A Derivative-Free CoMirror Algorithm for Convex Optimization

Heinz H. Bauschke[*], Warren L. Hare[†], Walaa M. Moursi[‡]

July 16, 2014

Abstract

We consider the minimization of a nonsmooth convex function over a compact convex set subject to a nonsmooth convex constraint. We work in the setting of derivative-free optimization (DFO), assuming that the objective and constraint functions are available through a black-box that provides function values for *lower-*$\mathcal{C}^2$ representation of the functions. Our approach is based on a DFO adaptation of the ϵ-comirror algorithm [6]. Algorithmic convergence hinges on the ability to accurately approximate subgradients of lower-$\mathcal{C}^2$ functions, which we prove is possible through linear interpolation. We show that, if the sampling radii for linear interpolation are properly selected, then the new algorithm has the same convergence rate as the original gradient-based algorithm. This provides a novel global rate-of-convergence result for nonsmooth convex DFO with nonsmooth convex constraints. We conclude with numerical testing that demonstrates the practical feasibility of the algorithm and some directions for further research.

Keywords: convex optimization, derivative-free optimization, lower-$\mathcal{C}^2$, approximate subgradient, Non-Euclidean projected subgradient, Bregman distance.
2010 Mathematics Subject Classification: Primary 90C25, 90C56; Secondary 49M30, 65K10.

1 Introduction

Derivative-free optimization (DFO) is a rapidly growing field of research that explores the minimization of a black-box function when first-order information (derivatives, gradients, or subgradients) is unavailable. While the majority of past work in DFO has focused on unconstrained optimization, several methods have recently been introduced for constrained optimization. In constrained optimization, most of the analysis of DFO methods has been

[*]Mathematics. Irving K. Barber school, University of British Columbia, Kelowna, B.C. V1V 1V7, Canada. Heinz.Bauschke@ubc.ca.

[†]Mathematics. Irving K. Barber school, University of British Columbia, Kelowna, B.C. V1V 1V7, Canada. Warren.Hare@ubc.ca.

[‡]Mathematics. Irving K. Barber school, University of British Columbia, Kelowna, B.C. V1V 1V7, Canada. Walaa.Mursi@ubc.ca.

done within the framework of *direct search* and *pattern search* methods. That is, methods that do not attempt to build interpolation (or other such) models of the objective function, but instead use concepts like positive bases to ensure convergence. Such methods can be adapted to constrained optimization through techniques such as projecting search directions onto constraint sets [27, 22], "pulling back" search directions onto manifolds [16, 17], the use of filtering techniques [3], barrier based penalties [4], or merit functions [21].

On the other hand, fairly little research has explored approaching constrained optimization via model-based DFO methods. Some notable advances in this area include [39, 40, 34, 20, 26]. Research in [39, 40] extends the UOBYQA algorithm (see [35]) to constrained optimization in an algorithm named CONDOR. The BOBYQA algorithm in [34] provides an interpolation-based trust region technique in the case of linear constraints. Similar techniques for bound constraints is given in [20]. In [26] a sequential penalty approach is proposed. Common to all of these methods is the underlying assumption that the objective and constraint functions are smooth (i.e., twice continuously differentiable, $\mathcal{C}^2$).

In this paper we provide a novel model-based DFO method for problems of the form

$$\min \Big\{ f(x) : g(x) \leq 0, x \in X \Big\}, \tag{P}$$

where f and g are continuous convex functions defined on a nonempty open convex subset O of $\mathbb{R}^m$, and the constraint set X is a nonempty compact convex subset of O. Unlike past model based methods, we do not assume f or g are smooth functions. However, in order to work with nonsmoothness in a DFO setting, we do assume that we have access to lower-$\mathcal{C}^2$ representations of f and g.

Recall that a function f is called lower-$\mathcal{C}^2$, if it can be represented as a maximum over an appropriate (potentially infinite) set of smooth functions indexed by a compact set (a formal definition is given in Definition 2.1). It is easily shown that all convex and $\mathcal{C}^2$ functions are lower-$\mathcal{C}^2$, as are all finite max functions (functions defined by the maximum of a finite number of smooth functions) and a wide variety of nonconvex functions [38, §10]. Lower-$\mathcal{C}^2$ functions, in particular finite max functions, have recently arisen in DFO theory [25, 23, 24] and applications [9]. (While lower-$\mathcal{C}^2$ representations are most commonly available for finite max functions, in this research we chose to maintain the broadest assumption possible.) To work with lower-$\mathcal{C}^2$ functions, we develop and analyze a method to approximate subgradients for such functions. In particular, in Theorem 3.2 we define the approximate subgradient for an arbitrary lower-$\mathcal{C}^2$ function and prove that it satisfies an error bound analogous to the one introduced in [12, Theorem 2.11] for smooth functions.

The algorithm developed in this paper is based on the ϵ-comirror algorithm (developed and named in [6]). The ϵ-comirror algorithm finds its roots in mirror-descent methods [31, 8, 7] and the earlier subgradient projection method in [33]. The ϵ-comirror is designed to find an ϵ-feasible and an ϵ-optimal solution of (P). These methods can be viewed as nonlinear projected subgradient methods that use a general distance-like function (the Bregman distance) instead of the usual Euclidean squared distance [7]. The ϵ-comirror algorithm adapts the mirror-descent method to work for convex constrained optimization where the constraint set is provided by a convex function.

We present a global rate-of-convergence result that quantifies the difference between the function values of the iterates and the optimal function value. Recent research has

established similar results for other classes of functions. For example, Vicente [41] has derived a global rate-of-convergence for DFO (direct search) algorithms for the case of smooth nonconvex function. Vicente and Garmanjani [18], as well as Nesterov [32], have established a global rate-of-convergence for DFO methods in the nonsmooth nonconvex case. Vicente and Dodangeh [15] established a global rate-of-convergence for the smooth convex case. To our knowledge, this paper represents the first global rate-of-convergence for DFO of a nonsmooth convex objective with nonsmooth convex constraints. Moreover, provided that the linear interpolation subroutines are properly controlled, the DFO algorithm herein has the same convergence result as the original gradient-based algorithm presented in [6].

The remainder of this paper is organized as follows. Section 2 is a brief introduction to the main building blocks we use. First, we provide the formal definition of lower-$\mathcal{C}^2$ functions and some of their important properties. Second, we provide the definition of the linear interpolation model of a function f over a subset Y of $\mathbb{R}^m$ and a sufficient condition to be well-defined. Finally, we give the definition and the main properties of Bregman distances. In Section 3 we give the first key result, Theorem 3.2, on which we build our convergence results. In Section 4 we describe our derivative-free ϵ-comirror algorithm, which seeks an ϵ-feasible and an ϵ-optimal solution of (P) by way of DFO. In Theorem 4.3 we establish the convergence analysis. In Section 5 we provide some numerical results that confirm the practical feasibility of the algorithm. Section 6 provides some concluding remarks.

2 Auxiliary Results

We shall work in $\mathbb{R}^m$, equipped with the usual Euclidean norm $\|\cdot\|$. Throughout the remainder of the paper, we suppose that

$$O \text{ is a nonempty open convex subset of } \mathbb{R}^m.$$

Recall that for a convex function $f : O \to \mathbb{R}$, the subdifferential ∂f at a point $x \in O$ is defined by

$$\partial f(x) = \Big\{ v \in \mathbb{R}^m : f(y) \geq f(x) + \langle v, y - x \rangle \text{ for all } y \in O \Big\}. \tag{2.1}$$

We denote the *closed* ball in $\mathbb{R}^m$ centred at x_0 with radius $\Delta > 0$ by

$$B(x_0; \Delta) = \Big\{ x \in \mathbb{R}^m : \|x - x_0\| \leq \Delta \Big\},$$

and the set of *natural numbers* by

$$\mathbb{N} = \Big\{ 1, 2, 3, \dots \Big\}.$$

Given $r \in \mathbb{N}$, we abbreviate the *unit simplex* in $\mathbb{R}^r$ by

$$S_r := \Big\{ \lambda \in \mathbb{R}^r : \sum_{i=1}^{r} \lambda_i = 1, \lambda_i \in [0, 1], i \in \{1, \dots, r\} \Big\}.$$

For a set $S \subseteq \mathbb{R}^m$ we shall use $|S|$ to denote the cardinality of S. Finally, we shall use $\|L\|$ to denote the norm of a matrix $L \in \mathbb{R}^{m \times m}$:

$$\|L\| = \max_{\|x\|=1} \|Lx\|.$$

2.1 Lower-$\mathcal{C}^k$ Functions

We next formally review the class of lower-$\mathcal{C}^k$ functions.

