x}_n) \frac{\partial}{\partial x_n} \right)^j \underline{f}(x). \tag{10}$$

For completeness we define $\underline{P}_0(\bar{x}, x) = f(\bar{x})$.

If $\underline{M}_{\bar{x}}(x)$ is the Taylor polynomial of order p of $\underline{f}$ around $\bar{x}$ and the pth-order derivatives of $\underline{f}$ satisfy a Lipschitz condition with Lipschitz constant L, then Assumption A1 is satisfied. However, the situations in which Assumption A1 holds are not restricted to the case in which $\underline{M}_{\bar{x}}(x) = \underline{T}_p(\bar{x}, x)$. For example, we may choose $\underline{M}_{\bar{x}}(x) = \underline{f}(x)$. (Note that, in this case, p may be arbitrarily large but only first derivatives of $\underline{f}(x)$ need to exist.) Although the results in [8] only mention the choice $\underline{M}_{\bar{x}}(x) = \underline{T}_p(\bar{x}, x)$, these results only depend on Assumption A1. Thus, they can be trivially extended to the general choice of $\underline{M}_{\bar{x}}(x)$.

Theorem 2.1 *Suppose that Assumption A1 holds. If the regularization parameter σ in (3) satisfies $\sigma \geq L + \alpha$, then the trial point x^{trial} satisfies the sufficient descent condition (5). Moreover,*

$$\left\| \underline{g}_P(x^+) \right\| \leq \left(L + \tau_2 \left(L + \alpha \right) (p + 1) + \theta \right) \|x^+ - \bar{x}\|^p \tag{11}$$

and

$$\underline{f}(x^+) \leq \underline{f}(\bar{x}) - \alpha \left(\frac{\|\underline{g}_P(x^+)\|}{L + \tau_2 \left(L + \alpha \right) (p + 1) + \theta} \right)^{(p+1)/p}. \tag{12}$$

Proof: This theorem condensates the results in [8, Lemmas 3.2–3.4]. □

Theorem 2.1 justifies the definition of an algorithm for solving (1) based on repetitive application of Algorithm 2.1 and shows that such algorithm enjoys good properties in terms of convergence and complexity. On the one hand, each iteration of the algorithm requires $O(1)$ functional evaluations and finishes satisfying a suitable sufficient descent condition. On the other hand, that condition implies that infinitely many iterations with gradient-norm bounded away from zero are not possible if the function is bounded below. Moreover, (12) provides a complexity bound on the number of iterations based on the norm of the projected gradient. In the following sections, we prove that, thanks to Theorem 2.1, similar convergence and evaluation complexity properties holds for a coordinate descent algorithm.

3 High-order coordinate descent algorithm

In this section, we consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in \Omega, \tag{13}$$

where $\Omega \subset \mathbb{R}^n$ is given by

$$\Omega = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\} \tag{14}$$

and $\ell < u$. We assume that f has continuous first derivatives onto Ω.

At each iteration of the coordinate descent method introduced in this section for solving (13), (i) a nonempty set of indices $I_k \subseteq \{1, \ldots, n\}$ is selected, (ii) coordinates corresponding to indices that are not in I_k remain fixed, and (iii) Algorithm 2.1 is applied to the minimization of f over Ω with respect to the free variables, i.e. variables with indices in I_k. From now on, given $v \in \mathbb{R}^n$, we denote by $v_I \in \mathbb{R}^{|I|}$ the vector whose components are the components of v whose indices belong to $I \subseteq \{1, \ldots, n\}$. For all $x \in \Omega$, we define $g_{P,I}(x) \in \mathbb{R}^n$ by

$$[g_{P,I}(x)]_i = \begin{cases} [g_P(x)]_i, & \text{if } i \in I, \\ 0, & \text{if } i \notin I. \end{cases}$$

Since Ω is a box, this definition is equivalent to $g_{P,I}(x) = P_\Omega(x - g_I(x)) - x$, where

$$[g_I(x)]_i = \begin{cases} [g(x)]_i, & \text{if } i \in I, \\ 0, & \text{if } i \notin I. \end{cases}$$

This equivalence, that will be used in the theoretical convergence results below, is not true if Ω is an arbitrary closed and convex set. This is the reason for which we consider CD algorithms only with box constraints.

Algorithm 3.1. Assume that $p \in \{1, 2, 3, \ldots\}$, $\alpha > 0$, $\sigma_{\min} > 0$, $\tau_2 \geq \tau_1 > 1$, $\theta > 0$, and $x^0 \in \Omega$ are given. Initialize $k \leftarrow 0$.

Step 1. Choose a nonempty set $I_k \subseteq \{1, \ldots, n\}$.

Step 2. Consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in \Omega \text{ and } x_i = x_i^k \text{ for all } i \notin I_k. \tag{15}$$

Let $\bar{x} = x^k_{I_k}$. Setting $\underline{f}$, $\underline{\Omega}$, and $\underline{M}_{\bar{x}}$ properly, apply Algorithm 2.1 to obtain x^+.

Step 3. Define x^{k+1} as $x^{k+1}_{I_k} = x^+$ and $x^{k+1}_i = x^k_i$ for all $i \notin I_k$, set $k \leftarrow k+1$, and go to Step 1.

Assumption A2 *There exists $L > 0$ such that for all k, $\bar{x}$, $\underline{f}$, and $\underline{M}_{\bar{x}}$ set at the kth iteration of Algorithm 3.1 and for all x^{trial} computed by Algorithm 2.1 when called at the kth iteration of Algorithm 3.1, (7) and (8) take place with $x = x^{\text{trial}}$.*

Theorem 3.1 *Suppose that Assumption A2 holds. Then, there exists $c > 0$, which only depends on L, τ_2, α, p, and θ such that, for all $k = 0, 1, 2, \ldots$, the point x^{k+1} computed by Algorithm 3.1 is well defined and satisfies*

$$f(x^{k+1}) \leq f(x^k) - \alpha \|x^{k+1} - x^k\|^{p+1} \tag{16}$$

and

$$\left\| g_{P,I_k}(x^{k+1}) \right\| \leq c \|x^{k+1} - x^k\|^p. \tag{17}$$

Proof: The proof follows from the application of Theorem 2.1. □

Theorem 3.2 *Suppose that Assumption A2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3.1. Then,*

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0, \tag{18}$$

$$\lim_{k\to\infty} \left\| g_{P,I_k}(x^{k+1}) \right\| = 0, \tag{19}$$

and

$$\lim_{k\to\infty} \left\| g_{P,I_k}(x^{k}) \right\| = 0. \tag{20}$$

Proof: Since Ω is compact, we have that f is bounded below onto Ω. Thus, (18) follows from (16) and, in consequence, (19) follows from (18) and (17). Let us prove (20). Assume that $I \subseteq \{1,\dots,n\}$ is nonempty and arbitrary. By the continuity of the gradient, the function $\|g_{P,I}(x)\|$ is continuous for all $x \in \Omega$ and, since Ω is compact, it is uniformly continuous. Then, given $\varepsilon > 0$, there exists $\delta_I > 0$ such that, whenever $\|x - y\| \leq \delta_I$, we have that $\|g_{P,I}(x) - g_{P,I}(y)\| \leq \varepsilon/2$. Since the number of different subsets of $\{1,\dots,n\}$ is finite, we have that $\delta \equiv \min\{\delta_I \mid \emptyset \neq I \subseteq \{1,\dots,n\}\} > 0$. Thus, for all $I \subseteq \{1,\dots,n\}$, if $\|x - y\| \leq \delta$, we have that $\|g_{P,I}(x) - g_{P,I}(y)\| \leq \varepsilon/2$. Now, by (18), there exists k_0 such that, whenever $k \geq k_0$, we have that $\|x^{k+1} - x^k\| \leq \delta$. Then, by the definition of δ, if $k \geq k_0$, $\|g_{P,I}(x^{k+1}) - g_{P,I}(x^k)\| \leq \varepsilon/2$ for all nonempty $I \subseteq \{1,\dots,n\}$. In particular, taking $I = I_k$, if $k \geq k_0$, we have that $\|g_{P,I_k}(x^{k+1}) - g_{P,I_k}(x^k)\| \leq \varepsilon/2$. Finally, by (19), there exists $k_1 \geq k_0$ such that, for all $k \geq k_1$, $\|g_{P,I_k}(x^{k+1})\| \leq \varepsilon/2$. By the triangular inequality, adding the last two inequalities we have that $\|g_{P,I_k}(x^k)\| \leq \varepsilon$. Since $\varepsilon > 0$ was arbitrary, this completes the proof of (20). □

The following assumption guarantees that all the indices $i \in \{1,\dots,n\}$ belong to some I_k at least every $\bar{m}$ iterations. This guarantees that the CD method tries to reduce the function with respect to each variable x_i infinitely many times.

Assumption A3 *There exists $\bar{m} < +\infty$ such that, for all $i \in \{1,\dots,n\}$:*

1. *There exists $k \leq \bar{m}$ such that $i \in I_k$;*
2. *If $i \in I_k$, then there exists $m \leq \bar{m}$ such that $i \in I_{k+m}$.*

Theorem 3.3 *Suppose Assumptions A2 and A3 hold. Let $\{x^k\}$ be the sequence generated by Algorithm 3.1. Then,*

$$\lim_{k\to\infty} \|g_P(x^k)\| = 0. \tag{21}$$

Moreover, if $x^* \in \Omega$ is a limit point of $\{x^k\}$, then we have that $\|g_P(x^*)\| = 0$.

Proof: Let $i \in \{1,\dots,n\}$. By Assumption A3, there exists an infinite set of increasing indices $K = \{k_1, k_2, k_3, \dots\}$ such that $i \in I_{k_\ell}$ and $k_{\ell+1} \leq k_\ell + \bar{m}$ for all $\ell = 1, 2, 3, \dots$ Then, by (20) in Theorem 3.2, since, by definition, given $I \subseteq \{1,\dots,n\}$, $[g_{P,I}(x)]_i = [g_P(x)]_i$ for any $i \in I$,

$$\lim_{k\in K} [g_P(x^k)]_i = 0. \tag{22}$$

Let $j \in \{1, 2, \dots\}$ be arbitrary. By (18), the triangular inequality, and the uniform continuity of g_P, we have that

$$\lim_{k \in K} |[g_P(x^{k+j})]_i - [g_P(x^k)]_i| = 0.$$

Therefore, by (22),

$$\lim_{k \in K} [g_P(x^{k+j})]_i = 0. \tag{23}$$

In particular, (23) holds for all $j = 1, \dots, \bar{m}$. This implies that

$$\lim_{k \to \infty} [g_P(x^k)]_i = 0. \tag{24}$$

Thus, the thesis is proved. □

Theorem 3.3 shows that limit points of sequences generated by Algorithm 3.1 are first-order stationary. The rest of this section is dedicated to prove that, under suitable conditions, pth-order stationarity with respect to each variable also holds. More precisely, if the same nonempty set I_k is repeated infinitely many times, p-stationarity holds in the limit for the variables x_i with $i \in I_k$. For this purpose, we need to define different notions of stationarity. We begin with functions of a single variable. Although the corresponding optimality conditions are, of course, well known, we find it useful to state these conditions with a strong reference to the Taylor polynomial, stressing a property that holds only in the case of univariate functions.

Theorem 3.4 *Assume that $\varphi : \mathbb{R} \to \mathbb{R}$ is continuous and its derivatives up to order p are continuous for all x in the convex set $\underline{\Omega}$. Let $x^* \in \underline{\Omega}$ be a local minimizer of $\varphi(x)$ subject to $x \in \underline{\Omega}$. Then x^* is also a local minimizer of its Taylor polynomial of order p around x^* subject to $x \in \underline{\Omega}$.*

The key argument for proving Theorem 3.4 is that, in one variable, only the null polynomial can vanish infinitely many times. Of course, this property is not true for polynomials of $\underline{n} > 1$ variables. This is the reason why the thesis of Theorem 3.4 is not true if $\underline{n} > 1$. In fact, in the multivariate case $\underline{f} : \mathbb{R}^{\underline{n}} \to \mathbb{R}$, a local minimizer x^* may not be a local minimizer of its Taylor polynomial of order $p > 2$. For example, if $\underline{f}(x_1, x_2) = x_2^2 - x_1^2 x_2 + x_1^4$, we have that $(0, 0)$ is a global minimizer of $\underline{f}$, but it is not a local minimizer of its Taylor polynomial of order $p = 3$. Therefore, in the $\underline{n}$-dimensional case, although motivated by the case $\underline{n} = 1$, the definition of p-stationarity does not correspond to local minimization of the Taylor polynomial.

Definition 3.1 *We say that x^* is pth-order stationary of $\underline{f}$ over the convex set $\underline{\Omega}$ if, for all $x \in \underline{\Omega}$, 0 is a local minimizer of the Taylor polynomial of order p that corresponds to the univariate function $\varphi(t) = \underline{f}(x^* + t(x - x^*))$ restricted to the constraint $x^* + t(x - x^*) \in \underline{\Omega}$.*

If the derivatives of $\underline{f}$ up to order p are continuous, local minimizers are necessarily pth-order stationary according to Definition 3.1. The following theorem motivates a stronger definition of p-stationarity that holds when a Lipschitz condition is satisfied.