Definition 2.1. [38, Definition 10.29] *A function $f : O \to \mathbb{R}$ is said to be a* lower-$\mathcal{C}^k$ function *at $\bar{x} \in O$ if there exists a neighbourhood $V = V(\bar{x}) \subseteq O$ and a representation*

$$f(x) = \max_{t \in T} f_t(x) \tag{2.2}$$

in which all functions f_t are of class $\mathcal{C}^k$ on V, the index set $T := T(\bar{x})$ is a compact topological space, and f_t and the first k derivatives of f_t depend continuously not just on $x \in V$ but even on $(t, x) \in T \times V$. In this case we say that (2.2) *provides a* lower-$\mathcal{C}^k$ representation *of f at $x \in O$. The function f is said to be* lower-$\mathcal{C}^k$ on O *if f is lower-$\mathcal{C}^k$ at every point $x \in O$.*

We note that (2.2) is well defined. Indeed, by the definition of lower-$\mathcal{C}^k$ functions we know that f_t is continuous on $(t, x) \in T \times V$. Since $T := T(\bar{x})$ is a compact topological space by assumption, one can easily see that $\max_{t \in T} f_t(x)$ exists. We shall use $A(\bar{x})$ to denote the *active set of f at $\bar{x}$* defined as

$$A(\bar{x}) = \operatorname*{argmax}_{t \in T} f_t(\bar{x}). \tag{2.3}$$

The next Lemma provides details regarding when a convex function is lower-$\mathcal{C}^2$.

Lemma 2.2. [38, Theorem 10.33] *Let $f : O \to \mathbb{R}$ be convex. Then f is lower-$\mathcal{C}^2$ on O.*

It should be noted that our algorithm will require access to a lower-$\mathcal{C}^2$ representation of the objective and constraint functions. The next example shows that any finite max function is not only lower-$\mathcal{C}^2$, but also provides a natural lower-$\mathcal{C}^2$ representation.

Example 2.3. *Let $f : O \to \mathbb{R}$ be defined as $f = \max\{f_1, \ldots, f_n\}$, where each f_i is of class $\mathcal{C}^k$ on O. Then f is lower-$\mathcal{C}^k$ on O and $\max\{f_1, \ldots, f_n\}$ is a lower-$\mathcal{C}^2$ representation of the function. (This is the case where T is $\{1, \ldots, n\}$ equipped with the discrete topology.)*

The value of working with lower-$\mathcal{C}^2$ functions is seen in Lemma 2.4, which demonstrates how to compute the subdifferential of a lower-$\mathcal{C}^2$ function.

Theorem 2.4. [38, Theorem 10.31] Let $f : O \to \mathbb{R}$ be a convex function that has a lower-$\mathcal{C}^2$ representation $f(x) = \max_{t \in T} f_t(x)$ at $\bar{x} \in O$. Then

$$\partial f(\bar{x}) = \operatorname{conv}\Big\{\nabla f_t(\bar{x}) \ : \ t \in A(\bar{x})\Big\}.$$

Theorem 2.5. [38, Proposition 10.54] *Let $f : O \to \mathbb{R}$ be a lower-$\mathcal{C}^2$ function on O, and let X be a nonempty compact subset of O. Then there exists an open set O' with $X \subseteq O' \subseteq O$, such that f has a common lower-$\mathcal{C}^2$ representation valid at all points $x \in O'$, i.e., there exists a compact topological space T, and a family of functions $(f_t)_{t \in T}$ defined on O' such that*

$$f = \max_{t \in T} f_t \quad \text{on} \quad O', \tag{2.4}$$

and the functions $(t, x) \mapsto f(t, x)$, $(t, x) \mapsto \nabla f(t, x)$, and $(t, x) \mapsto \nabla^2 f(t, x)$ are continuous on $T \times O'$.

To prove convergence of the algorithm introduced in this paper, we require bounds on the subgradients of the objective and the constraint functions. Lemma 2.6 provides a proof of the existence of this bound.

Lemma 2.6. *Let $f : O \to \mathbb{R}$ be convex, and let X be a nonempty compact subset of O. Then*

$$\sup \left| \partial f(X) \right| < +\infty.$$

Proof. Since f is convex, Lemma 2.2 implies that f is lower-$\mathcal{C}^2$ on O. Since X is a nonempty compact subset of O, Theorem 2.5 guarantees the existence of an open subset O' with $X \subseteq O' \subseteq O$ such that f has a common lower-$\mathcal{C}^2$ representation valid at all points $x \in O'$. Let $f = \max_{t \in T} f_t$ be as stated in Theorem 2.5. The definition of lower-$\mathcal{C}^2$ implies that the mapping $(t, x) \mapsto \|\nabla f_t(x)\|$ is continuous on $T \times O'$. By the Weierstrass Theorem, $L := \max_{(t,x) \in T \times X} \|\nabla f_t(x)\| < +\infty$. Now, let $x \in X$, and let $v \in \partial f(x)$. Using Lemma 2.4 we know that $v = \sum_{t \in A(x)} \lambda_t \nabla f_t(x)$ for some $\lambda \in S^r$ where $r \in \mathbb{N}$ is the number of elements in $A(x)$. Therefore

$$\|v\| = \left| \sum_{t \in A(x)} \lambda_t \nabla f_t(x) \right| \leq \sum_{t \in A(x)} \lambda_t \|\nabla f_t(x)\| \leq \sum_{t \in A(x)} \lambda_t L = L,$$

and the proof is complete. (Alternatively, one may consider either the lower semicontinuous hull of f and apply [37, Theorem 24.7], or use [38, Corollary 12.38] after extending ∂f to a maximally monotone operator.) □

Lemma 2.7. *Let $f : O \to \mathbb{R}$ be a lower-$\mathcal{C}^2$ function on O, and let X be a nonempty compact convex subset of O. Let O', T, and $(f_t)_{t \in T}$ be as in Theorem 2.5. Then there exists $K_f \geq 0$ such that ∇f_t is K_f-Lipschitz on O' for every $t \in T$.*

Proof. By Theorem 2.5, $(t, x) \mapsto \nabla^2 f_t(x)$ is continuous on the compact set $T \times X$. Therefore, by the Weierstrass theorem, $K_f := \max_{(t,x) \in T \times X} \|\nabla^2 f_t(x)\| < +\infty$. Now apply the Mean Value Theorem [14, Theorem 5.1.12].

□

2.2 Linear Interpolation

In our method we use a derivative-free model-based technique. Therefore, in this section we introduce the definition of the linear interpolation model and related facts.

Definition 2.8. *Let $f : O \to \mathbb{R}$ be a function, and let $Y = (y_0, y_1, \ldots, y_m) \in \mathbb{R}^{m \times (m+1)}$. If the matrix*

$$Q = \begin{pmatrix} 1 & y_{0,1} & \cdots & y_{0,m} \\ 1 & y_{1,1} & \cdots & y_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_{m,1} & \cdots & y_{m,m} \end{pmatrix}$$

is invertible, then Y is said to be a poised tuple centred at y_0. *Moreover, if $\{y_0, y_1, \ldots, y_m\} \subseteq O$ then Y is said to be* a poised tuple centred at y_0 with respect to f. *In this case the linear system*

$$\begin{pmatrix} 1 & y_{0,1} & \cdots & y_{0,m} \\ 1 & y_{1,1} & \cdots & y_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_{m,1} & \cdots & y_{m,m} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} f(y_0) \\ f(y_1) \\ \vdots \\ f(y_m) \end{pmatrix}$$

has a unique solution $(\alpha_0, \alpha_1, \ldots, \alpha_m) \in \mathbb{R}^{m \times (m+1)}$, and the linear interpolation model *of the function f over Y is the unique (well defined) function*

$$F\colon \mathbb{R}^m \to \mathbb{R}\colon x \mapsto \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i.$$

Note that in this case F satisfies the interpolation conditions

$$F(y_i) = f(y_i), \quad \textit{for every} \quad i \in \Big\{0, 1, \ldots, m\Big\}.$$

The following theorem provides the error bound satisfied by the approximate gradient of the linear interpolation model.

Theorem 2.9. [12, Theorem 2.11] *Suppose that $f : O \to \mathbb{R}$ is $\mathcal{C}^1$ function on O. Let $y_0 \in O$. Assume that $Y = (y_0, y_1, \ldots, y_m) \in \mathbb{R}^{m \times (m+1)}$ is a poised tuple of sample points centred at y_0 with respect to f.*

Set $\Delta = \max\limits_{1 \le i \le m} \|y_i - y_0\|$. Suppose that $B(y_0; \Delta) \subseteq O$. Suppose ∇f is K_f Lipschitz over $B(y_0; \Delta)$. Then the gradient of the linear interpolation model F satisfies an error bound of the form

$$\Big|\nabla f(y) - \nabla F(y)\Big| \le K\Delta, \quad \textit{for all} \quad y \in B(y_0; \Delta),$$

where

$$K := K_f\Big(1 + \sqrt{m}\|\hat{L}^{-1}\|/2\Big), \quad L = L(Y) := \begin{pmatrix} y_1 - y_0 \\ y_2 - y_0 \\ \vdots \\ y_m - y_0 \end{pmatrix}, \text{ and } \hat{L} = \hat{L}(Y) := \frac{1}{\Delta} L. \quad (2.5)$$

2.3 The Bregman Distance

The last building block used in our analysis is the Bregman distance.

Definition 2.10. [10] *Let $\omega : O \to \mathbb{R}$ be a convex differentiable function. The corresponding* Bregman distance D_ω *is*

$$D_\omega\colon O \times O \to \mathbb{R}\colon (u, v) \mapsto \omega(u) - \omega(v) - \langle \nabla\omega(v), u - v\rangle. \quad (2.6)$$

Remark 2.11. *Notice that the Bregman distance, which shall be used in the algorithm, requires knowledge of $\nabla\omega$. Nonetheless, we refer to this research as derivative free optimization, as the objective function and constraint function are treated as black-boxes that only return function values. That is, the objective and constraint functions are 'derivative free'. In application, the function ω will be selected by the user, and it is assumed that the user will select ω such that the gradient is readily available.*

Definition 2.12. [43, Section 3.5] *Let C be a nonempty convex subset of $\mathbb{R}^m$. Let $\omega : C \to \mathbb{R}$. Then ω is said to be* strongly convex *with convexity parameter $\alpha > 0$, if for all $x, y \in C$, $\lambda \in [0, 1]$ we have*

$$\omega(\lambda x + (1-\lambda)y) \leq \lambda\omega(x) + (1-\lambda)\omega(y) - \frac{\alpha}{2}\lambda(1-\lambda)\|x-y\|^2.$$

Throughout the next arguments we shall assume that ω is a strongly convex and differentiable function on a nonempty convex subset of $\mathbb{R}^m$, with a convexity parameter $\alpha > 0$. In this paper we shall be interested in Bregman distances that are created from strongly convex functions.

When ω is differentiable, the following well-known alternative characterization of strong convexity is often used (see [33, Theorem 2.1.9], or the more generalized version in [43, Section 3.5]).

Lemma 2.13. [33, Theorem 2.1.9] *Let $\omega : O \to \mathbb{R}$ be a differentiable function. Let X be a nonempty subset of O. Then the following are equivalent:*

(i) $\omega(\lambda x + (1-\lambda)y) \leq \lambda\omega(x) + (1-\lambda)\omega(y) - \frac{\alpha}{2}\lambda(1-\lambda)\|x-y\|^2$ *for all $x, y \in X$ and $\lambda \in [0, 1]$.*

(ii) $D_\omega(x, y) = \omega(x) - \omega(y) - \langle\nabla\omega(y), x-y\rangle \geq \frac{\alpha}{2}\|x-y\|^2$ *for all $x, y \in X$ and $\lambda \in [0, 1]$.*

Following [6], we give the definition of the Bregman diameter of an arbitrary set X.

Definition 2.14. *Let $\omega : O \to \mathbb{R}$ be a convex differentiable function. Let X be a nonempty subset of O. The* Bregman diameter of the set X *is defined as*

$$\Theta = \sup\Big\{D_\omega(u, v) : u, v \in X\Big\}. \tag{2.7}$$

In the following lemma we prove that, if ω is differentiable and strongly convex, then the Bregman diameter is finite for every compact subset of $\mathbb{R}^m$.

Lemma 2.15. *Let $\omega : O \to \mathbb{R}$ be a differentiable convex function. Let X be a nonempty compact subset of O. Then D_ω is bounded on $X \times X$. Consequently, the Bregman diameter of the set X is finite.*

Proof. Since ω is convex and differentiable, therefore ω is continuously differentiable on O [37, Corollary 25.5.1]. Thus, ω and $\nabla\omega$ are continuous on X, and therefore D_ω is continuous on $X \times X$. Now, $X \times X$ is a nonempty compact subset of $\mathbb{R}^m \times \mathbb{R}^m$, and therefore D_ω is bounded on $X \times X$ and the Bregman diameter of the set X is finite. □

3 Functional Constraints and Assumptions

Recall that we are interested in the general convex problem of the form

$$\min \Big\{ f(x) : g(x) \leq 0, x \in X \Big\}. \tag{P}$$

In the sequel, we shall consider the following assumptions on f, g and X.

A1 $f : O \to \mathbb{R}$ and $g : O \to \mathbb{R}$ are continuous convex functions.

A2 X is a nonempty compact convex subset of O, and X is not a singleton.

A3 We have access to lower-$\mathcal{C}^2$ representations of f and g on some open subset O' of O such that $X \subseteq O'$ and

$$f = \max_{t \in T_f} f_t \qquad \text{and} \qquad g = \max_{t \in T_g} g_t \quad \text{on } O'.$$

A4 The set of optimal solutions of problem (P) is nonempty.

In examining our assumptions, let us make some remarks. Assumption **A1** and **A4** are both fairly standard within convex optimization literature, as is the assumption that X is a nonempty compact convex set. The assumption that X is not a singleton is necessary to guarantee that the Bregman diameter of the set X, defined in (2.7), is nonzero. While not standard, it is clear that if X is a singleton, then optimization is unnecessary.

Regarding Assumption **A3**, we note that under Assumption **A1**, the functions f and g are lower-$\mathcal{C}^2$ functions on O (by Lemma 2.2). Assumption **A3** provides the stronger statement that we have access to lower-$\mathcal{C}^2$ representations of these functions. In the case when the objective and constrained functions are for instance finite max functions, these representations are clearly accessible. However, for an arbitrary lower-$\mathcal{C}^2$ functions this assumption could be more restrictive. We maintain this assumption to provide the broadest class of functions for which the algorithm is well defined.

In the gradient-based version of this algorithm [6] the authors make the standing assumption that the subdifferentials of both the objective and the constrained functions are bounded. The next lemma shows that our base assumptions imply this result.

Lemma 3.1. *Suppose that Assumptions* **A1** *and* **A2** *hold. Then*

$$L_f := \sup \Big| \partial f(X) \Big| < +\infty \quad \textit{and} \quad L_g := \sup \Big| \partial g(X) \Big| < +\infty. \tag{3.1}$$

Proof. Combine Assumption **A2**, and Lemma 2.6. □

We are now ready to define and provide an error bound for approximate subgradients under Assumptions **A1**-**A4**. In the following, we use $A(y)^r$ to denote the Cartesian product of $A(y)$ with itself r times:

$$A(y)^r = A(y) \times A(y) \times \ldots \times A(y).$$

Theorem 3.2. *Suppose that* **A1**, **A2**, **A3**, *and* **A4** *hold. Let* $Y = (y_0, y_1, \ldots, y_m) \in \mathbb{R}^{m\times(m+1)}$ *be a poised tuple of sample points centred at* $y_0 \in X$ *with respect to* f. *Set* $\Delta = \max\limits_{1\le i\le m} \|y_i - y_0\|$. *Suppose that* $B(y_0;\Delta) \subseteq X$. *Let* $y \in B(y_0;\Delta)$. *Select a finite integer* $r(y) \le |A(y)|$. *Let* $(t_1, \ldots, t_{r(y)}) \in A(y)^{r(y)}$, *and* $\lambda \in S_{r(y)}$. *Define* $V = V(y) := \sum_{i=1}^{r(y)} \lambda_i \nabla F_{t_i}(y)$. *Then there exists* $v \in \partial f(y)$ *such that the following error bound holds:*

$$\|V - v\| \le K_f(1 + \sqrt{m}\|\hat{L}^{-1}\|/2)\ \Delta,$$

where K_f *is as in Lemma 2.7, and* $\hat{L} = \hat{L}(Y)$ *is as defined in Theorem 2.9.*

Proof. Recall that by assumption we have $V = \sum_{i=1}^{r} \lambda_i \nabla F_{t_i}(y)$. Lemma 2.4 implies that $v = v(y) := \sum_{i=1}^{r} \lambda_i \nabla f_{t_i}(y) \in \partial f(y)$. Using the triangle inequality, the error bound given in Theorem 2.9 (applied to O' instead of O) and Lemma 2.7, we have

$$\begin{aligned}\|V - v\| &= \Big|\sum_{i=1}^{r} \lambda_i \Big(\nabla F_{t_i}(y) - \nabla f_{t_i}(y)\Big)\Big| \le \sum_{i=1}^{r} \lambda_i \Big|\nabla F_{t_i}(y) - \nabla f_{t_i}(y)\Big| \\ &\le \sum_{i=1}^{r} \lambda_i K_f \Big(1 + \sqrt{m}\|\hat{L}^{-1}\|/2\Big) = K_f\Big(1 + \sqrt{m}\|\hat{L}^{-1}\|/2\Big)\ \Delta,\end{aligned}$$

as claimed. □

Our next corollary relates Theorem 3.2 to the algorithm presented later. Let us note that the function E and the nonnegative parameter ϵ in Corollary 3.3 are the same as in the algorithm. We also note that, although in Corollary 3.3 we provide the error bound for the approximate gradient function in a general format, in practice we shall use $x = y_0$.