Theorem 3.5 *Assume that $\mathcal{D} \subset \mathbb{R}^{\underline{n}}$, $\underline{f} : \mathcal{D} \to \mathbb{R}$, and x^* is a local minimizer of $\underline{f}(x)$ over $\mathcal{D}$ such that, for all $x \in \mathcal{D}$,*

$$\underline{f}(x) \leq \underline{T}_p(x^*, x) + L\|x - x^*\|^{p+1}, \tag{25}$$

where $\underline{T}_p$ is the Taylor polynomial of $\underline{f}$. Then, for all $\sigma \geq L$, x^* is a local minimizer of $\underline{T}_p(x^*, x) + \sigma\|x - x^*\|^{p+1}$ over $\mathcal{D}$.

Proof: Suppose that the thesis is not true. Then, x^* is not a local minimizer of $\underline{T}_p(x^*, x) + L\|x - x^*\|^{p+1}$ over $\mathcal{D}$. Thus, there exists $\{x^k\} \subset \mathcal{D}$ such that $\lim_{k\to\infty} x^k = x^*$ and

$$\underline{T}_p(x^*, x^k) + L\|x^k - x^*\|^{p+1} < \underline{T}_p(x^*, x^*) = \underline{f}(x^*).$$

Thus, by (25),

$$\underline{f}(x^k) < \underline{f}(x^*)$$

for all $k = 0, 1, 2, \ldots$ This contradicts the fact that x^* is a local minimizer of $\underline{f}$ over $\mathcal{D}$. □

Definition 3.2 *Assume that $\mathcal{D} \subset \mathbb{R}^{\underline{n}}$, $\underline{f} : \mathcal{D} \to \mathbb{R}$, x^* is such that (25) holds for all $x \in \mathcal{D}$, and that $\sigma \geq L$. Then $x^* \in \mathcal{D}$ is said to be pth-order σ-stationary of $\underline{f}$ over $\mathcal{D}$ if x^* is a local minimizer of $\underline{T}_p(x^*, x) + \sigma\|x - x^*\|^{p+1}$ over $\mathcal{D}$.*

It is trivial to see that, if $\mathcal{D}$ is convex and x^* is pth-order σ-stationary of $\underline{f}$ over $\mathcal{D}$ according to Definition 3.2, then it is pth-order $\tilde{\sigma}$-stationary for every $\tilde{\sigma} \geq \sigma$ and it is also pth-order stationary according to Definition 3.1. However, pth-order L-stationarity is strictly stronger than pth-order stationarity. Consider the function $\underline{f}(x_1, x_2) = x_2^2 - x_1^2 x_2$ and $p = 3$. Note that $x^* = (0, 0)$ satisfies (25) with $L = 0$. Straightforward calculations show that that the point $(0, 0)$, that is not a local minimizer of $\underline{f}$, is pth-order stationary according to Definition 3.1. On the other hand, it is easy to see that $(0, 0)$ is not pth-order σ-stationarity if $\sigma < 1/4$. See Figure 1.

In Theorem 3.3 we proved that Algorithm 3.1 is satisfactory from the point of view of first-order stationarity. In the CD approach we cannot advocate for full stationarity of high order because cross derivatives that involve variables that are never optimized together are not computed at all. However, if optimization with respect to the same group of variables occurs at infinitely many iterations, it is reasonable to conjecture that high-order optimality with respect to those variables would, in the limit, take place. For obtaining such result, it is not enough to satisfy criteria (3) and (4) when solving subproblems. The reason is that (4) takes into account only first-order optimality of problem (6). A stronger assumption on the subproblem solution is made in the following theorem. Namely, it is assumed that, in addition to (4), a global solution to subproblem (6) is computed. This assumption could be rather mild in the case that all the subproblems are chosen to be small dimensional. In this case, it is possible to prove that, in the limit, suitable pth-order optimality conditions are satisfied. Observe that partial derivatives that are not necessary for computing Taylor approximations are not assumed to exist at all, let alone to be continuous.

Theorem 3.6 *Suppose that Assumption A2 holds and the sequence $\{x^k\}$ is generated by Algorithm 3.1. Suppose that, at iteration k, the function $\underline{f}$ has as variables x_i with $i \in I_k$, $\underline{\Omega}$ is the*

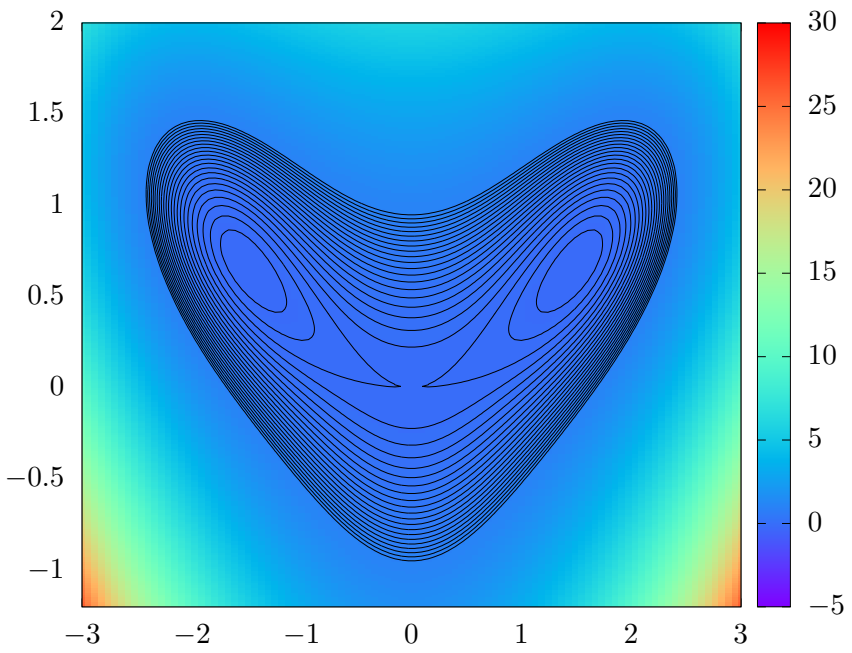

Figure 1: Level sets of $\underline{T}_p((0,0),(x_1,x_2)) + \sigma\|(x_1,x_2)-(0,0)\|^{(p+1)}$ with $p=3$ and $\sigma = 0.125$, where $\underline{T}_p((0,0),(x_1,x_2))$ is the pth-order Taylor polynomial of $\underline{f}(x_1,x_2) = x_2^2 - x_1^2x_2$ (that coincides with $\underline{f}$). The graphic shows that the pth-order σ-stationarity condition with $p=3$ and $\sigma = 0.125$ does not hold at $(0,0)$, since it is *not* a local minimizer of the regularized pth-order Taylor polynomial. There are two local minimizers at "the eyes of the cat".

box Ω restricted to the variables $i \in I_k$, $\underline{M}_{\bar{x}}(x)$ is chosen as the pth-order Taylor polynomial of $\underline{f}$ defined in (9), the derivatives involved in (9) exist and are continuous for all $x \in \Omega$, and Algorithm 2.1 computes x^+ as a global minimizer of (6). Let I be a set of indices such that $I = I_k$ at infinitely many iterations $k \in K$. Let x^* be a limit point of the sequence $\{x^k\}_{k\in K}$. Then, for all $j \le p$, x^* is jth-order stationary for problem (1) according to Definition 3.1 and it is also jth-order σ-stationary for some $\sigma \le \tau_2(L+\alpha)$ according to Definition 3.2 of problem (1).

Proof: Consider the problem

$$\text{Minimize } T_p(x^*,x) + \sigma\|x-x^*\|^{p+1} \text{ subject to } x \in \Omega \text{ and } x_i = x_i^* \text{ for all } i \notin I. \tag{26}$$

By the hypothesis, for all $k \in K$, x^+ is obtained as a global minimizer of

$$\text{Minimize } T_p(x^k,x) + \sigma\|x-x^k\|^{p+1} \text{ subject to } x \in \Omega \text{ and } x_i = x_i^k \text{ for all } i \notin I, \tag{27}$$

for some $\sigma > 0$. Then, by Theorem 2.1, x^{k+1} is a global minimizer of (27) with $\sigma = \sigma_k \le \tau_2(L+\alpha)$. By (18), $\lim_{k\in K} x^{k+1} = \lim_{k\in K} x^k = x^*$. Taking a convenient subsequence, assume, without loss of generality, that $\lim_{k\in K}\sigma_k = \sigma_* \le \tau_2(L+\alpha)$. Let $x \in \Omega$ be such that $x_i = x_i^*$ for

all $i \notin I$. Let $z^k \in \Omega$ be such that $z_i^k = x_i$ for all $i \in I$ and $z_i^k = x_i^k$ for all $i \notin I$. Then, by the definition of x^{k+1}, for all $k \in K$,

$$T_p(x^k, x^{k+1}) + \sigma_k \|x^{k+1} - x^k\|^{p+1} \leq T_p(x^k, z^k) + \sigma_k \|z^k - x^k\|^{p+1}. \tag{28}$$

Taking limits for $k \in K$, by the definition of z^k, we have that

$$T_p(x^*, x^*) + \sigma_* \|x^* - x^*\|^{p+1} \leq T_p(x^*, x) + \sigma_* \|x - x^k\|^{p+1}. \tag{29}$$

Since x was arbitrary, this implies that x^* is a global solution of (26). Consequently, x^* is also a local solution of (26). Since the Taylor polynomial of order p of $T_p(x^*, x) + \sigma_* \|x - x^*\|^{p+1}$ coincides with the Taylor polynomial of order p of $\underline{f}$, the thesis is proved. □

Remark. Note that the hypothesis of Theorem 3.6 implies a stronger thesis than the one stated. In fact, we proved that, in the limit, each partial Taylor polynomial has a global minimizer. This is interesting because that fact is not an optimality condition. In other words, the algorithm does not converge to points that satisfy the optimality conditions, unless the stronger "global" property is also satisfied. However, this could be an advantage if one is essentially interested in finding global minimizers.

Corollary 3.1 *Consider the assumptions of Theorem 3.6 and assume that, for all k,*

$$I_k = \{\text{mod}(k, n) + 1\}.$$

If x^* is a limit point of the sequence generated by Algorithm 3.1, then for all $i = 1, \ldots, n$, x_i^* is a jth-order stationary point of the problem

$$\text{Minimize } f(x_1^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, \ldots, x_n^*) \text{ subject to } \ell_i \leq x_i \leq u_i \tag{30}$$

for all $j \leq p$.

Proof: The proof is a direct application of Theorem 3.6. □

4 Complexity

Given a tolerance $\varepsilon > 0$, we wish to know the worst possible computer effort that we need to obtain an iterate x at which the objective function is smaller than a given target or the projected gradient norm $\|g_P(x)\|$ is smaller than ε. We show that the number of iterations that are needed to obtain $|[\|g_P(x^{k+1})]_i| \leq \varepsilon$ for all $i \in I_k$ is, at most, a constant times $\varepsilon^{-(p+1)/p}$ as in typical high-order methods. However, obtaining $|[\|g_P(x^{k+1})]_i| \leq \varepsilon$ for all $i \notin I_k$ is harder as, for this purpose, we need that consecutive iterations be close enough. This difficulty is intrinsic to coordinate descent methods. Powell's example of non-convergence of CD methods [52] satisfies the requirement $|[\|g_P(x^{k+1})]_i| \leq \varepsilon$ for all $i \in I_k$ at every iteration but never satisfies $|[\|g_P(x^{k+1})]_i| \leq \varepsilon$ for $i \notin I_k$. Our method converges even in Powell's example because we require sufficient descent based on regularization but it is affected by Powell's effect because the number of iterations at which the distance between consecutive iterates is bigger than a fixed distance grows with the order p. Then, it is not surprising that our worst-case complexity bound be significantly worse than $O(\varepsilon^{-(p+1)/p})$. These results are rigorously proved in this section and discussed in Section 5.

Theorem 4.1 *Suppose that Assumptions A2 holds. Let* $f_{\text{target}} \leq f(x^0)$ *and* $\varepsilon > 0$ *be given. Then, the quantity of iterations* k *such that*

(i) $f(x^{k+1}) > f_{\text{target}}$ *and*

(ii) $|[g_P(x^{k+1})]_i| > \varepsilon$ *for some* $i \in I_k$

is bounded by

$$\frac{f(x^0) - f_{\text{target}}}{c\,\varepsilon^{(p+1)/p}}, \tag{31}$$

where c *only depends on* α, τ_2, L, p, *and* θ.

Proof: By (12) in Theorem 2.1,

$$f(x^{k+1}) \leq f(x^k) - c\|g_{P,I_k}(x^{k+1})\|^{(p+1)/p},$$

where $c = (\alpha/(L + \tau_2(L+\alpha)(p+1) + \theta))^{(p+1)/p}$. Therefore, if $i \in I_k$,

$$f(x^{k+1}) \leq f(x^k) - c\left|[g_P(x^{k+1})]_i\right|^{(p+1)/p}.$$

So, if $\left|[g_P(x^{k+1})]_i\right| > \varepsilon$,

$$f(x^{k+1}) \leq f(x^k) - c\varepsilon^{(p+1)/p}. \tag{32}$$

Since the sequence $\{f(x^k)\}$ decreases monotonically, the number of iterations at which (32) occurs together with $f(x^{k+1}) > f_{\text{target}}$ cannot exceed $(f(x^0) - f_{\text{target}})/(c\varepsilon^{(p+1)/p})$. This completes the proof. □

Theorem 4.2 *Suppose that AssumptionsA2 holds. Let* $f_{\text{target}} \leq f(x^0)$ *and* $\delta > 0$ *be given. Then, the quantity of iterations* k *such that* $f(x^k) > f_{\text{target}}$ *and* $\|x^{k+1} - x^k\| > \delta$ *is bounded by*

$$\frac{f(x^0) - f_{\text{target}}}{\alpha\,\delta^{p+1}}. \tag{33}$$

Proof: The proof follows directly from (16) in Theorem 3.1. □

Theorem 4.3 *Suppose that AssumptionsA2 holds. Let* $f_{\text{target}} \leq f(x^0)$, $\varepsilon > 0$, *and* $\delta > 0$ *be given. Then, the quantity of iterations* k *such that*

(i) $f(x^{k+1}) > f_{\text{target}}$ *and*

(ii) $\|x^{k+1} - x^k\| > \delta$ *or* $|[g_P(x^{k+1})]_i| > \varepsilon$ *for some* $i \in I_k$

is bounded by

$$\frac{f(x^0) - f_{\text{target}}}{c\,\varepsilon^{(p+1)/p}} + \frac{f(x^0) - f_{\text{target}}}{\alpha\,\delta^{p+1}}, \tag{34}$$

where c *only depends on* α, τ_2, L, p, *and* θ.

Proof: The proof follows directly from Theorems 4.1 and 4.2. □

We now divide the iterations of Algorithm 3.1 in *cycles*. Each cycle is composed by $\bar{m}$ iterations, where $\bar{m}$ is the one assumed to exist in Assumption A3. Therefore, the successive cycles start at iterations $x^0, x^{\bar{m}}, x^{2\bar{m}}, \ldots, x^{\ell\bar{m}}, \ldots$ The iterates $x^{\ell\bar{m}+1}, \ldots, x^{\ell\bar{m}+\bar{m}}$ are said to be *produced* at cycle ℓ. Iterations $k = \ell\bar{m}, \ldots, \ell\bar{m} + \bar{m} - 1$, at which these iterates were produced, are said to be *internal* iterations of cyle ℓ. Each iteration k is associated with a set of indices I_k. Due to Assumption A3, for every coordinate $i = 1, \ldots, n$ and every cycle $\ell \geq 0$, there is at least an iteration k internal to cycle ℓ such that $i \in I_k$. In other words, all coordinates are considered in at least an iteration of every cycle. With the notion of cycle at hand, we can now restate Theorems 4.1, 4.2, and 4.3 as follows.

Theorem 4.4 *Suppose that Assumptions A2 and A3 hold. Let $f_{\text{target}} \leq f(x^0)$ and $\varepsilon > 0$ be given. Then, the quantity of cycles ℓ that contain an internal iteration k such that*

(i) *$f(x^{k+1}) > f_{\text{target}}$ and*

(ii) *$|[g_P(x^{k+1})]_i| > \varepsilon$ for some $i \in I_k$*

is not larger than

$$\frac{f(x^0) - f_{\text{target}}}{c\,\varepsilon^{(p+1)/p}}, \tag{35}$$

where c only depends on α, τ_2, L, p, and θ.

Proof: Let ℓ be a cycle that contains an internal iteration k satisfying (i) and (ii). By Theorem 4.1, the quantity of this type of iteration is bounded by (35); and so the same bound applies to the quantity of cycles containing an iterations with these properties. This completes the proof. □

Theorem 4.5 *Suppose that Assumptions A2 and A3 hold. Let $f_{\text{target}} \leq f(x^0)$ and $\varepsilon > 0$ be given. Then, the quantity of cycles ℓ that contain an internal iteration k such that $f(x^k) > f_{\text{target}}$ and $\|x^{k+1} - x^k\| > \delta$ is bounded by*

$$\frac{f(x^0) - f_{\text{target}}}{\alpha\,\delta^{p+1}}. \tag{36}$$

Proof: Let ℓ be a cycle that contains an internal iteration k such that $f(x^k) > f_{\text{target}}$ and $\|x^{k+1} - x^k\| > \delta$. By Theorem 4.2, the quantity of this type of iteration is bounded by (36); and so the same bound applies to the quantity of cycles containing an iterations with these properties. This completes the proof. □

Theorem 4.6 *Suppose that Assumptions A2 and A3 hold. Let $f_{\text{target}} \leq f(x^0)$, $\varepsilon > 0$, and $\delta > 0$ be given. Then, the quantity of cycles ℓ that contain an internal iteration k such that*

(i) *$f(x^{k+1}) > f_{\text{target}}$ and*

(ii) *$\|x^{k+1} - x^k\| > \delta$ or $|[g_P(x^{k+1})]_i| > \varepsilon$ for some $i \in I_k$*

is bounded by

$$\frac{f(x^0) - f_{\text{target}}}{c\,\varepsilon^{(p+1)/p}} + \frac{f(x^0) - f_{\text{target}}}{\alpha\,\delta^{p+1}}, \tag{37}$$

where c only depends on α, τ_2, L, p, and θ.

Proof: The proof follows directly from Theorems 4.4 and 4.5. □

The following assumption guarantees that small increments cause small differences on the projected gradients.

Assumption A4 *There exists $L_g > 0$ such that for all $i = 1, \ldots, n$ and $x, z \in \Omega$,*

$$|[g_P(x)]_i - [g_P(z)]_i| \leq L_g \|x - z\|. \tag{38}$$

With the tools given by Assumption A4 and Theorem 4.6, we are now able to establish a bound on the number of cycles at which the whole projected gradient is bigger than a given tolerance.

Theorem 4.7 *Suppose that Assumptions A2, A3, and A4 hold. Let $f_{\text{target}} \leq f(x^0)$, $\varepsilon > 0$, and $\delta > 0$ be given. Then, there exists a cycle ℓ, with ℓ exceeding (37) by one in the worst case, such that either*

(i) *for some iteration k internal to cycle ℓ, we have that $f(x^k) \leq f_{\text{target}}$ or*

(ii) *for all the iterations k internal to cycle ℓ we have that*

$$\left|[g_P(x^{k+1})]_i\right| \leq \varepsilon + \bar{m} L_g \delta \text{ for all } i = 1, \ldots, n. \tag{39}$$

Proof: By Theorem 4.6, there exists a cycle ℓ that does not exceeds (37) by more than one such that, for each iterations k internal to cycle ℓ, either $f(x^{k+1}) \leq f_{\text{target}}$ or

$$\|x^{k+1} - x^k\| \leq \delta \text{ and } |[g_P(x^{k+1})]_i| \leq \varepsilon \text{ for all } i \in I_k. \tag{40}$$

If there exists an iteration k internal to cycle ℓ such that $f(x^{k+1}) \leq f_{\text{target}}$, then we are done. So, we assume that, for all iterations k internal to cycle ℓ, (40) holds. Let $i \in \{1, \ldots, n\}$ be arbitrary. Assumption A3 implies that there is an iteration k internal to cycle ℓ such that $i \in I_k$ and, thus, by (40), $|[g_P(x^{k+1})]_i| \leq \varepsilon$. For any other iterate z produced at cycle ℓ, by Assumption A4, the triangle inequality, and the first inequality in (40), we have that

$$|[g_P(z)]_i - [g_P(x^{k+1})]_i| \leq L_g \|z - x^{k+1}\| \leq \bar{m} L_g \delta.$$

Thus,

$$|[g_P(z)]_i| \leq \varepsilon + \bar{m} L_g \delta,$$

as we wanted to prove. □

Theorem 4.8 *Suppose that Assumptions A2, A3, and A4 hold. Let* $f_{\text{target}} \leq f(x^0)$, $\varepsilon > 0$, *and* $\delta > 0$ *be given. Then, there exists a cycle* ℓ *not larger than*

$$\frac{f(x^0) - f_{\text{target}}}{c\,(\varepsilon/2)^{(p+1)/p}} + \frac{f(x^0) - f_{\text{target}}}{\alpha\,(\varepsilon/(2\bar{m}L_g)^{p+1}} + 1, \tag{41}$$

where c *only depends on* α, τ_2, L, p, *and* θ, *such that, in its first internal iteration* k, *either* $f(x^k) \leq f_{\text{target}}$ *or*

$$\left|[g_P(x^{k+1})]_i\right| \leq \varepsilon \text{ for all } i = 1, \ldots, n. \tag{42}$$

Proof: The proof follows from Theorem 4.7 replacing ε with $\varepsilon/2$ and defining $\delta = \varepsilon/(2\bar{m}L_g)$. Note that the thesis holds for the first iteration of the cycle becasue, in fact, due to Theorem 4.7, it holds for all its iterations. □

Theorems 4.1–4.8 give upper bounds on the number of iterations of Algorithm 3.1. (Bounds on the number of cycles translate into bounds on the number of iterations if multiplied by $\bar{m}$.) By definition, the sequence of σ's generated by Algorithm 2.1 is bounded from below by the sequence 0, $\tau_1^0\sigma_{\min}$, $\tau_1^1\sigma_{\min}$, $\tau_1^2\sigma_{\min}$, $\tau_1^3\sigma_{\min}$, ... Thus, by Theorem 2.1, the number of functional evaluations per call to Algorithm 2.1 at Step 2 of Algorithm 3.1 is bounded by

$$\log_{\tau_1}((L + \alpha)/\sigma_{\min}) + 2.$$

This establishes analogous bounds on the number of functional evaluations of Algorithm 3.1.

5 Discussion

Theorems 4.4 and 4.5 are complementary for showing that, eventually, Algorithm 3.1 computes an iterate x^k such that $\|\underline{g}_P(x^k)\|$ is smaller than a given tolerance; and that this task employs an amount of computer time that depends on tolerances and problem parameters. In Theorem 4.4, we proved that within $O(\varepsilon^{-(p+1)/p})$ iterations Algorithm 3.1 computes a sequence (cycle) of $\bar{m}$ iterates such that, for each $i = 1, \ldots, n$, there is at least one k such that $|[g_P(x^{k+1})]_i| \leq \varepsilon$. The number of required iterations for this purpose decreases with p and tends to $O(1/\varepsilon)$ when p tends to infinity. However, this result does not guarantee that the projected gradient norm is smaller than ε at a single iterate. For this purpose, we need the different iterates within a cycle to be clustered in a ball of small size. Unfortunately, in order to guarantee that this happens with tolerance δ, we need, according to Theorem 4.5, $O(1/\delta^{p+1})$ iterations. This quantity increases with p, which seems to indicate that, in the worst case, high-order coordinate descent is less efficient than low-order coordinate descent.

Examples given by Powell in [52] indicate that, in fact, this may be the case. In these examples, if coordinate descent is employed with exact coordinate minimization and cyclic coordinate descent, the generated sequence has more than one limit point. So, the distance between consecutive iterations does not tend to zero. This behavior is not observed if Algorithm 3.1 is applied because the descent condition (5) implies that $\lim \|x^{k+1} - x^k\| = 0$. However, exact minimization at each iteration evokes the case $p = \infty$ of Algorithm 3.1 in the sense that the trial

point computed as an exact minimizer satisfies the conditions for accepting the trial steps for any p. So, the conjecture arises that whether if one applies Algorithm 3.1 to Powell's examples with different values of p, the resulting sequence, although convergent to as solution, stays an increasing number of iterations oscilating around Powell's limiting cycle.

This conjecture is not easy to verify because, except one, Powell's examples are unstable in the sense that small perturbations cause convergence to the true minimizers far from the limit spurious cycle. In any case, we can emulate the application of Algorithm 3.1 to the most famous of Powell's examples (slightly modified here):

$$\text{Minimize } f(x_1, x_2, x_3) \equiv -(x_1x_2 + x_1x_3 + x_2x_3) + \sum_{i=1}^{3}(|x_i| - 0.1)_+^2. \tag{43}$$

If coordinate descent method employing exact coordinate minimization and cyclic coordinate descent is applied to problem (43) starting from

$$x^0 = (-0.1 - \epsilon, 0.1 + \epsilon/2, -0.1 - \epsilon/4),$$

it generates, after six iterations, an iterate x^6 that corresponds to x^0 with ϵ substituted with $\epsilon/64$, i.e.

$$x^6 = (-0.1 - \epsilon/64, 0.1 + \epsilon/128, -0.1 - \epsilon/256);$$

and, in general, for all k,

$$x^{6k} = (-0.1 - \epsilon/64^k, 0.1 + \epsilon/(2 \times 64^k), -0.1 - \epsilon/(4 \times 64^k)).$$

In the intermediate iterations, that are not multiples of 6, one has that

$$x^{6k+j} = (\pm 0.1 \pm \epsilon/\nu_{k,j}, \pm 0.1 \pm \epsilon/ \times \nu_{k,j}, \pm 0.1 \pm \epsilon/\nu_{k,j})$$

where $\nu_{k,j} \leq 4 \times 64^{k+1}$ for all k, j.