Corollary 3.3. *Suppose that* **A1**, **A2**, **A3** *and* **A4** *hold. Let* $\epsilon \ge 0$. *Let* $Y = (y_0, y_1, \ldots, y_m)$ *be a poised tuple of sample points centred at* $y_0 \in X$ *with respect to* f. *Set* $\Delta = \max\limits_{1\le i\le m} \|y_i - y_0\|$ *and suppose that* $B(y_0;\Delta) \subseteq X$. *For every* $x \in B(y_0;\Delta)$, *select a finite integers* $r(x) \le |A_f(x)|$ *and* $\bar{r}(x) \le |A_g(x)|$. *Let* $\big(t_1, \ldots, t_{r(x)}\big) \in A_f(x)^{r(x)}$, $\lambda \in S_{r(x)}$, $\big(\bar{t}_1, \ldots, \bar{t}_{\bar{r}(x)}\big) \in A_g(x)^{\bar{r}(x)}$, $\bar{\lambda} \in S_{\bar{r}(x)}$,

$$v_f(x) = \sum_{i=1}^{r(x)} \lambda_i \nabla f_{t_i}(x) \in \partial f(x), \qquad V_f(x) = \sum_{i=1}^{r(x)} \lambda_i \nabla F_{t_i}(x),$$

$$v_g(x) = \sum_{i=1}^{\bar{r}(x)} \bar{\lambda}_i \nabla g_{\bar{t}_i}(x) \in \partial g(x), \qquad V_g(x) = \sum_{i=1}^{\bar{r}(x)} \bar{\lambda}_i \nabla G_{\bar{t}_i}(x),$$

and

$$e(x) := \begin{cases} v_f(x), & \text{if } g(x) \le \epsilon, \\ v_g(x), & \text{otherwise,} \end{cases} \tag{3.2}$$

and

$$E(x) := \begin{cases} V_f(x), & \text{if } g(x) \le \epsilon \\ V_g(x), & \text{otherwise.} \end{cases} \tag{3.3}$$

Then:

(i) *The following error bound holds*

$$\Big|e(x) - E(x)\Big| \leq \kappa\ \Delta, \quad \textit{for all} \quad x \in B(y_0; \Delta), \tag{3.4}$$

where $\kappa = \max\{K_f, K_g\}(1 + \sqrt{m}\|\hat{L}^{-1}\|/2)$, K_f *is defined as in Lemma 2.7 and* K_g *is obtained by replacing* f *by* g *in Lemma 2.7, and* $\hat{L}$ *is as defined in Theorem 2.9.*

(ii) *The function* E *induced by* (3.3) *satisfies*

$$\Big|E(x)\Big| \leq \max\Big\{L_f, L_g\Big\} + \kappa\ \Delta, \quad \textit{for all} \quad x \in B(y_0; \Delta), \tag{3.5}$$

where L_f *and* L_g *are defined as in Lemma 3.1.*

Proof. (i): Use (3.2) and (3.3), and apply Theorem 3.2 to f and g. (ii): Let $x \in X$. Using the triangle inequality, (3.1), and (3.4) we have $\|E(x)\| \leq \|e(x)\| + \|e(x) - E(x)\| \leq \max\{L_f, L_g\} + \kappa\ \Delta$. □

4 Algorithm and Discussion

In this section we introduce the Derivative-Free ϵ-comirror algorithm and present a convergence analysis. We remind the reader that the algorithm is designed to find an ϵ-feasible and an ϵ-optimal solution of (P).

The Derivative-Free ϵ-CoMirror algorithm (DFO$_\epsilon$CM)

Initialization Input

- $\epsilon \geq 0$,
- $x_0 \in X$,
- $M > 0$.

General step for every $k \in \Big\{1, 2, \dots\Big\}$

- Select
$$0 < \Delta_k \leq \frac{1}{\sqrt{k+1}}\ . \tag{4.1}$$
- Select a poised tuple $Y_k = (y_0, y_1, \dots, y_m)$ centred at y_0 with respect to f such that the set $\{y_0, y_1, \dots, y_m\} \subseteq B(x_k, \Delta_k)$, $x_k = y_0$ and $\|\hat{L}_k^{-1}\| \leq M$, where $\hat{L}_k = \hat{L}(Y_k)$ is as defined in Theorem 2.9.
- Set
$$x_{k+1} = \operatorname*{argmin}_{x \in X}\{\langle t_k E_k - \nabla\omega(x_k), x\rangle + \omega(x)\}, \tag{4.2}$$
where
$$E_k := \begin{cases} V_f(x_k), & \text{if } g(x_k) \leq \epsilon; \\ V_g(x_k), & \text{otherwise,} \end{cases} \tag{4.3}$$

$$t_k = \frac{\sqrt{\Theta\alpha}}{\|E_k\|\sqrt{k}}\,, \tag{4.4}$$

and where $\alpha > 0$ is the strong convexity parameter of the strongly convex function $\omega\colon O \to \mathbb{R}$, Θ is the corresponding Bregman diameter of the set X, and V_f and V_g are defined as in Corollary 3.3.

Before proceeding to convergence analysis, we provide some remarks on the DFO$_\epsilon$CM.

Remark 4.1.

(i) *In generating the points of the tuple $Y_k \subseteq \mathbb{R}^{m\times(m+1)}$ we need to check that $\|\hat{L}_k^{-1}\| \leq M$. If this inequality fails, then we resample. It is always possible to generate the tuple Y_k for all $k \in \mathbb{N}$ provided that M is set to be sufficiently large* [42]. *For a detailed discussion on how to choose M we refer the reader to* [13].

(ii) *The poised tuple $Y_k = (y_0, y_1, \ldots, y_m)$ must satisfy $\max_{i\in\{1,\ldots,m\}} \|y_i - x_k\| \leq \Delta_k$ to guarantee that the error bound in Theorem 3.2 still holds true. This does not create a conflict* (i) *because by the definition of the matrix $\hat{L}$ in* (2.5)*, the value of $\|\hat{L}^{-1}\|$ remains unchanged under scaling or shifting.*

(iii) *The update of x_k in* (4.2) *is well defined, since the function $\langle t_k E_k - \nabla\omega(x_k), \cdot\rangle + \omega$ is strongly convex and differentiable over X, and therefore it has a unique minimizer over X.*

(iv) *The step length t_k is well defined for all $k \in \{1, 2, \ldots\}$ except when $E_k = 0$ in which case either we have an ϵ-feasible minimizer, or we change the search radius Δ_k to get a better approximation of the gradients. Moreover, the Bregman diameter Θ is finite by Lemma 2.15. Finally, by Lemma 2.13* (ii)*, we have that $D_\omega(x, y) \geq \frac{\alpha}{2}\|x - y\|^2$, and therefore, since X is not a singleton, the Bregman diameter Θ is strictly positive.*

(v) *In general, the Bregman diameter Θ is not easy to calculate. However, if the set X is simple and the function ω is separable, calculating Θ becomes simpler. For example, if $X = [\alpha_1, \beta_1] \times \cdots \times [\alpha_m, \beta_m]$ and $\omega(x) = \sum_{i=1}^m \omega_i(x_i)$, then $\Theta = \sum_{i=1}^m D_{\omega_i}(\alpha_i, \beta_i)$.*

4.1 Convergence Analysis

We devote this subsection to study the convergence of the algorithm. We begin with the following technical lemma, which provides a key tool in the proof of convergence. Lemma 4.2 and its proof are an adaptation of [6, Lemma 2.2]. For the sake of completeness, we include the adapted proof.

Lemma 4.2. *Let $(x_k)_{k\in\mathbb{N}}$ be the sequence generated by DFO$_\epsilon$CM. Let $i < j$ be two strictly positive integers. Then for all $k \in \{1, 2, \ldots\}$*

$$\sum_{k=i}^{j} t_k\langle E_k, x_k - u\rangle \leq \Theta + \frac{1}{2\alpha}\sum_{k=i}^{j} t_k^2\|E_k\|^2, \tag{4.5}$$

for every $u \in X$.

Proof. By the optimality condition in (4.2) we have

$$\langle t_k E_k - \nabla\omega(x_k) + \nabla\omega(x_{k+1}), u - x_{k+1}\rangle \geq 0 \text{ for every } u \in X.$$

Hence,

$$t_k\langle E_k, u - x_{k+1}\rangle \geq \langle \nabla\omega(x_k) - \nabla\omega(x_{k+1}), u - x_{k+1}\rangle \text{ for every } u \in X. \tag{4.6}$$

The three-point property of the Bregman distance [11, Lemma 3.1] tells us

$$D_\omega(u, x_{k+1}) - D_\omega(u, x_k) + D_\omega(x_{k+1}, x_k) = \langle \nabla\omega(x_k) - \nabla\omega(x_{k+1}), u - x_{k+1}\rangle. \tag{4.7}$$

Combining (4.6) and (4.7) yields

$$t_k\langle E_k, u - x_{k+1}\rangle \geq D_\omega(u, x_{k+1}) - D_\omega(u, x_k) + D_\omega(x_{k+1}, x_k).$$

That is

$$t_k\langle E_k, x_{k+1} - u\rangle \leq D_\omega(u, x_k) - D_\omega(x_{k+1}, x_k) - D_\omega(u, x_{k+1}).$$

Adding $t_k\langle E_k, x_k - x_{k+1}\rangle$ to both sides of the above inequality and using Lemma 2.13 (ii) and the Cauchy-Schwarz inequality we get

$$\begin{aligned} t_k\langle E_k, x_k - u\rangle &\leq D_\omega(u, x_k) - D_\omega(u, x_{k+1}) - D_\omega(x_{k+1}, x_k) + t_k\langle E_k, x_k - x_{k+1}\rangle \\ &\leq D_\omega(u, x_k) - D_\omega(u, x_{k+1}) - \frac{\alpha}{2}\|x_k - x_{k+1}\|^2 + t_k\|E_k\|\|x_k - x_{k+1}\|. \end{aligned}$$

Notice that, $t_k\|E_k\|\|x_k - x_{k+1}\| - \frac{\alpha}{2}\|x_k - x_{k+1}\|^2$ is a quadratic function of $\|x_k - x_{k+1}\|$ that has a maximum value of $\frac{1}{2\alpha}t_k^2\|E_k\|^2$, i.e., $t_k\|E_k\|\|x_k - x_{k+1}\| - \frac{\alpha}{2}\|x_k - x_{k+1}\|^2 \leq \frac{1}{2\alpha}t_k^2\|E_k\|^2$. This yields

$$t_k\langle E_k, x_k - u\rangle \leq D_\omega(u, x_k) - D_\omega(u, x_{k+1}) + \frac{1}{2\alpha}t_k^2\|E_k\|^2.$$

Summing the last inequality over $k \in \{i, i+1, \ldots, j\}$ we obtain

$$\sum_{k=i}^{j} t_k\langle E_k, x_k - u\rangle \leq D_\omega(u, x_i) - D_\omega(u, x_{j+1}) + \sum_{k=i}^{j}\frac{1}{2\alpha}t_k^2\|E_k\|^2.$$

Using the definition of Θ and the Bregman distance we note that $0 \leq D_\omega(u, x_i) \leq \Theta$, and $0 \leq D_\omega(u, x_{j+1}) \leq \Theta$. Therefore, $D_\omega(u, x_i) - D_\omega(u, x_{j+1}) \leq \Theta$, from which we get (4.5). □

The following theorem presents the efficiency estimate for DFO$_\epsilon$CM. In proving Theorem 4.3 we are motivated by the techniques used in the proof of [6, Theorem 2.1]. Given $n \in \mathbb{N}$, we denote the set of indices of the ϵ-feasible solutions among the first n iterations by

$$I_n^\epsilon = \Big\{k \in \{1, 2, ..., n\} : g(x_k) \leq \epsilon\Big\}.$$

Theorem 4.3. *Suppose that Assumptions* **A1**, **A2**, **A3** *and* **A4** *hold. Let* $\epsilon > 0$ *and let* $(x_k)_{k\in\mathbb{N}}$ *be the sequence generated by DFO*$_\epsilon$*CM. Denote by* f_{opt} *the optimal function value of* (P)*. Then for every* $n \in \{6, 7, \ldots\}$

$$\min\left\{\min_{k\in I_n^\epsilon}\left(f(x_k) - f_{\text{opt}}\right), \epsilon\right\} \leq \frac{C}{\sqrt{n}},$$

where

$$C = 2\sqrt{\frac{\Theta}{\alpha}}\max\left\{\kappa_1, \kappa_2\right\}\frac{1+\ln(2)}{2-\sqrt{2}} + \kappa_2\ \Omega,$$

$$\kappa_1 = \max\left\{L_f, L_g\right\},$$

$$\kappa_2 = K(1+\sqrt{m}M/2),$$

$$\Omega = \max_{x,y\in X}\|x-y\|,$$

L_f *and* L_g *are as defined in* (3.1)*,* $K = \max\{K_f, K_g\}$*, where* K_f *and* K_g *are as defined in Lemma 2.7 (applied to* f *and* g *respectively), and* $M > 0$ *satisfies that* $\|\hat{L}_k^{-1}\| \leq M$ *for all* $k \in \{1, 2, \ldots\}$*.*

Proof. Using assumption **A4**, suppose that x_{opt} is an optimal solution of (P). Fix $n \in \{1, 2, \ldots\}$, and $k \in \{1, 2, \ldots, n\}$. We begin by considering the following two cases:

Case I: $k \in I_n^\epsilon$. Then $g(x_k) \leq \epsilon$, and, by (4.3), (3.2), and (3.3) we have $e_k := e(x_k) = v_f(x_k) \in \partial f(x_k)$ and $E_k := E(x_k) = V_f(x_k)$, and hence

$$f(x_k) \leq f(x_{\text{opt}}) + \langle e_k, x_k - x_{\text{opt}}\rangle. \tag{4.8}$$

Therefore, using Cauchy-Schwarz inequality and the error bound in equation (3.4)

$$\begin{aligned} f(x_k) &\leq f(x_{\text{opt}}) + \langle E_k, x_k - x_{\text{opt}}\rangle + \langle e_k - E_k, x_k - x_{\text{opt}}\rangle \\ &\leq f(x_{\text{opt}}) + \langle E_k, x_k - x_{\text{opt}}\rangle + \|e_k - E_k\|\|x_k - x_{\text{opt}}\| \\ &\leq f(x_{\text{opt}}) + \langle E_k, x_k - x_{\text{opt}}\rangle + \kappa_2\ \Delta_k\ \Omega. \end{aligned}$$

Hence

$$f(x_k) - f(x_{\text{opt}}) \leq \langle E_k, x_k - x_{\text{opt}}\rangle + \kappa_2\ \Delta_k\ \Omega. \tag{4.9}$$

Case II: $k \notin I_n^\epsilon$. Then $g(x_k) > \epsilon$. Using (4.3), (3.2), and (3.3) we have $e_k = v_g(x_k) \in \partial g(x_k)$ and $E_k = V_g(x_k)$, and hence

$$g(x_k) \leq g(x_{\text{opt}}) + \langle e_k, x_k - x_{\text{opt}}\rangle. \tag{4.10}$$

Since $g(x_{\text{opt}}) \leq 0$ we have

$$\begin{aligned} \epsilon &< g(x_k) \\ &\leq g(x_{\text{opt}}) + \langle e_k, x_k - x_{\text{opt}}\rangle \\ &\leq \langle e_k, x_k - x_{\text{opt}}\rangle = \langle E_k, x_k - x_{\text{opt}}\rangle + \langle e_k - E_k, x_k - x_{\text{opt}}\rangle. \end{aligned}$$

Hence, using Cauchy-Schwarz inequality, the assumption that $\|\hat{L}_k^{-1}\| \leq M$ for all $k \in \{1, 2, \ldots\}$, and the error bound in equation (3.4) we have

$$\begin{aligned} \epsilon &\leq \langle E_k, x_k - x_{\text{opt}} \rangle + \|e_k - E_k\| \|x_k - x_{\text{opt}}\| \\ &\leq \langle E_k, x_k - x_{\text{opt}} \rangle + \kappa_2 \ \Delta_k \ \Omega. \end{aligned} \tag{4.11}$$

By combining Case I and Case II, we have

$$\langle E_k, x_k - x_{\text{opt}} \rangle + \kappa_2 \ \Delta_k \ \Omega \geq \begin{cases} f(x_k) - f(x_{\text{opt}}), & \text{if } \ k \in I_n^\epsilon, \\ \epsilon, & \text{if } \ k \notin I_n^\epsilon. \end{cases} \tag{4.12}$$

Using (4.12) we have for all $1 \leq l \leq n$, with $\Delta_l \leq 1/\sqrt{l+1}$

$$\min\{\min_{k \in I_n^\epsilon} \left(f(x_k) - f(x_{\text{opt}})\right), \epsilon\} \leq \langle E_l, x_l - x_{\text{opt}} \rangle + \kappa_2 \ \Delta_l \ \Omega.$$

Let $n_0 \in \{1, 2, \ldots, n\}$, then using (4.1)

$$\begin{aligned} \min \left\{ \min_{k \in I_n^\epsilon} \left(f(x_k) - f(x_{\text{opt}})\right), \epsilon \right\} &\leq \min_{n_0 \leq l \leq n} \left(\langle E_l, x_l - x_{\text{opt}} \rangle + \kappa_2 \ \Delta_l \ \Omega \right) \\ &\leq \min_{n_0 \leq l \leq n} \left(\langle E_l, x_l - x_{\text{opt}} \rangle + \kappa_2 \ \Omega \max_{n_0 \leq l \leq n} \Delta_l \right) \\ &\leq \min_{n_0 \leq l \leq n} \left(\langle E_l, x_l - x_{\text{opt}} \rangle \right) + \frac{\kappa_2 \ \Omega}{\sqrt{n_0 + 1}}. \end{aligned} \tag{4.13}$$

Substituting $u = x_{\text{opt}}, \ i = n_0, \ j = n$ in Lemma 4.2 we see that

$$\sum_{k=n_0}^{n} t_k \langle E_k, x_k - x_{\text{opt}} \rangle \leq \Theta + \frac{1}{2\alpha} \sum_{k=n_0}^{n} t_k^2 \|E_k\|^2. \tag{4.14}$$

On the other hand, since X is not a singleton, we have $t_k > 0$ for every $k \in \{1, 2, \ldots, n\}$, and thus

$$\sum_{k=n_0}^{n} t_k \langle E_k, x_k - x_{\text{opt}} \rangle \geq \left(\min_{n_0 \leq k \leq n} \langle E_k, x_k - x_{\text{opt}} \rangle \right) \sum_{k=n_0}^{n} t_k. \tag{4.15}$$