Now, we wish to show that this sequence could be generated by Algorithm 3.1. Moreover, for any given p, we wish to know how many iterations takes to obtain consecutive iterations such that $\|x^{k+1} - x^k\| \leq 0.01$. Starting with

$$x^0 = (-0.1 - \epsilon, 0.1 + \epsilon/2, -0.1 - \epsilon/4).$$

The global minimizer of $f(x_1, x_2, x_3)$ subject to $x_2 = x_2^0$ and $x_3 = x_3^0$ is

$$z^0 = (0.1 + \epsilon/8, 0.1 + \epsilon/2, -0.1 - \epsilon/4).$$

(The iterate x^1 in the Powell's sequence is given by $x^1 = z^0$, but we preserve the notation z^0 for the sake of simplicity.) On the one hand,

$$f(x^0) = -(x_1^0x_2^0 + x_1^0x_3^0 + x_2^0x_3^0) + \sum_{i=1}^{3}(|x_i^0| - 0.1)_+^2.$$

On the other hand, since $z_2^0 = x_2^0$ and $z_3^0 = x_3^0$,

$$f(z^0) = -(z_1^0 x_2^0 + z_1^0 x_3^0 + x_2^0 x_3^0) + (|z_1^0| - 0.1)_+^2 + \sum_{i=2}^{3} (|x_i^0| - 0.1)_+^2.$$

Therefore,

$$f(x^0) - f(z^0) = (z_1^0 - x_1^0)(x_0^2 + x_0^3) + (|x_1^0| - 0.1)_+^2 - (|z_1^0| - 0.1)_+^2.$$

Thus,

$$\begin{aligned} f(x^0) - f(z^0) &= ((0.1 + \epsilon/8) - (-0.1 - \epsilon))(\epsilon/2 - \epsilon/4) + (|-0.1 - \epsilon| - 0.1)_+^2 - (|0.1 + \epsilon/8| - 0.1)_+^2 \\ &= (0.2 + 9\epsilon/8)\epsilon/4 + \epsilon^2 - \epsilon^2/64 = 0.2\epsilon/4 + 9\epsilon^2/32 + \epsilon^2 - \epsilon^2/64 \\ &= 0.2\epsilon/4 + 9\epsilon^2/32 + \epsilon^2 - \epsilon^2/64 = 0.2\epsilon/4 + 81\epsilon^2/64 \geq \epsilon/20. \end{aligned}$$

Consider Algorithm 3.1 using $f(x)$ as the model of the objective function. We must verify whether (3), (4), and (5) are satisfied with $x^{\text{trial}} = z^0$. Trivially, for $\sigma = 0$, (3) and (4) hold by the definition of the model and the fact that z^0 is a global minimizer. In order to show that (5) also holds, let as assume that $\epsilon < 0.1$ and $2^{p+1} \geq 20\alpha/\epsilon$, i.e. $\alpha/2^{p+1} \leq \epsilon/20$. So, by the calculations above,

$$f(x^0) - f(x^{\text{trial}}) \geq \alpha/2^{p+1}.$$

Since $\epsilon < 0.1$, we have that $\|x^{\text{trial}} - x^0\| \leq 0.5$. Thus,

$$f(x^0) - f(x^{\text{trial}}) \geq \alpha \|x^{\text{trial}} - x^0\|^{p+1}.$$

This implies (5). Therefore, a sufficient condition for the acceptance of $x^1 = z^0$ as an iterate of Algorithm 3.1 is

$$\alpha/2^{p+1} \leq \frac{\epsilon}{20 \times 4 \times 64^k}.$$

In other words,

$$20 \times 4 \times 64^k \alpha \leq \epsilon 2^{p+1}.$$

Taking logarithms, this condition is

$$\log_2 80 + 6k + \log_2 \alpha \leq p + 1.$$

That is, if

$$k_0 \leq (p + 1 - \log_2 80 - \log_2 \alpha)/6,$$

the first k_0 iterations of Algorithm 3.1 will reproduce the cycling example of Powell. In all these iterations we have that $\|x^{k+1} - x^k\| \geq 0.1$. Note that k_0 tends to infinity as p tends to infinite, as we wanted to show. In addition, note also that k_0 tends to infinity as α tends to zero, which reflects the obvious fact that, if we are more tolerant with the acceptance of the trial point, the probability of staying around Powell's six-points cycle increases.

It is not sensible to decide about usefulness of algorithms based only on theoretical convergence or complexity results. Since these results deal with worst-case behavior the possibility exists that a class of problems in which practitioners are interested always exhibit characteristics that exclude extreme unfortunate cases. However, it is pertinent to examine pure mathematical properties in order to foster unexpected good or bad computer behaviors.

1. Many optimization users believe that if a smooth function has a minimizer at a point x^*, then this point is a local minimizer of all its Taylor polynomials. This is true only if the dimension n is equal to 1. For arbitrary n, it is true only up to second order polynomials. Examples that illustrates this phenomenon have been given in this paper with the purpose of justifying adequate high-order optimality conditions (for example, $f(x_1, x_2) = x_2^2 - x_1^2 x_2 + x_1^4$). This fact implies that, in the vicinity of a global minimizer, a high-order algorithm may try to find improvements far from the current point, being subject to a painful sequence of "backtrackings" before obtaining descent. Does this imply that only quadratic approximations are useful in the minimization context? It is too soon to give a definite response to this question.

2. Our regularization approach for CD-algorithms makes it impossible cyclical behavior as the one exhibited by Powell's examples [52]. The reason is that, under regularization descent algorithms, the difference between consecutive iterates tends to zero. However, it seems to be possible that convergence to zero of consecutive iterates could be very slow, as predicted by complexity results. Is this an argument for discarding high-order CD algorithms? We believe that the answer is no, as far as the use of CD algorithms is, in general, motivated by the structure of the problems, which in some sense should evoke some degree of separability. Moreover, since high-order models are also low-order models one can use high-order associated with a small p in (3), (4), and (5).

3. It is interesting to consider the case in which we use $f(x)$ as a model for $f(x)$. In this case, high-order analysis makes a lot of sense. In fact, efficient algorithms for finding global minimizers of functions of one variable exist, a possibility that decreases very fast as the number of variables grow. Moreover high-order one-dimensional models are certainly affordable and many numerical analysis papers handle efficiently the problem of minimizing or finding roots of univariate polynomials [51]. Recall that, in this case, the model satisfies the approximation requirements for every value of p. Therefore we may choose the value of p that promises better efficiency, which, according to Theorem 4.8, should be $p = 1$ giving complexity $O(\varepsilon^{-2})$ as gradient-like methods.

4. In most practical situations one is interested in finding global minimizers or, at least, feasible points at which the objective function value is smaller than a given f_{target}. Complexity and convergence analyses in the nonconvex world concern only the approximation to stationary points although every practical algorithm must be devised taking into account the global implicit goal. It turns out that low coordinate global strategies for finding initial points are available in many real-life problems. These strategies fit well with CD algorithms as we will illustrate in Section 6.

5. The reader will observe that in our experiments we used $p = 2$, in spite that, according to the complexity results, the optimal p should be 1. The reason is that, as we stated in the convergence section, the employment of $p = 2$ guarantees convergence to points that satisfy second order conditions that are not guaranteed by $p = 1$. Moreover, subproblems with $p = 2$ are computationally affordable in the applications considered. Summing up, we could say that making an informal balance regarding theoretical results, using $p = 2$ should be the default choice for practical applications.

6 Implementation and experiments

In this section, we illustrate with numerical experiments the applicability of Algorithm 3.1 by applying it to Multidimensional Scaling (MS) problems [29, 48, 54]. Multidimensional Scaling methods emerged as statistical tools in Psychophysics and sensory analysis. The MS problem considered in this section may be described in the following way: Let $x_1, \ldots, x_{n_p} \in \mathbb{R}^d$ be a set of unknown points. Let $D = (d_{ij}) \in \mathbb{R}^{n_p \times n_p}$ be such that $d_{ij} = \|x_i - x_j\|$; and assume that only entries $\{d_{ij} \mid (i,j) \in S\}$ for a given $S \subset \{1, \ldots, n_p\} \times \{1, \ldots, n_p\}$ are known. (Of course, D is symmetric, $d_{ii} = 0$, and $(i,j) \in S$ if and only if $(j,i) \in S$.) Then the MS problem consists of finding $x_1, \ldots, x_{n_p}$ such that $\|x_i - x_j\| = d_{ij}$ for all $(i,j) \in S$. Glunt, Hayden, and Raydan [35] were the first to apply unconstrained continuous optimization tools to the nowadays called Molecular Distance Geometry Problem (MDGP), as defined in [41, 42] in a Multidimensional Scaling context. This problem appears when points $x_1, \ldots, x_{n_p}$ correspond to the positions of atoms in a molecule and distances not larger than 6 Angstroms (i.e. 6×10^{-10} meters) are obtained via nuclear magnetic resonance (NMR); see [1]. This problem can be modeled as the following unconstrained nonlinear optimization problem

$$\underset{x_1,\ldots,x_{n_p} \in \mathbb{R}^d}{\text{Minimize}} \; f(x_1, \ldots, x_{n_p}) := \frac{1}{|S|} \sum_{(i,j) \in S} \left(\|x_i - x_j\|_2^2 - d_{ij}^2 \right)^2. \tag{44}$$

In the MDGP, the idea of associating the set I_k of iteration k with the components of a point $x_k \in \mathbb{R}^d$ that is modified at each iteration of Algorithm 3.1 is quite natural, i.e. we define $x = (x_1^T, \ldots, x_{n_p}^T)^T \in \mathbb{R}^n$ with $n := d\, n_p$ and, at iteration k,

$$I_k = \{(\ell(k)-1)d + 1, \ldots, (\ell(k)-1)d + d\} \text{ with } \ell(k) = \text{mod}(k, n_p) + 1, \tag{45}$$

or any alternative choice of $\ell(k) \in \{1, \ldots, n_p\}$. This is equivalent to say that, at iteration k, the subproblem considered at Step 2 of Algorithm 3.1 is given by

$$\underset{z \in \mathbb{R}^d}{\text{Minimize}} \; \underline{f}(z), \tag{46}$$

where $\underline{f} : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\underline{f}(z) := \frac{1}{|S|} \left[\sum_{(i,j) \in S \setminus S(\ell(k))} \left(\|x_i - x_j\|_2^2 - d_{ij}^2 \right)^2 + 2 \sum_{(i,\ell(k)) \in S} \left(\|x_i - z\|_2^2 - d_{i,\ell(k)}^2 \right)^2 \right], \tag{47}$$

$S(\ell(k)) := \{(i,j) \in S \mid i = \ell(k) \text{ or } j = \ell(k)\}$, and $\ell(k)$ given in (45) is constant. Note that the time complexity for evaluating f is $O(d|S|)$; while, since the first summation in (47) does not depend on z, the time complexity for evaluating $\underline{f}$ is, in average $O(d|S|/n_p)$.

For approximately solving (46) in Algorithm 2.1, we considered a second-order Taylor expansion of $\underline{f}$ at $\bar{x} = x^k_{\ell(k)} \in \mathbb{R}^d$, i.e.

$$M_{\bar{x}}(z) := \underline{f}(\bar{x}) + \nabla \underline{f}(\bar{x})^T (z - \bar{x}) + (z - \bar{x})^T \nabla^2 \underline{f}(\bar{x})^T (z - \bar{x}).$$

This means that the underlying model-based subproblem, when Algorithm 2.1 is used at Step 2 of the kth iteration of Algorithm 3.1 is given by

$$\underset{z\in\mathbb{R}^d}{\text{Minimize}}\, M_{\bar{x}}(z) + \sigma\|z-\bar{x}\|^3. \tag{48}$$

Since problem (44) is unconstrained, i.e. $\Omega = \mathbb{R}^n$, subproblems (46) and model-based subproblems (48) are unconstrained as well. Thus, if in (48) and, in consequence, in (5), we consider $\|\cdot\|$ as $\|\cdot\|_3$, then the *global* minimizer of (48) can be easily obtained at the expense of a single factorization of $\nabla^2 \underline{f}(\bar{x}) \in \mathbb{R}^{d\times d}$. (When $\sigma = 0$, (48) may have no solution. This case can be detected with the same cost as well.) See [16, 46, 47, 9]. Since the exact global minimizer x^{trial} of (48) is being computed at Step 2 of Algorithm 2.1, (3) and (4) always hold, for any $\theta > 0$; thus, in the implementation, their verification can be ignored.

As shown in Section 3, Algorithm 3.1 has convergence properties towards stationary points which, probably, are local minimizers. Obviously, as we are interested in finding *global* minimizers of MDGP, we need suitable strategies for choosing initial approximations. We employed two different strategies for this purpose. On the one hand, an initial guess suggested in [34] was adopted. On the other hand, we devised a new coordinate descent procedure based on the structure of MDGP.

In order to describe the Fang-O'Leary strategy [34], consider the weighted graph $G = (\{1,\ldots,n_p\}, S)$ in which the weight of an edge (i,j) is given by d_{ij}. We assume this graph is connected. Otherwise, the molecule's structure can not be recovered; and problem (44) can be decomposed in as many independent problems as connected components of the graph G in order to recover partial structures. Let $\bar{S} = \{1,\ldots,n_p\}\times\{1,\ldots,n_p\}\setminus S$, i.e. $\bar{S}$ corresponds to the missing arcs in G or, equivalently, the unknown entries of D. For each $(i,j)\in S$, define $\tilde{d}_{ij} = d_{ij}$; and for each $(i,j)\in\bar{S}$, define $\tilde{d}_{ij}$ as the weight of the shortest path between i and j in G. Matrix $\tilde{D} = (\tilde{d}_{ij})$ is a distance matrix that completes D; but with high probability it is *not* an Euclidean distance matrix. Computing $\tilde{D}$ requires $O(n_p^2)$ space and has time complexity $O(n_p^3)$ (using the Floyd-Warshall algorithm as suggested in [34]), which can be an issue for instances with large n_p. Obtaining points $x_1^0,\ldots,x_{n_p}^0\in\mathbb{R}^d$ from $\tilde{D}$ requires to compute the d largest positive eigenvalues of the matrix $\mathcal{T}(\tilde{D})$ given by $\mathcal{T}(\tilde{D}) := -\frac{1}{2}J\tilde{D}J$, where $J := I - \frac{1}{n}ee^T$ and $e = (1,\ldots,1)^T$. If the truncated spectral decomposition of $\mathcal{T}(\tilde{D})$ is given by $U\Delta_d U^T$ then the initial point $x^0 = ((x_1^0)^T,\ldots,(x_{n_p}^0)^T)^T$ is given by $X = (x_1^0,\ldots,x_{n_p}^0) = U\Delta_d^{1/2}$. If matrix $\mathcal{T}(\tilde{D})$ has only $\underline{d} < d$ positive eigenvalues, then computed points are in $\mathbb{R}^{\underline{d}}$ and their last $d-\underline{d}$ components can be completed with zeros. In [34], alternative initial guesses are obtained by perturbations of matrix $\tilde{D}$ and/or by stretching the computed points $x_1^0,\ldots,x_{n_p}^0$.

Our coordinate-descent strategy for choosing the initial approximation to the solution of MDGP was inspired on the structure of local solutions. Consider a point $p\in\mathbb{R}^3$ and three other points $q_1,q_2,q_3\in\mathbb{R}^3$ such that the distances from p to q_i, $i=1,2,3$, are known and satisfied. Assume that there is an additional point q_4 for which its known distance $d(p,q_4)$ to p is *not* satisfied. Assume, in addition, that $(\|r(p)-q_4\|_2^2 - d(p,q_4)^2)^2 < (\|p-q_4\|_2^2 - d(p,q_4)^2)^2$, where $r(p)$ is the reflection of p on the plane determined by q_i, $i=1,2,3$. If there were no more points in the problem, replacing p by $r(p)$, would produce a reduction in the objective function. Our coordinate descent algorithm with a coordinate-descent strategy for choosing initial points is

described in Algorithm 6.1. The coordinate-descent strategy for initial approximations, based on this intuition, is described at Step 4 of Algorithm 6.1.

Algorithm 6.1. Assume $\hat{x}$ is a given arbitrary initial point (that might be obtained using the Fang-O'Leary technique described above).

Step 1. Using $\hat{x}$ as initial guess, run Algorithm 3.1 until the obtention of an iterate $\tilde{x}$ such that $f(\tilde{x}) \leq f_{\text{target}}$ or such that its projected gradient is small enough according to criteria given below.

Step 2. If $f(\tilde{x}) \leq f_{\text{target}}$ then **stop** declaring that $\tilde{x}$ is a global minimizer up to the precision given by f_{target}. Otherwise, update $\hat{x}$ by means of the coordinate-descent strategy in Step 3 below.

Step 3. For $j = 1, \ldots, n_p$ execute Steps 3.1–3.2.

Step 3.1. Let $\hat{f}_j := \sum_{(i,j)\in S}(\|\hat{x}_i - \hat{x}_j\|_2^2 - d_{ij}^2)^2$.

Step 3.2. For every triplet (i_1, i_2, i_3) such that $(i_1, j), (i_2, j), (i_3, j) \in S$, in an arbitrary order, if

$$\sum_{(i,j)\in S}(\|\hat{x}_i - r(\hat{x}_j)\|_2^2 - d_{ij}^2)^2 < \hat{f}_j,$$

where $r(\hat{x}_j)$ is the reflection of $\hat{x}_j$ on the plane determined by $\hat{x}_{i_1}$, $\hat{x}_{i_2}$, and $\hat{x}_{i_3}$, then update $\hat{x}_j \leftarrow r(\hat{x}_j)$. (Note that $\hat{f}_j$ is not updated at this point. This means that a sequence of reflections can be applied to $\hat{x}_j$, with a non-monotone behavior of f, provided it improves the "reference value" $\hat{f}_j$.)

Step 4. If $\hat{x}$ was not updated at Step 3, then stop returning $\tilde{x}$. (Note that f_{target} was not reached in this case.) Otherwise, go to Step 1.

At Step 1 of Algorithm 6.1, we consider that "the projected gradient is small enough" if, during n_p consecutive iterations of Algorithm 3.1, we have that "the final σ" of Algorithm 2.1 is larger than 10^{20} or $f(x^{k+1}) \not\leq f(x^k) - 10^{-8}\min\{1, |f(x^k)|\}$. By (16), (17) and the boundedness of σ, these are practical symptoms of stationarity.

We implemented Algorithms 2.1, 3.1, and 6.1 in Fortran. In the numerical experiments, we considered, $\alpha = 10^{-8}$, $\sigma_{\min} = 10^{-8}$, and $\tau_1 = \tau_2 = 100$, and $f_{\text{target}} = 10^{-10}$. All tests were conducted on a computer with a 3.4 GHz Intel Core i5 processor and 8GB 1600 MHz DDR3 RAM memory, running macOS Mojave (version 10.14.6). Code was compiled by the GFortran compiler of GCC (version 8.2.0) with the -O3 optimization directive enabled.

The Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank [56] is an open access repository that provides access to 3D structure data for large biological molecules (proteins, DNA, and RNA). There are more than 167,000 molecules available. In [34], where Newton and quasi-Newton methods are applied to problem (44), six protein molecules are considered, namely, 2IGG, 1RML, 1AK6, 1A24, 3MSP, and 3EZA (see [34, Table 6.9, p.20]); while in [1], where the Douglas–Rachford is applied, other six protein molecules are considered, namely, 1PTQ, 1HOE, 1LFB, 1PHT, 1POA, and 1AX8 (see [1, Table 1, p.313]). In the first work, only protein atoms (identified with ATOM in the molecule file) were considered; while in the second work there were considered protein atoms plus atoms in small molecules (identified with

HETATM in the protein molecule file). In the current work, both options were considered. Following [34], for each protein molecule, when multiple structures are available, only the first one was considered. Each molecule is given as the set of 3D coordinates of its atoms. An instance of problem (44) is built by computing a complete Euclidean distance matrix and then eliminating distances larger than 6 Angstroms. Since not all molecules have atoms in small molecules, we arrived to eighteen different instances. Table 1 shows, for each instance, the number of variables n of the optimization problem (44), the number of atoms n_p, the number of distances considered to be known $|S|$, and the CPU time in seconds required to construct the initial guess x^0 using the Fang-O'Leary strategy [34].

		Molecule	n	n_p	$\lvert S\rvert$	Time x^0
Points may correspond to protein atoms (ATOM) only or to protein atoms plus atoms in small molecules (HETATM)	ATOM only	1ptq	1,206	402	14,176 (8.79%)	0.21
		1hoe	1,674	558	20,356 (6.55%)	0.49
		1lfb	1,923	641	22,870 (5.57%)	0.70
		1pht	2,433	811	35,268 (5.37%)	1.41
		1poa	2,742	914	33,966 (4.07%)	2.03
		2igg	2,919	973	62,574 (6.62%)	2.54
		1ax8	3,009	1,003	37,590 (3.74%)	2.76
		1rml	6,192	2,064	153,660 (3.61%)	24.14
		1ak6	8,214	2,738	224,568 (3.00%)	52.04
		1a24	8,856	2,952	212,364 (2.44%)	64.90
		3msp	11,940	3,980	262,876 (1.66%)	157.90
		3eza	15,441	5,147	356,544 (1.35%)	335.84
	ATOM+HETATM	1ptq	1,212	404	14,370 (8.83%)	0.21
		1hoe	1,743	581	21,422 (6.36%)	0.55
		1pht	2,964	988	44,542 (4.57%)	2.59
		1poa	3,201	1,067	41,034 (3.61%)	3.23
		1ax8	3,222	1,074	40,866 (3.55%)	3.29
		1rml	6,273	2,091	156,550 (3.58%)	23.90

Table 1: Description of the instances built with the molecules considered in [1] or [34].

Note that considered instances are *gedanken* in the sense that points $\bar{x}_1, \ldots, \bar{x}_{n_p} \in \mathbb{R}^3$ such that $f(\bar{x}) = 0$ with $\bar{x}^T = (\bar{x}_1^T, \ldots, \bar{x}_{n_p}^T)^T$ are known. Thus, given x^* such that $f(x^*) \approx 0$, we may wonder whether x^* is close to $\bar{x}$. The answer to this questions is "Not necessarily." since any rotation or translation of $\bar{x}$ also annihilates f. So the question would be "How close is x^* to $\bar{x}$ after performing the appropriate rotations and translations?". The answer to this question is obtained by solving an orthogonal Procrustes problem. Let $\bar{X} = (\bar{x}_1, \ldots, \bar{x}_{n_p})$ and $X^* = (x_1^*, \ldots, x_{n_p}^*) \in \mathbb{R}^{3\times n_p}$. It is easy to see that matrices $\bar{X}J$ and X^*J have their centroid at the origin, since $\bar{X}Je = X^*Je = 0$. (Recall that $J = I - \frac{1}{n_p}ee^T$ and $e = (1, \ldots, 1)^T$.) The orthogonal Procrustes problem consists in finding an orthogonal matrix $Q \in \mathbb{R}^{3\times 3}$ which most

closely maps X^*J to $\bar{X}J$, i.e.

$$Q = \underset{R\in\mathbb{R}^{3\times 3}}{\operatorname{argmin}} \|RX^*J - \bar{X}J\|_F^2 \text{ subject to } RR^T = I.$$

This problem has a closed form solution given by $Q = VU^T$, where $U\Sigma V^T$ is the singular values decomposition of the matrix $C := X^*J(\bar{X}J)^T$. Thus, the measure we were looking for is given by

$$E(x^*) := \max_{\{j=1,\ldots,n_p\}} \left\{E(x_j^*)\right\},$$

where

$$E(x_j^*) := \frac{\|[QX^*J - \bar{X}J]_j\|_\infty}{\max\{1, \|[\bar{X}J]_j\|_\infty\}}, \tag{49}$$

and $[A]_j$ denotes the jth column of matrix A.

Table 2 shows the performance of Coordinate Descent, the Spectral Projected Gradient (SPG) method [11, 12], and Gencan [7, 10]. In all cases, the initial point given by the Fang-O'Leary technique was used. Since problem (44) is unconstrained, applying SPG corresponds to applying the Spectral Gradient methods as proposed in [35]; while applying Gencan corresponds to applying a line search Newton's method as considered in [34]. All three methods used as stopping criterion $f(x^k) \leq f_{\text{target}} := 10^{-10}$. In addition, SPG and Gencan also stopped if $\|\nabla f(x^k)\|_\infty \leq \varepsilon_{\text{opt}} := 10^{-8}$. For all three methods the table shows the number of iterations (#iter), the CPU time in seconds (Time), the value of the objective function at the final iterate ($f(x^*)$), and the error with respect to the known solution ($E(x^*)$). In addition, the table shows, for the coordinate descent method the number of evaluations of f; while it shows for the other two methods, the number of evaluations of f and $\|\nabla f(x^*)\|_\infty$. In the table, highlighted figures in column $f(x^*)$ are the ones that correspond to local minimizers. Highlighted figures in column $E(x^*)$ correspond to final iterates that are far from the known solution. In most cases, this fact is associated with having found a local minimizer. However, in some cases, it corresponds to an alternative global minimizer. We may observe that coordinate descent stands out as the only method to have found a global minimizer in all the eighteen considered instances. Figures 2 and 3 illustrate three molecules in which the coordinate descent method found a global solution while SPG and Gencan found local non-global minimizers. It is worth mentioning that the numerical experiments reported in [1] show that the Douglas-Rachford method, that requires an SVD decomposition of a $n_p \times n_p$ matrix per iteration, with a limit of 5,000 iterations, was able to reconstruct the two smallest molecules (1PTQ and 1HOE) only. As reported in [1], the reconstruction of molecules 1LFB and 1PHT was "satisfactory"; while the reconstruction of molecules 1POA and 1AX8 was "poor".