Combining (4.14) and (4.15) yields

$$\min_{n_0 \leq k \leq n} \langle E_k, x_k - x_{\text{opt}} \rangle \leq \frac{\Theta + \frac{1}{2\alpha} \sum_{k=n_0}^{n} t_k^2 \|E_k\|^2}{\sum_{k=n_0}^{n} t_k}. \tag{4.16}$$

Using (4.4), we have

$$\sum_{k=n_0}^{n} t_k^2 \|E_k\|^2 = \Theta \alpha \sum_{k=n_0}^{n} \frac{1}{k}, \tag{4.17}$$

and

$$\sum_{k=n_0}^{n} t_k = \sqrt{\Theta\alpha} \sum_{k=n_0}^{n} \frac{1}{\|E_k\|\sqrt{k}}. \tag{4.18}$$

We recall that $\|\hat{L}_k^{-1}\| \leq M$ for all $k \in \{1, 2, \ldots\}$, $\kappa_1 = \max\{L_f, L_g\}$ and $\kappa_2 = K(1 + \sqrt{m}M/2)$. Now, for every $k \in \{1, 2, \ldots\}$ using Corollary 3.3 and (4.1) we have

$$\begin{aligned}
\|E_k\|\sqrt{k} &\leq (\kappa_1 + \kappa_2\ \Delta_k)\sqrt{k} \leq \kappa_1\sqrt{k} + \kappa_2 \frac{\sqrt{k}}{\sqrt{k+1}} \\
&\leq \kappa_1\sqrt{k} + \kappa_2 \leq \max\Big\{\kappa_1, \kappa_2\Big\}\Big(\sqrt{k} + 1\Big) \\
&\leq 2\max\Big\{\kappa_1, \kappa_2\Big\}\sqrt{k}.
\end{aligned} \tag{4.19}$$

Using (4.18) and (4.19) we get

$$\sum_{k=n_0}^{n} t_k \geq \frac{\sqrt{\Theta\alpha}}{2\max\Big\{\kappa_1, \kappa_2\Big\}} \sum_{k=n_0}^{n} \frac{1}{\sqrt{k}}, \tag{4.20}$$

Using equations (4.17) and (4.20), inequality (4.16) becomes

$$\min_{n_0 \leq l \leq n} \langle E_k, x_k - x_{\text{opt}} \rangle \leq \frac{2\Theta\max\Big\{\kappa_1, \kappa_2\Big\}\left(1 + \frac{1}{2}\sum_{k=n_0}^{n} \frac{1}{k}\right)}{\sqrt{\Theta\alpha}\sum_{k=n_0}^{n} \frac{1}{\sqrt{k}}}. \tag{4.21}$$

Now, set $n_0 = \lfloor n/2 \rfloor$. On the one hand, using (4.21) and Lemma A.1 we get

$$\min_{n_0 \leq l \leq n} \langle E_l, x_l - x_{\text{opt}} \rangle \leq \frac{C_1}{\sqrt{n}}, \tag{4.22}$$

where $C_1 = 2\sqrt{\frac{\Theta}{\alpha}}\max\{\kappa_1, \kappa_2\}\,\frac{1 + \ln(2)}{2 - \sqrt{2}}$. On the other hand, using the fact that $\lfloor n/2 \rfloor + 1 > n/2$ we have

$$\frac{\kappa_2\ \Omega}{\sqrt{n_0 + 1}} = \frac{\kappa_2\ \Omega}{\sqrt{\lfloor n/2 \rfloor + 1}} \leq \frac{C_2}{\sqrt{n}}, \tag{4.23}$$

where $C_2 = \sqrt{2}\ \kappa_2\ \Omega$. Using (4.13), (4.22) and (4.23) we deduce that

$$\min\Big\{\min_{k \in I_n^\epsilon} f(x_k) - f(x_{\text{opt}}), \epsilon\Big\} \leq \frac{C_1 + C_2}{\sqrt{n}} = \frac{C}{\sqrt{n}}, \tag{4.24}$$

which completes the proof. □

Remark 4.4. *In DFO convergence rates are often provided in terms of number of function evaluations, as apposed to number of iterations.*

If f and g are considered single black-boxes that return the full lower-$\mathcal{C}^2$ representation at each function call (i.e., a single function call to f returns a list of all function values f_t, $t \in T$), then the number of function evaluations per iteration is easily computed. In particular, one evaluation of g is required to determine if V_f or V_g is required in the creation of E_k. Then $m+1$ evaluations of f or g are required to generate V_f or V_g as required (where $O \subseteq \mathbb{R}^m$).

On the other hand, if f and g are considered flexible black-boxes, where individual sub-functions f_{t_i} or g_{t_i} can be called independently, then the number of function evaluations becomes more complicated to compute. In particular, the subgradients will be approximated using $(m+1)(r)$ sub-function calls, where r is the number of elements selected from the active set of the appropriate function.

5 Numerical Testing

5.1 Test Problems and Solvers

The DFO$_\epsilon$CM algorithm was implemented in MATLAB and tested using a benchmark set of 126 randomly generated test problems. The test set was created as follows.

For a fixed dimension d_p, we create a finite max function composed of N_f randomly generated quadratics of the form

$$Q_i : x \mapsto x^\top A_i x + C_i, \quad i \in \Big\{1, 2, \ldots, N_f\Big\}$$

where A_i is a random $d_p \times d_p$ positive definite symmetric matrix with entries in $[0, 1]$, and C_i is a random value in $[-1, 0[$. As all such functions are minimized at 0, the finite max function

$$F = \max_{i \in \{1,2,\ldots,N_f\}} \{Q_i\}$$

will have a unique minimizer at 0. We force exactly A_f of these quadratics to be active at the minimizer, by setting $C_i = 0$ for $i \in \{1, 2, \ldots, A_f\}$.

We consider three lower-$\mathcal{C}^2$ constraint functions $g_i : \mathbb{R}^{d_p} \to \mathbb{R}$, defined by

$$g_1(x) = 0,$$

$$g_2(x) = \max_{j \in \{1,\ldots,d_p\}} g_{2,j}(x), \quad \text{with} \quad \begin{array}{l} g_{2,1} = \frac{\sqrt{3}}{3} x_1 - x_2, \\ g_{2,2} = x_2 - \sqrt{3} x_1 \text{ and} \\ g_{2,j}(x) = -x_j, j \in \{3, \ldots, d_p\}, \end{array}$$

and

$$g_3(x) = \max_{j \in \{1,\ldots,\lceil d_p/2 \rceil\}} g_{3,j}(x), \quad \text{with} \quad g_{3,j}(x) = -x_j, j \in \{1, \ldots, \lceil d_p/2 \rceil\}.$$

For all tests the convex compact set X is the box

$$X := [-1, 1]^{d_p} = \Big\{x = (x_1, x_2, \cdots, x_{d_p}) \in \mathbb{R}^{d_p} : \|x_i\| \leq 1\Big\}.$$

Thus, the problems in the benchmark set are identified by the tuple (d_p, N_f, A_f, C), where d_p is the problem dimension, N_f is the number of objective sub-functions, A_f is the number of active sub-fuctions at the minimizer, and $C \in \{1, 2, 3\}$ defines the lower-$\mathcal{C}^2$ constraint function used. Our testing considered

$$\begin{aligned} d_p &\in \Big\{3, 10, 25, 50, 100, 200\Big\}, \\ N_f &\in \Big\{1, \lceil d_p/2 \rceil, d_p\Big\}, \\ A_f &\in \Big\{1, \lceil N_f/2 \rceil, N_f\Big\}. \end{aligned}$$

Note that if $N_f = 1$, then $A_f = 1$, this provides a total of 126 different randomly generated tests.

Notice that, as the minimizer of the objective is the origin, our three lower-$\mathcal{C}^2$ constraint functions can classify the functions into three distinct sets: the solution is in the interior of the constraint set, the solution is at the vertex of the constraint set, and the solution is on the edge of the constraint set. For each problem we set $\epsilon = 0$, $x_0 = (1, 1, \ldots, 1)$, and $M = d_p$.

Two DFO$_\epsilon$CM algorithms were tested; DFO$_\epsilon$CM$_1$ using the function $\omega_1(x) = \frac{1}{2}\|x\|^2$, and DFO$_\epsilon$CM$_2$ using the function $\omega_2(x) = \sum_{i=1}^{d_p}(x_i)\ln(x_i)$. For each problem, the constant M is set equal to the dimension of the problem d_p. These are compared to two other derivative free solvers; `NOMAD` [1, 2, 4] and the `patternsearch` [4, 28] MATLAB built-in solver. All solvers were set to terminate after $100d_p$ function calls.

5.2 Results: Data Profiles

We begin by providing a visual comparison of the results using data profiles introduced in [30]. For each data profile we have the set of solvers and the benchmarking set of test problems $\mathcal{P}$. We use the convergence test

$$f(x_0) - f(x) \geq (1 - \tau)(f(x_0) - f_L), \tag{5.1}$$

where f is the objective function, $\tau > 0$ is a tolerance, x_0 is the starting point for each problem $p \in \mathcal{P}$, f_L denotes the minimum value of the function f achieved by any of the solvers, with a fixed number of function evaluations. Our data profiles define $t_{p,s}$ to be the number of function evaluations required by the solver S to solve problem p. Following [30], the data profile of a solver S is defined by

$$d_S(\alpha) = \frac{1}{n_p} \operatorname{size}\left\{p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \alpha\right\}, \tag{5.2}$$

where n_p is the number of variables in problem $p \in P$.