The natural question that arises is whether the tendency of the coordinate descent method in finding global minimizers could be observed in a larger set of instances. To check this hypothesis, we downloaded 64 additional *random* molecules with no more than 6,000 atoms from the ones that were uploaded in 2020; 56 of which have, other than protein atoms, atoms in small molecules. However there were 19 molecules for which, considering protein atoms only or protein atoms plus atoms in small molecules, the graph associated with the incomplete Euclidean matrix obtained by eliminating distances larger than 6 Angstroms is disconnected. Therefore, we were left with 45 and 37 molecules in each set, totalizing 82 new instances. Table 3 shows the performance of

		Molecule	Coordinate descent					Spectral Projected Gradient						Gencan					
			#iter	#f	Time	$f(x^*)$	$E(x^*)$	#iter	#f	Time	$f(x^*)$	$\|\nabla f(x^*)\|_\infty$	$E(x^*)$	#iter	#f	Time	$f(x^*)$	$\|\nabla f(x^*)\|_\infty$	$E(x^*)$
Points may correspond to protein atoms (ATOM) only or to protein atoms plus atoms in small molecules (HETATM)	ATOM only	1ptq	57,671	57,686	0.13	9.99e-11	1.66e-06	333	334	0.05	1.12e-11	3.32e-07	2.60e-06	9	13	0.42	5.82e-13	1.45e-07	2.58e-06
		1hoe	135,886	135,907	0.30	9.99e-11	2.36e-06	126	128	0.03	8.52e-11	4.39e-07	3.62e-06	7	10	0.62	5.49e-11	1.26e-06	7.25e-06
		1lfb	811,486	811,613	1.79	9.99e-11	7.56e-06	738	755	0.19	1.77e-11	5.19e-07	1.11e-06	13	20	1.15	5.96e-11	1.05e-06	4.27e-06
		1pht	786,655	786,831	1.99	9.99e-11	1.79e-05	5,945	6,856	2.50	2.85e-02	5.04e-09	2.08e-01	127	340	28.67	2.85e-02	9.10e-07	2.08e-01
		1poa	704,652	704,762	1.59	9.99e-11	9.79e-06	7,367	8,716	3.00	9.95e-11	3.00e-08	5.84e-04	18	19	2.71	1.79e-11	3.68e-07	2.32e-04
		2igg	484,388	484,473	1.56	9.99e-11	9.12e-06	304	305	0.21	8.79e-11	3.18e-07	3.43e-06	11	22	4.58	2.45e-13	1.24e-08	2.58e-07
		1ax8	353,820	353,895	0.80	9.99e-11	2.54e-06	325	326	0.14	8.11e-11	1.74e-07	2.02e-05	14	20	3.59	7.98e-13	1.97e-08	2.14e-06
		1rml	340,528	340,586	1.24	9.99e-11	4.07e-06	236	238	0.39	1.21e-12	9.05e-08	1.46e-06	9	10	30.48	3.77e-13	6.47e-08	1.23e-06
		1ak6	15,138,479	15,138,810	229.85	9.99e-11	1.35e-05	1,662	1,755	4.06	5.18e-02	9.92e-09	1.97e-01	166	421	953.67	5.18e-02	7.47e-09	1.97e-01
		1a24	2,840,577	2,840,834	9.87	9.99e-11	1.39e-05	322	325	0.74	8.10e-11	5.88e-08	1.15e-05	19	48	74.23	8.76e-12	1.76e-07	7.89e-06
		3msp	12,873,352	12,874,426	42.40	9.99e-11	1.61e-05	672	688	1.93	1.41e-10	8.47e-09	1.86e-05	31	55	138.70	1.24e-11	1.56e-08	5.46e-06
		3eza	17,122,466	17,123,479	58.89	9.99e-11	1.03e-05	580	586	2.26	4.43e-10	9.66e-09	2.11e-05	23	51	224.11	1.24e-10	3.08e-09	1.18e-05
	ATOM+HETATM	1ptq	57,640	57,659	0.13	9.99e-11	1.61e-06	334	335	0.05	8.45e-11	2.81e-07	3.60e-05	10	15	0.45	3.83e-17	9.27e-10	2.24e-08
		1hoe	129,571	129,590	0.31	9.99e-11	2.25e-06	143	144	0.04	3.48e-11	3.47e-07	2.82e-06	8	11	0.76	8.61e-18	4.80e-10	3.14e-09
		1pht	946,496	946,610	8.26	9.99e-11	1.60e-05	1,541	1,608	0.78	1.61e-05	9.85e-09	1.04e-01	21	31	8.48	1.61e-05	1.49e-09	1.04e-01
		1poa	409,610	409,655	0.93	9.99e-11	5.20e-05	12,996	15,710	6.43	1.84e-10	9.99e-09	1.57e-02	15	18	4.31	1.93e-11	4.49e-07	1.38e-02
		1ax8	308,962	309,026	0.70	9.99e-11	2.18e-06	148	149	0.07	9.43e-11	2.25e-07	1.04e-05	8	11	3.18	1.54e-12	2.35e-07	6.49e-07
		1rml	344,977	345,021	1.28	9.99e-11	4.01e-06	305	307	0.52	9.48e-11	1.55e-07	4.36e-05	10	11	36.41	9.82e-17	5.53e-10	4.35e-08

Table 2: Performance of Coordinate Descent, SPG, and Gencan applied to the instances of problem (44) built with the molecules considered in [1] or [34].

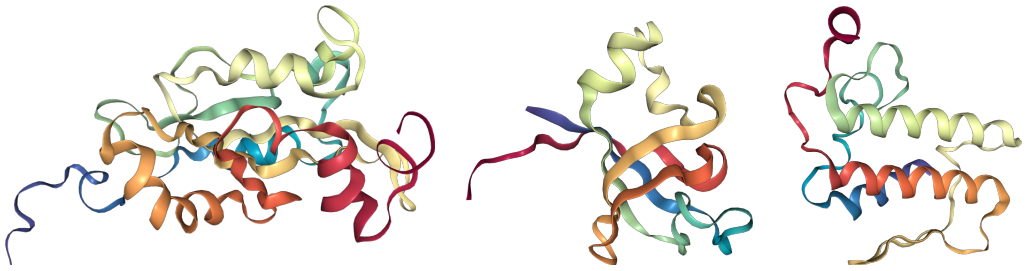

Figure 2: Representation of molecules 1AK6, 1PHT, and 1POA for which Coordinate Descent found a global minimizer; while SPG and Gencan found a local minimizer.

Coordinate Descent and SPG when applied to the 45 instances that consider protein atoms only; while Table 4 shows the performance of both methods when applied to the 37 instances that consider protein atoms plus atoms in small molecules. In the 45 instances in Table 3, Coordinate Descent found 37 global minimizers; while SPG found 30 global minimizers. This means that Coordinate Descent found 23% more global minimizers than SPG. In the 37 instances in Table 4, Coordinate Descent found 30 global minimizers; while SPG found 26 global minimizers. This means that Coordinate Descent found 15% more global minimizers than SPG.

7 Conclusions

Methods based on high-order models for optimization are difficult to implement due to the necessity of computing and storing high-order derivatives and the complexity of solving the subproblems. These difficulties are not so serious if the subproblems are low-dimensional, which is the most frequent situation in the case of CD methods. In the extreme case, in which one solves only univariate problems, the number of high-order partial derivatives that are necessary is a small multiple of the number of variables. Therefore, the theory that shows that CD algorithms

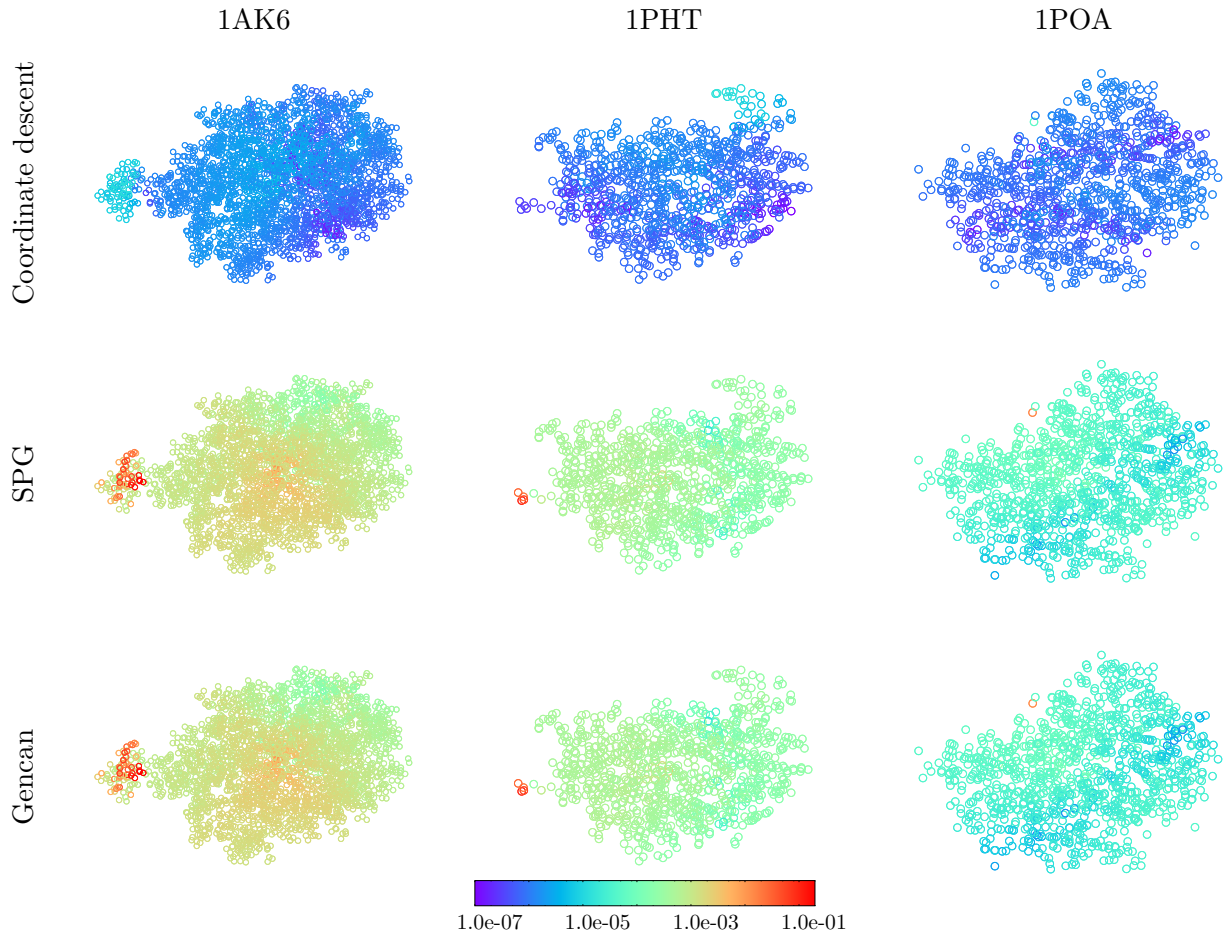

Figure 3: Molecules 1AK6, 1PHT, and 1POA for which Coordinate Descent found a global minimizer; while SPG and Gencan found a local minimizer. To the naked eye, solutions would appear to be indistinguishable. Therefore, the figures show, for each point $x_1^*, \ldots, x_{n_p}^*$, the value of $E(x_j^*)$ as defined in (49).

with high-order models enjoy good convergence and complexity properties seems to be useful to support the efficiency of practical implementations.

Sometimes the fulfillment of a necessary high-order optimality condition can be expressed as fulfillment of $\Phi(x) = 0$, where Φ is a continuous nonnegative function. In this case, it makes sense to say that $\Phi(x) \leq \varepsilon$ is an approximate high-order optimality condition. Moreover, instead of requiring globality for the solution to the regularized model-based subproblem (6), we may require only that $\Phi(x^{k+1}) \to 0$ when $k \to +\infty$, where Φ corresponds to the high-order optimality condition of (6). Careful choices of Φ and the subproblems' stopping criterion may give rise to complexity results associated with the attainment of these high-order optimality conditions. See [24, 25, 26].

In this paper the defined algorithms were applied to the identification of proteins under

Molecule	n	n_p	$\|S\|$	Time x^0	Coordinate descent					Spectral Projected Gradient					
					#iter	$\#f$	Time	$f(x^*)$	$E(x^*)$	#iter	$\#f$	Time	$f(x^*)$	$\|\nabla f(x^*)\|_\infty$	$E(x^*)$
6kbq	7,554	2,518	98,650	1.56	40.49	10,543,341	10,543,995	36.22	9.99e-11	2.65e-05	810	911	0.96	2.65e-02	7.65e-09	1.87e-01
6kc2	7,530	2,510	98,480	1.56	39.97	7,682,589	7,683,021	20.93	9.99e-11	2.62e-05	954	984	1.11	4.48e-12	3.43e-08	1.29e-05
6khu	3,147	1,049	39,372	3.58	3.09	,316,876	316,959	0.83	9.99e-11	6.12e-06	411	413	0.19	5.66e-11	1.05e-06	7.83e-06
6kir	3,861	1,287	48,596	2.94	5.58	7,574,444	7,574,767	54.37	1.60e-03	1.50e-01	871	895	0.49	5.27e-11	1.01e-07	3.82e-05
6kk9	17,538	5,846	218,662	0.64	479.82	215,549,505	215,561,546	712.59	9.29e-01	1.19e+00	6,262	7,030	17.33	9.09e-01	9.83e-09	1.15e+00
6kki	8,193	2,731	109,472	1.47	53.06	2,113,286	2,113,472	5.68	9.99e-11	5.04e-06	432	434	0.56	9.34e-11	1.33e-07	2.99e-05
6kkj	7,866	2,622	105,642	1.54	47.70	4,068,836	4,070,082	10.88	9.99e-11	5.18e-06	5,868	6,787	7.70	3.24e-10	9.99e-09	1.58e-03
6kkl	8,262	2,754	112,646	1.49	54.76	6,064263	6,064,648	16.61	9.99e-11	5.68e-06	592	604	0.78	9.47e-11	5.99e-08	6.48e-06
6kkv	14,607	4,869	357,016	1.51	290.74	31,543,917	31,545,196	128.60	9.99e-11	1.86e-05	1,095	1,140	4.65	2.88e-11	6.93e-08	2.00e-06
6kx0	9,885	3,295	128,930	1.19	91.69	38,237,232	38,239,417	196.30	1.57e-01	1.02e+00	1,428	1,546	2.26	1.57e-01	9.87e-09	1.02e+00
6kys	5,442	1,814	70,296	2.14	16.13	984,382	984,559	2.49	9.99e-11	4.00e-06	359	360	0.29	8.44e-11	1.22e-07	2.95e-05
6l29	5,427	1,809	70,480	2.15	16.68	965,426	965,620	2.49	9.99e-11	4.51e-06	433	436	0.36	8.43e-11	7.93e-07	8.26e-06
6l2a	2,463	821	29,876	4.44	1.55	411,298	411,381	1.04	9.99e-11	5.55e-06	1,612	1,699	0.59	9.95e-11	7.47e-08	6.46e-05
6laf	8,442	2,814	105,164	1.33	58.06	31,413,895	31,418,665	104.66	9.99e-11	2.51e-05	7,706	9,068	10.22	1.53e-02	9.76e-09	2.74e-01
6li7	7,515	2,505	98,374	1.57	40.21	10,533,446	10,534,072	37.27	9.99e-11	2.64e-05	698	713	0.83	8.73e-11	4.32e-07	1.54e-05
6lik	7,515	2,505	97,354	1.55	39.67	7,579,646	7,580,087	19.88	9.99e-11	2.63e-05	727	749	0.85	8.89e-11	2.61e-07	6.42e-05
6lty	5,964	1,988	73,296	1.86	19.95	3,800,264	3,800,792	9.82	9.99e-11	7.86e-06	672	683	0.58	9.76e-11	9.54e-08	1.78e-05
6ltz	3,432	1,144	41,230	3.15	3.97	,975,177	975,323	2.48	9.99e-11	3.13e-06	8,025	9,404	4.08	1.27e-10	9.94e-09	9.66e-04
6m37	9,999	3,333	125,394	1.13	91.51	130,035,728	130,040,179	422.49	5.07e-02	4.33e-01	4,936	5,479	7.63	5.06e-02	9.99e-09	4.30e-01
6m5n	5,781	1,927	73,890	1.99	18.48	8,183,149	8,183,657	27.54	9.99e-11	6.22e-06	6,419	7,461	5.81	3.06e-03	9.98e-09	2.74e-01
6m6j	759	253	13,914	21.82	0.07	13,076	13,076	0.04	9.99e-11	1.84e-06	59	61	0.01	3.99e-12	1.99e-07	6.82e-07
6m6k	747	249	13,672	22.14	0.07	11,376	11,376	0.04	9.97e-11	1.29e-06	53	55	0.01	2.83e-11	5.59e-07	1.41e-06
6pq0	6,087	2 029	84,104	2.04	22.54	2,346,681	2,346,842	31.31	9.99e-11	3.83e-06	440	442	0.43	2.39e-04	8.43e-09	1.83e-01
6pup	4,035	1,345	50,372	2.79	6.57	544,519	544,617	1.42	9.99e-11	4.87e-06	409	410	0.24	2.52e-11	7.99e-08	2.38e-05
6pxf	4,944	1,648	64,944	2.39	11.70	811,879	812,002	2.13	9.99e-11	2.77e-05	4,907	5,583	3.88	1.81e-10	9.41e-09	2.91e-03
6q08	741	247	13,232	21.78	0.07	22,723	22,744	0.07	9.99e-11	7.19e-06	173	176	0.03	5.75e-11	1.85e-06	3.82e-06
6sx6	2,340	780	44,862	7.38	1.28	143,230	143,260	0.49	9.99e-11	2.50e-06	136	137	0.07	5.51e-12	1.80e-07	2.60e-07
6syk	2,718	906	54,052	6.59	1.99	156,648	156,707	0.55	9.99e-11	2.50e-06	585	592	0.36	9.41e-12	2.34e-07	1.42e-05
6t1z	8,943	2,981	119,580	1.35	65.75	14,818,888	14,819,492	40.30	9.99e-11	8.88e-06	3,247	3,539	4.71	9.94e-11	1.37e-08	2.70e-04
6tad	4,362	1,454	58,736	2.78	8.07	2,385,243	2,385,423	6.35	9.99e-11	6.84e-06	632	643	0.44	7.87e-11	3.81e-07	5.70e-06
6twe	7,902	2,634	163,598	2.36	0.18	5,773,930	5,774,426	20.85	9.99e-11	8.20e-06	598	612	1.16	7.04e-11	1.66e-07	6.93e-06
6ubh	9,009	3,003	113,766	1.26	69.74	8,782,138	8,782,736	22.99	9.99e-11	1.64e-05	3,565	4,019	4.95	1.23e-10	9.86e-09	1.63e-03
6ucd	8,199	2,733	97,910	1.31	52.10	86,618,396	86,621,926	278.99	1.35e+00	1.88e+00	9,391	11,180	11.51	9.92e-01	9.29e-09	1.10e+00
6veh	7,431	2,477	182,676	2.98	42.54	4,087,012	4,087,391	16.46	9.99e-11	1.43e-05	658	662	1.37	3.50e-11	1.38e-07	5.83e-06
6vk2	4,704	1,568	108,520	4.42	11.94	463,141	463,360	1.79	9.99e-11	9.46e-06	3,069	3,419	3.95	1.40e-10	9.62e-09	2.22e-03
6vnz	1,392	464	25,364	11.81	0.32	88,484	88,530	0.29	9.99e-11	3.40e-06	206	208	0.06	6.02e-11	2.16e-07	4.36e-06
6vv6	7,464	2,488	92,878	1.50	45.42	7,583,602	7,584,292	33.37	9.99e-11	4.04e-06	1,846	1,943	2.05	6.23e-02	1.00e-08	2.23e-01
6vv7	7,452	2,484	93,102	1.51	44.46	7,692,189	7,692,886	41.06	9.99e-11	3.80e-06	1,981	2,233	2.26	1.07e-01	7.39e-09	2.23e-01
6vv9	7,452	2,484	92,506	1.50	44.29	6,251,096	6,251,759	29.79	9.99e-11	3.11e-06	1,022	1,040	1.14	7.95e-02	1.25e-09	2.17e-01
6wcr	11,766	3,922	151,164	0.98	158.07	118,185,068	118,209,777	421.33	1.67e-01	3.74e-01	7,460	8,498	13.92	1.99e-01	9.96e-09	2.38e-01
6yuc	8,637	2,879	107,258	1.29	60.03	17,737,727	17,738,988	117.20	3.77e-01	6.86e-01	1,450	1,668	1.91	1.09e-04	9.93e-09	1.19e-01
6z4c	5,838	1,946	72,838	1.92	18.87	1,951,179	1,951,337	5.00	9.99e-11	2.72e-06	369	370	0.31	6.24e-11	2.43e-07	2.91e-06
6zcm	7,899	2,633	102,030	1.47	46.05	12,338,443	12,339,335	40.78	9.99e-11	2.53e-05	7,445	8,812	9.41	1.38e-10	9.93e-09	9.87e-04
7ckj	4,731	1,577	59,608	2.40	10.24	2,681,962	2,682,163	11.61	9.99e-11	3.27e-06	768	784	0.54	6.51e-07	5.85e-09	2.03e-01
7jjl	9,690	3,230	125,976	1.21	83.34	77,851,519	77,854,837	303.65	4.82e-01	4.52e-01	4,830	5,451	7.49	7.94e-01	8.12e-09	4.73e-01

Table 3: Performance of Coordinate Descent and SPG methods in the 46 instances that consider protein atoms only.

NMR data. Moreover, we extended the CD approach to the computation of a suitable initial approximation that avoids, in many cases, the convergence to local non-global minimizers. Our choice of the most adequate parameter p, that defines the approximating models, and the strategy for choosing the groups of variables were dictated by theoretical considerations discussed in Section 5 and by the specific characteristics of the problem. Our computing results are fully reproducible and the codes are available in `http://www.ime.usp.br/~egbirgin/`.

In future works we will apply the new CD techniques to the case in which data uncertainty is present and outliers are likely to occur. Possible improvements also include the choice of different models at each iteration or at each group of variables with the aim of making a better use of current information.

Molecule	n	n_p	$\|S\|$	Time x^0	Coordinate descent					Spectral Projected Gradient					
					#iter	$\#f$	Time	$f(x^*)$	$E(x^*)$	#iter	$\#f$	Time	$f(x^*)$	$\|\nabla f(x^*)\|_\infty$	$E(x^*)$
6kbq	8,106	2,702	108,066 (1.48%)	50.64	6,476,123	6,476,435	17.57	9.99D-11	2.35D-05	571	580	0.71	9.97D-11	6.32D-08	1.85D-05
6kc2	8,397	2,799	111,198 (1.42%)	56.89	15,115,487	15,118,534	50.50	9.99D-11	2.15D-01	1,102	1,121	1.42	4.03D-10	9.36D-09	2.19D-01
6khu	3,465	1,155	44,672 (3.35%)	4.24	334,415	334,473	0.87	9.99D-11	6.35D-06	490	500	0.25	6.20D-11	1.63D-06	7.45D-06
6kir	4,203	1,401	54,634 (2.79%)	7.42	560,278	560,406	1.49	9.99D-11	4.20D-06	450	454	0.28	9.28D-11	4.63D-07	6.28D-05
6kk9	17,895	5,965	225,776 (0.63%)	515.33	202,467,161	202,477,953	691.77	4.00D-01	3.09D-01	5,084	5,637	13.87	3.82D-01	9.95D-09	3.70D-01
6kki	8,364	2,788	111,596 (1.44%)	56.23	2,049,210	2,049,408	5.51	9.99D-11	3.93D-06	592	594	0.76	9.92D-11	1.13D-07	5.68D-05
6kkj	7,992	2,664	106,472 (1.50%)	49.62	17,809,921	17,816,613	132.85	9.54D-04	2.81D-01	2,921	3,188	3.67	9.54D-04	9.12D-09	2.81D-01
6kkl	8,421	2,807	114,184 (1.45%)	58.35	6,082,607	6,082,924	16.57	9.99D-11	5.67D-06	521	537	0.70	6.88D-11	2.32D-07	6.82D-06
6kkv	14,820	4,940	364,756 (1.49%)	304.81	27,357,834	27,358,998	113.43	9.99D-11	1.80D-05	1,193	1,220	5.01	7.35D-10	7.99D-09	4.74D-05
6kx0	10,044	3,348	131,624 (1.17%)	94.64	45,983,700	45,985,561	226.68	7.16D-02	2.47D-01	1,044	1,096	1.61	1.78D-01	9.27D-09	1.01D+00
6kys	5,886	1,962	77,926 (2.03%)	20.28	816,852	816,930	2.19	9.99D-11	3.60D-06	2,389	2,688	2.22	9.98D-11	3.11D-08	1.72D-04
6l29	5,841	1,947	77,602 (2.05%)	19.25	1,807,474	1,807,572	18.75	9.99D-11	1.26D-01	589	592	0.52	5.31D-04	4.39D-09	1.60D-01
6l2a	2,643	881	32,644 (4.21%)	1.82	787,753	787,881	4.57	9.99D-11	5.77D-06	1,769	1,892	0.67	1.79D-10	9.03D-09	1.24D-01
6laf	8,535	2,845	106,084 (1.31%)	57.40	38,880,745	38,890,853	174.71	1.57D-02	3.40D-01	10,824	12,773	13.95	1.71D-02	9.71D-09	3.41D-01
6li7	8,520	2,840	114,120 (1.42%)	57.40	6,141,620	6,141,906	16.33	9.99D-11	2.29D-05	443	453	0.58	9.37D-11	5.62D-08	1.50D-05
6lik	8,208	2,736	108,364 (1.45%)	51.66	12,686,358	12,686,897	52.81	9.99D-11	2.48D-05	7,235	8,363	9.40	2.99D-03	9.91D-09	1.81D-01
6ltz	3,798	1,266	47,246 (2.95%)	5.62	767,531	767,612	1.93	9.99D-11	7.56D-06	985	1,002	0.54	9.73D-11	5.68D-08	1.28D-04
6m5n	6,687	2,229	88,996 (1.79%)	29.41	2,380,904	2,381,175	6.33	9.99D-11	1.41D-01	14,921	17,730	15.90	6.98D-10	9.55D-09	1.42D-01
6m6j	840	280	15,602 (19.97%)	0.09	18,319	18,319	0.06	9.99D-11	1.86D-06	76	78	0.01	9.94D-11	3.25D-07	2.06D-06
6m6k	828	276	15,388 (20.27%)	0.09	15,964	15,964	0.05	9.99D-11	1.63D-06	74	76	0.01	1.00D-11	3.36D-07	7.45D-07
6pq0	6,360	2,120	89,018 (1.98%)	0.12	1,086,898	1,086,994	12.19	9.99D-11	3.98D-06	3,540	4,285	3.81	9.00D-11	1.21D-07	6.60D-05
6pup	4,272	1,424	55,442 (2.74%)	7.63	593,071	593,135	1.55	9.99D-11	3.60D-06	363	372	0.23	3.28D-11	6.55D-07	2.12D-06
6pxf	5,661	1,887	76,154 (2.14%)	17.47	6,753,157	6,775,873	25.50	9.97D-11	1.10D-01	4,541	5,069	4.07	1.04D-05	9.49D-09	1.16D-01
6t1z	9,492	3,164	127,488 (1.27%)	79.93	3,427,330	3,427,847	9.17	9.99D-11	1.14D-02	7,688	9,017	11.70	2.94D-10	9.82D-09	1.23D-02
6tad	5,172	1,724	71,178 (2.40%)	13.52	1,050,635	1,050,702	2.89	9.99D-11	3.64D-06	273	275	0.22	9.83D-11	3.61D-08	3.95D-06
6twe	7,905	2,635	163,694 (2.36%)	45.84	5,768,235	5,768,740	20.83	9.99D-11	8.19D-06	637	643	1.17	9.14D-11	1.78D-07	8.01D-06
6ubh	9,999	3,333	130,348 (1.17%)	91.62	22,598,823	22,602,027	107.35	9.99D-11	3.60D-01	1,234	1,279	1.90	6.89D-11	3.25D-07	3.63D-01
6veh	7,566	2,522	186,586 (2.93%)	42.40	4,241,879	4,242,258	17.14	9.99D-11	1.52D-05	381	384	0.80	8.74D-11	2.80D-07	4.54D-05
6vv6	8,169	2,723	107,072 (1.44%)	52.54	2,824,276	2,824,520	7.62	9.99D-11	9.75D-06	391	395	0.49	7.43D-11	8.50D-08	3.05D-06
6vv7	8,247	2,749	109,326 (1.45%)	52.54	3,807,178	3,807,395	10.23	9.99D-11	2.33D-06	4,434	5,002	5.83	5.68D-11	3.59D-07	2.89D-04
6vv9	8,250	2,750	108,784 (1.44%)	52.41	3,494,562	3,494,812	9.30	9.99D-11	8.41D-06	404	409	0.51	6.69D-11	2.00D-07	7.49D-06
6wcr	12,225	4,075	157,032 (0.95%)	167.40	229,852,446	229,885,273	721.26	9.77D-05	8.68D-02	6,972	7,906	13.17	8.93D-02	9.97D-09	1.97D-01
6yuc	8,640	2,880	107,366 (1.29%)	59.34	50,987,720	50,988,983	203.00	1.09D-04	1.19D-01	1,879	1,942	2.37	1.09D-04	7.65D-09	1.19D-01
6z4c	5,994	1,998	76,300 (1.91%)	20.35	1,697,728	1,697,882	4.45	9.99D-11	2.95D-06	270	271	0.24	9.71D-11	2.20D-07	3.12D-06
6zcm	9,696	3,232	135,448 (1.30%)	83.59	6,978,627	6,978,943	19.30	9.99D-11	7.42D-06	2,131	2,288	3.43	9.67D-11	5.81D-08	1.55D-04
7ckj	5,199	1,733	68,624 (2.29%)	13.55	1,630,610	1,630,694	10.52	9.99D-11	4.11D-06	541	549	0.43	2.14D-05	9.60D-09	1.96D-01
7jjl	9,693	3,231	126,082 (1.21%)	83.84	102,233,476	102,237,339	369.96	2.46D-01	3.25D-01	5,332	6,097	8.12	7.94D-01	8.91D-09	4.73D-01