In Figure 1, we provide data profiles that compare DFO$_\epsilon$CM$_1$, DFO$_\epsilon$CM$_2$, `NOMAD`, and `patternsearch`, using $\tau = 10^{-1}$, $\tau = 10^{-3}$ and $\tau = 10^{-5}$.

Examining the data profiles, we see that `NOMAD` provides the strongest performance on this test set. Moreover, DFO$_\epsilon$CM$_2$ outperforms DFO$_\epsilon$CM$_1$, and both of these solidly outperform `patternsearch`.

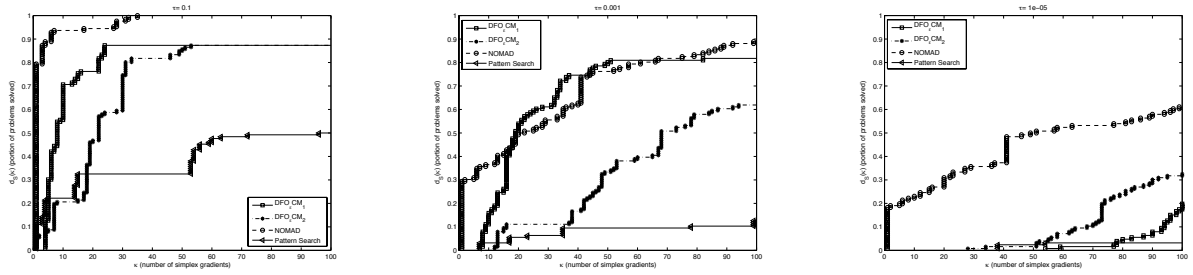


Figure 1: *Data profiles using* $\tau = 10^{-1}$ *(left),* $\tau = 10^{-3}$ *(middle) and* $\tau = 10^{-5}$ *(right).*

5.3 Results: Accuracy Profiles

To better understand the results of our testing, we next examine the data via accuracy profiles. That is, we plot the ratio of the problems whose function value has improved by a certain number of digits from the initial function value. More precisely, we define

$$t_{p,s} = \frac{f_{p,s}}{f_{p,0}},$$

where $f_{p,s}$ is the minimum function value achieved by the solver s in problem p, and $f_{p,0}$ is the initial function value at problem p. We then plot

$$\rho_s(\alpha) = \frac{1}{n_p} \text{size} \left\{ p \in \mathcal{P} : t_{p,s} \leq \tfrac{1}{2} \cdot 10^{1-\alpha} \right\},$$

where n_p is the total number of problems, and $\mathcal{P}$ is the set of problems under consideration.

Figure 2 presents the accuracy profile when all problems are considered.

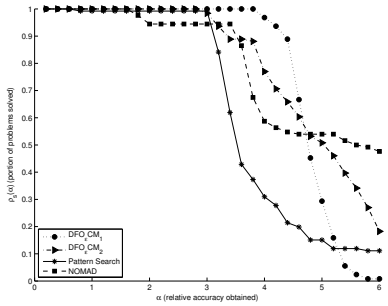


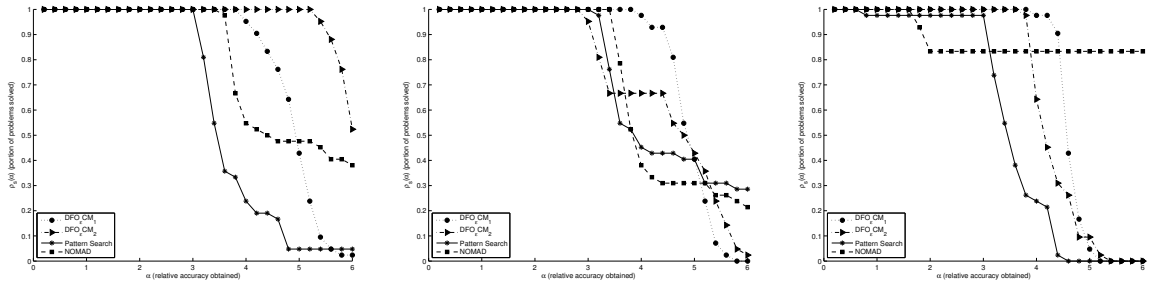
Figure 2: *Accuracy profiles comparing the results of the solvers on the entire set of test problems.*

Examining Figure 2, we note that DFO$_\epsilon$CM is very competitive up to about 4 digits of accuracy. After this, `NOMAD` becomes a clear winner, while DFO$_\epsilon$CM and `patternsearch` remain competitive.

More interesting conclusions arise when the problem set is subdivided into some simple categorizations. In Figure 3 we create the accuracy profiles based on the location of the minimizer. In particular, we group the problems into the categories:

- $\mathcal{P}_I$, problems with the solutions in the interior of the constraint set ($C = 1$),

- $\mathcal{P}_E$, problems with the solutions on the edge of the constraint set ($C = 2$), and
- $\mathcal{P}_V$, problems with the solutions at the vertex of the constraint set ($C = 3$).



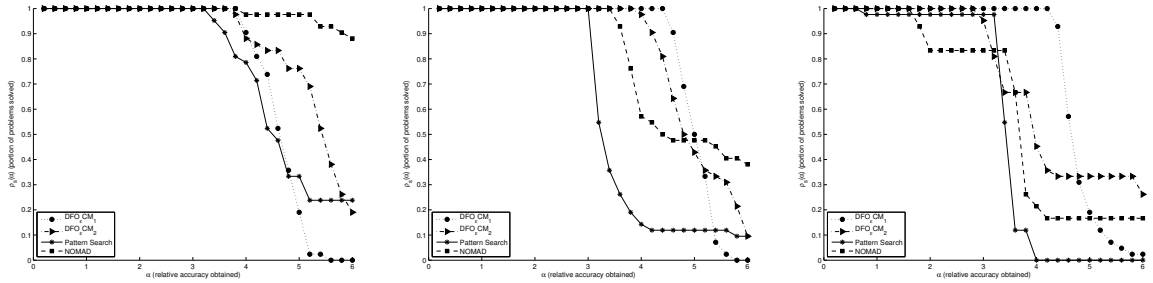
(a) Problem Set $\mathcal{P}_I$ (b) Problem Set $\mathcal{P}_E$ (c) Problem Set $\mathcal{P}_V$

Figure 3: *Accuracy profiles comparing the results of the solvers based on the location of the solution.*

Examining Figure 3 we note that `NOMAD` preforms extremely well when the solution is at a vertex. `patternsearch` provides its most accurate results when the solution is on the edge of the constraint set. The DFO$_\epsilon$CM algorithm preforms fairly consistently in all three cases.

In Figure 4 we create accuracy profiles based on the dimension of the problem. In particular, we group the problems into the categories:

- $\mathcal{P}_{ld}$, problems in low dimensions, $n_d \in \{3, 10\}$,
- $\mathcal{P}_{md}$, problems in medium dimension $n_d \in \{25, 50\}$, and
- $\mathcal{P}_{hd}$, problems in high dimensions $n_d \in \{100, 200\}$.



(a) Problem Set $\mathcal{P}_{ld}$ (b) Problem Set $\mathcal{P}_{md}$ (c) Problem Set $\mathcal{P}_{hd}$

Figure 4: *Accuracy profiles comparing the results of the solvers based on the problem dimension.*

Examining Figure 4, we notice the accuracy profile of DFO$_\epsilon$CM appears largely independent of the problem dimension. Conversely, `NOMAD` and `patternsearch` both enjoy significantly better performance in lower dimensions.

6 Conclusions

This paper addresses the development and analysis of a DFO algorithm for nonsmooth convex objective function with nonsmooth convex constraint function and a simple constraint set. The functions considered in the paper are convex with lower-$\mathcal{C}^2$ representations. We define approximate subgradients for this class of functions and provide an error bound on this approximation. The DFO$_\epsilon$CM algorithm is presented and a novel rate of convergence is introduced for this class of functions. Our numerical results suggests that the proposed algorithm is competitive with the `NOMAD` and `patternsearch`. A nice feature of our algorithm is that its performance appears to be independent of the dimension of the problem or the location of the minimizers.

Considerable future work is possible in this direction. One way to expand this work is explore methods to improve the step size which may yield more rapid or accurate results.

Another direction of future research is to examine whether the convexity assumptions on the objective function and constraint set can be relaxed. In Theorem 4.3, convexity is invoked in equations (4.8) and (4.10) to provide cutting plane approximations of f and g. While in general this is not easily extended to non-convex functions, it might be possible to achieve a similar result for prox-regular functions. Since all lower-$\mathcal{C}^2$ functions are prox-regular [38, Section 13F], this might allow the algorithm herein to be extended to a non-convex setting.