Table 4: Performance of Coordinate Descent and SPG methods in the 37 instances that consider protein atoms plus atoms in small molecules.

References

[1] F. J. Aragón Artacho, J. M. Borwein, and M. K. Tam, Douglas-Rachford feasibility methods for matrix completion problems, *The ANZIAM Journal* 55, pp. 299–326, 2014.

[2] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality, *Mathematics of Operations Research* 38, pp. 438–457, 2010.

[3] A. Beck and L. Tetruashvili, On the convergence of block coordinate descent methods, *SIAM Journal on Optimization* 23, pp. 2037–2060, 2013.

[4] S. Bellavia, G. Gurioli, B. Morini, and Ph. L. Toint, Adaptive regularization algorithms with inexact evaluations for nonconvex optimization, *SIAM Journal on Optimization* 29, pp. 2881-2915, 2019.

[5] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, and S. A. Santos, On the use of third-order models with fourth-order regularization for unconstrained optimization, *Optimization Letters*, to appear (DOI: 10.1007/s11590-019-01395-z).

[6] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models, *Mathematical Programming* 163, pp. 359–368, 2017.

[7] E. G. Birgin and J. M. Martínez, Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101–125, 2002.

[8] E. G. Birgin and J. M. Martínez, On regularization and active-set methods with complexity for constrained optimization, *SIAM Journal on Optimization* 28, pp.1367–1395, 2018.

[9] E. G. Birgin and J. M. Martínez, A Newton-like method with mixed factorizations and cubic regularization for unconstrained minimization, *Computational Optimization and Applications* 73, pp. 707–753, 2019.

[10] E. G. Birgin and J. M. Martínez, Complexity and performance of an Augmented Lagrangian algorithm, *Optimization Methods and Software*, to appear (DOI: 10.1080/10556788.2020.1746962).

[11] E. G. Birgin, J. M. Martínez, and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.

[12] E. G. Birgin, J. M. Martínez, and M. Raydan, Algorithm 813: SPG - software for convex-constrained optimization, *ACM Transactions on Mathematical Software* 27, pp. 340–349, 2001.

[13] J. Bolte, S. Sabach and M. Teboulle, Proximal alternating linearized minimization of nonconvex and nonsmooth problems, *Mathematical Programming* 146. pp. 1–36, 2014.

[14] C. A. Bouman and K. Sauer, A unified approach to statistical tomography using coordinate descent optimization, *IEEE Transactions on Image Processing* 5, pp. 480–492, 1996.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning* 3, pp. 1–122, 2011.

[16] C. P. Bras, J. M. Martínez, and M. Raydan, Large-scale unconstrained optimization using separable cubic modeling and matrix-free subspace minimization, *Computational Optimization and Applications* 75, pp. 169–205, 2020.

[17] P. Breheny and J. Huang, Coordinate descent algorithms for nonconvex penalized regression with applications to biological feature selection, *Annals of Applied Statistics* 5, pp. 232–252, 2011.

[18] H. Calandra, S. Gratton, E. Riccietti, and X. Vasseur, On high-order multilevel optimization strategies, arXiv preprint arXiv:1904.04692, 2019.

[19] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Adaptive cubic regularization methods for unconstrained optimization. Part I: motivation motivation, convergence and numerical results, *Mathematical Programming* 127, pp. 245–295, 2011.

[20] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function and derivative complexity, *Mathematical Programming* 130, pp. 295–319, 2011.

[21] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Universal regularization methods - varying the power, the smoothness and the accuracy, *SIAM Journal on Optimization* 29, pp. 595–615, 2019.

[22] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Improved worst-case evaluation complexity for potentially rank-deficient nonlinear least-Euclidean-norm problems using higher-order regularized models, Technical Report naXys-12-2015, Namur Center for Complex Systems, University of Namur, Belgium, 2015.

[23] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Worst-case evaluation complexity for regularization models for smooth unconstrained optimization using Hölder-continuous gradients, *Optimization Methods and Software* 6, pp. 1273–1298, 2017.

[24] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization, *Foundations of Computational Mathematics* 18, pp. 1073–1107, 2018.

[25] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints, *SIAM Journal on Optimization* 30, pp. 513–541, 2020.

[26] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions, arXiv preprint arXiv:2001.10802, 2020.

[27] A. A, Canutescu and R. L. Dunbrack, Cyclic coordinate descent: a robotics algorithm for protein loop closure, *Protein Science* 12, pp. 963–972, 2003.

[28] X. Chen and Ph. L. Toint, High-order evaluation complexity for convexly-constrained optimization with non-lipschitzian group sparsity terms, *Mathematical Programming*, to appear (DOI:10.1007/s10107-020-01470-9).

[29] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd. ed., Chapman and Hall/CRC, New York, NY, USA, 2001.

[30] F. E. Curtis, D. P. Robinson, and M. Samadi, A trust-region algorithm with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$, *Mathematical Programming* 162, pp. 1–32, 2017.

[31] F. E. Curtis, D. P. Robinson, and M. Samadi, An inexact regularized Newton framework with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$ for nonconvex optimization, *IMA Journal of Numerical Analysis*, pp. 1–32, 2018.

[32] J. P. Dussault, ARCq: a new adaptive regularization by cubics, *Optimization Methods and Software* 33, pp. 322–335, 2018.

[33] J. Eckstein and W. Yao, Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives, *Pacific Journal of Optimization* 11, pp. 619–644, 2015.

[34] H.-R. Fang and D. P. O'Leary, Euclidean distance matrix completion problems, *Optimization Methods and Software* 27, pp. 695–717, 2012.

[35] W. Glunt, T. L. Hayden, and M. Raydan, Molecular conformations from distance matrices, *Journal of Computational Chemistry* 14, pp. 114–120, 1993.

[36] G. N. Grapiglia and Y. Nesterov, Regularized Newton methods for minimizing functions with Hölder continuous Hessians, *SIAM Journal on Optimization* 27, pp. 478–506, 2017.

[37] G. N. Grapiglia and Y. Nesterov, Tensor methods for minimizing functions with Hölder continuous higher-order derivatives, arXiv preprint arXiv:1904.12559, 2019.

[38] G. N. Grapiglia, J-Y Yuan, and Y-X Yuan, On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization, *Mathematical Programming* 152, pp. 491–520, 2015.

[39] S. Gratton, E. Simon, and Ph. L. Toint, An algorithm for the minimization of nonsmooth nonconvex functions using inexact evaluations and its worst-case complexity, *Mathematical Programming*, to appear (DOI: 10.1007/s10107-020-01466-5).

[40] A. Griewank, *The modification of Newton's method for unconstrained optimization by bounding cubic terms*, Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1981.

[41] C. Lavor, L. Liberti, and N. Maculan, Molecular Distance Geometry Problem, in C. Floudas, P. Pardalos (eds) *Encyclopedia of Optimization*, Springer, Boston, MA, 2008, pp. 2304–2311.

[42] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, Euclidean distance geometry and applications, *SIAM Review* 56, pp. 3–69, 2014.

[43] T. Lin and M. I. Jordan, A control-theoretic perspective on optimal high-order optimization, arXiv preprint arXiv:1912.07168, 2019.

[44] Q. Lin, Z. Lu, and L. Xiao, An accelerated proximal coordinate descent method and its application to empirical risk minimization, arXiv preprint arXiv:1407.1296, 2014.

[45] J. M. Martínez, On high-order model regularization for constrained optimization, *SIAM Journal on Optimization* 27, pp. 2447–2458, 2017.

[46] J. M. Martínez and M. Raydan, Separable cubic modeling and a trust-region strategy for unconstrained minimization with impact in global optimization, *Journal of Global Optimization* 63, pp. 315–342, 2015.

[47] J. M. Martínez and M. Raydan, Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization, *Journal of Global Optimization* 68, pp. 367–385, 2017.

[48] A. Mead, Review of the development of multidimensional scaling methods, *Journal of the Royal Statistical Society. Series D (The Statistician)* 41, pp. 27-–39, 1992.

[49] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, *SIAM Journal on Optimization* 22, pp. 341–362, 2012.

[50] Y. Nesterov and B. T. Polyak, Cubic regularization of Newton's method and its global performance, *Mathematical Programming* 108, pp. 177–205, 2006.

[51] M. S. Petković, B. Neta, L. S. Petković, and J. Džunic, Multipoint methods for solving nonlinear equations: A survey, *Applied Mathematics and Computation* 226, pp. 635–660, 2014.

[52] M. J. D. Powell , On search directions for minimization algorithms, *Mathematical Programming* 4, pp. 193–201, 1973.

[53] C. W. Royer and S. J. Wright, Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization, *SIAM Journal on Optimization* 28, pp. 1448–1477, 2018.

[54] W. S. Torgerson, *Theory & Methods of Scaling*, John Wiley and Sons, New York, NY, USA, 1958.

[55] S. J. Wright, Coordinate descent methods, *Mathematical Programming* 151, pp. 3–34, 2015.

[56] `https://www.rcsb.org`, accessed on August, 14th, 2020.

[57] J. C. Yu, K. J. Webb, C. A. Bouman, and R. P. Milane, Optical diffusion tomography by iterative coordinate-descent optimization in a bayesian framework, *Journal of the Optical Society of America A* 16, pp. 2400–2412, 1999.

[58] X. Zhu, J. Han, and B. Jiang, An adaptive high-order method for finding third-order critical points of nonconvex optimization, arXiv preprint arXiv:2008.04191, 2020.