A Appendix

Lemma A.1. *For any integer $n \in \{6, 7, \dots\}$ the following inequalities hold true*

$$\sum_{k=\lfloor n/2 \rfloor}^{n} \frac{1}{k} \leq 2\ln(2), \tag{A.1}$$

$$\sum_{k=\lfloor n/2 \rfloor}^{n} \frac{1}{\sqrt{k}} \geq (2-\sqrt{2})\sqrt{n}. \tag{A.2}$$

Proof. To see inequality (A.1), notice

$$\begin{aligned}\sum_{k=\lfloor n/2 \rfloor}^{n} \frac{1}{k} &\leq \sum_{k=\lfloor n/2 \rfloor -1}^{n-1} \int_k^{k+1} \frac{1}{x} dx \\ &= \int_{\lfloor n/2 \rfloor -1}^{n} \frac{1}{x} dx \\ &= \ln\left(\frac{n}{\lfloor n/2 \rfloor - 1}\right). \end{aligned} \tag{A.3}$$

We now consider two cases (n is even and n is odd). Case I: suppose $n = 2m$ with $m \in \{1, 2, \dots\}$. Then

$$\frac{n}{\lfloor n/2 \rfloor - 1} \leq 4 \iff \frac{2m}{m-1} \leq 4 \iff n = 2m \geq 4. \tag{A.4}$$

Case II: suppose $n = 2m + 1$ with $m \in \{1, 2, \dots\}$. Then

$$\frac{n}{\lfloor n/2 \rfloor - 1} \leq 4 \iff \frac{2m+1}{m-1} \leq 4 \iff n = 2m + 1 \geq 7. \tag{A.5}$$

Combining (A.4) and (A.5) proves that the first inequality holds true for all $n \in \{6, 7, \dots\}$.

Finally,

$$\begin{aligned}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n} \frac{1}{\sqrt{k}} &= \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n} \frac{1}{\sqrt{k}}(k+1-k) \geq \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n} \int_{k}^{k+1} \frac{1}{\sqrt{x}} dx = \int_{\lfloor \frac{n}{2} \rfloor}^{n+1} \frac{1}{\sqrt{x}} dx \\ &\geq \int_{\frac{n}{2}}^{n} \frac{1}{\sqrt{x}} dx = (2 - \sqrt{2})\sqrt{n}.\end{aligned}$$

which proves inequality (A.2) □

Acknowledgments

HHB was partially supported by the Natural Sciences and Engineering Research Council of Canada and by the Canada Research Chair Program. WLH was partially supported by the Natural Sciences and Engineering Research Council of Canada and UBC Internal Research Funding. WMM was partially supported by the Natural Sciences and Engineering Research Council of Canada grants of HHB and WLH, and UBC Internal Research Funding.

The authors would like to thank the two anonymous referees for useful feedback in revising this paper.

References

[1] M.A. Abramson, C. Audet, G. Couture, J. E. Dennis, Jr., S. Le Digabel, and C. Tribes. The NOMAD Project, `http://www.gerad.ca/nomad`.

[2] M. A. Abramson, C. Audet, J. E. Dennis, Jr., and S. Le Digabel, OrthoMADS. A deterministic MADS instance with orthogonal directions. *SIAM J. Optim.*, 948–966, 20 (2009)

[3] M.A. Abramson, C. Audet, and J.E. Dennis, Jr. Filter pattern search algorithms for mixed variable constrained optimization problems. *Pac. J. Optim.*, 3(3):477–500, 2007.

[4] C. Audet and J. E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17(1):188–217 (electronic), 2006.

[5] C. Audet and J. E. Dennis, Jr. Analysis of generalized pattern searchs. *SIAM J. Optim.*, 13:889–903, 2002.

[6] A. Beck, A. Ben-Tal, N. Guttmann-Beck, and L. Tetruashvili. The CoMirror algorithm for solving nonsmooth constrained convex problems. *Oper. Res. Lett.*, 38(6):493–498, 2010.

[7] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.

[8] A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM J. Optim.*, 12(1):79–108 (electronic), 2001.

[9] K. Bigdeli, W. Hare, S. Tesfamariam. Configuration optimization of dampers for adjacent buildings under seismic excitations. *Engineering Optimization*, 44(12), 1491–1509 (2012)

[10] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. U.S.S.R. Comp. Math. Math. Phys., 7:200–217, 1967.

[11] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM J. Optim.*, 3(3):538–543, 1993.

[12] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization*, SIAM, 2009.

[13] A.R. Conn, K. Scheinberg, and L.N. Vicente. Geometry of interpolation sets in derivative free optimization. *Math. Program. (Ser. B)*, 111(1-2):141–172, 2008.

[14] Z. Denkowski, S. Migórski, and N.S. Papageorgiou. *An Introduction to Nonlinear Analysis: Theory*. Kluwer, Boston, MA, 2003.

[15] M. Dodangeh and L.N. Vicente. Worst Case Complexity for Direct-Search under convexity. Preprint 13–10, Dept. Mathematics, Univ. Coimbra.
`http://www.mat.uc.pt/~lnv/papers/cds.pdf`.

[16] D.W. Dreisigmeyer. Direct search algorithms over Riemannian manifolds. Los Alamos Technical Report LA-UR-06-7416, 2006.
`http://www.optimization-online.org/DB_FILE/2007/08/1742.pdf`.

[17] D.W. Dreisigmeyer. Equality constraints, Riemannian manifolds and direct search methods, Los Alamos Technical Report LA-UR-06-7406, 2006.
`http://www.optimization-online.org/DB_FILE/2007/08/1743.pdf`.

[18] R. Garmanjani and L.N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, volume 33, 1008–1028, 2013.

[19] N. Ghosh and W.W. Hager. A derivative-free bracketing scheme for univariate minimization. *Comput. Math. Appl.*, 20(2):23–34, 1990.

[20] S. Gratton, P. L. Toint and A. Troeltzch. An active-set trust-region method for derivative-free nonlinear bound- constrained optimization, *Optimization Methods and Software*, volume 21(4–5), 873–894, 2011.

[21] S. Gratton and L.N. Vicente. A merit function approach for direct search. Preprint 13–08, Dept. Mathematics, Univ. Coimbra.
http://www.mat.uc.pt/~lnv/papers/merit.pdf.

[22] W.L. Hare. Using derivative free optimization for constrained parameter selection in a home and community care forecasting model. In *International Perspectives on Operations Research and Health Care, Proceedings of the 34th Meeting of the EURO Working Group on Operational Research Applied to Health Sciences*, pages 61–73, 2010.

[23] W. Hare, M. Macklem. Derivative-free optimization methods for finite minimax problems. *Optimization Methods and Software*, 28(2), 300–312 (2011)

[24] W. Hare, J. Nutini. A derivative-free approximate gradient sampling algorithm for finite minimax problems. *Computational Optimization and Applications*, 56(1), 1–38 (2013)

[25] G. Liuzzi, S. Lucidi, M. Sciandrone. A derivative-free algorithm for linearly constrained finite minimax problems. *SIAM Journal on Optimization* 16, 1054–1075 (2006)

[26] G. Liuzzi, S. Lucidi, M. Sciandrone. Sequential penalty derivative-free methods for nonlinear constrained optimization. *SIAM Journal on Optimization* 20, 2814–2835 (2010)

[27] S. Lucidi, M. Sciandrone, and P. Tseng. Objective-derivative-free methods for constrained optimization. *Math. Program. (Ser. A)*, 92:37–59, 2002.

[28] Mathworks. http://www.mathworks.com/help/gads/patternsearch.html

[29] R. Mifflin. A superlinearly convergent algorithm for minimization without evaluating derivatives. *Math. Program.*, 9:100–117, 1975.

[30] J. J. Moré, S. M. Wild. Benchmarking Derivative-Free Optimization Algorithms. *SIAM J. Optim.*, 20(1), 17–191. (2009)

[31] A. S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, New York, 1983.

[32] Y. Nesterov. Random gradient-free minimization of convex functions. *Technical Report* 2011/1, CORE, 2011. http://www.ecore.be/DPs/dp_1297333890.pdf

[33] Y. Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Boston(2004).

[34] M. J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge 2009. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

[35] M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Program. (Ser. B)*, 92:555–582, 2002.

[36] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis*, 51–67. Kluwer Academic, Dordrecht, 1994.

[37] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[38] R. T. Rockafellar and R.J-B Wets. *Variational Analysis*, Springer, Berlin, 1998.

[39] F. Vanden Berghen. *CONDOR: A Constrained, Non-Linear, Derivative-Free Parallel Optimizer for Continuous, High Computing Load, Noisy Objective Functions*. PhD thesis, Université Libre de Bruxelles, Belgium, 2004.

[40] F. Vanden Berghen and H. Bersini. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: experimental results and comparison with the DFO algorithm. *J. Comput. Appl. Math.*, 181(1):157–175, 2005.

[41] L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, volume 1, issue 1-2, 143–153.

[42] M. Wschebor. Smoothed analysis of $\kappa(A)$. *J. Complexity*, 20(1):97–107, 2004.

[43] C. Zălinescu. *Convex analysis in general vector spaces*. World Scientific Publishing, River Edge, NJ, 2002.