Generalized Order-Value Optimization

José Mario Martínez*

September 22, 2010 - October 29, 2010 - December 13, 2010

Abstract

Generalized Order-Value Optimization (GOVO) problems involve functions whose evaluation depends on order relations on some representation functional set. Examples will be given of GOVO problems that may be analyzed in the context of Piecewise-Smooth Optimization. Generalizations of algorithms that have been proved to be effective for proving special classes of GOVO problems will be introduced. The case of Low Order-Value Optimization (LOVO) will be considered as an example of GOVO in which one needs specialized algorithms with stronger convergence results. Applications of constrained LOVO problems and problems with OVO constraints will be presented. The state of the art of Protein Alignment problems from the LOVO point of view will be discussed.

1 Introduction

Generalized Order-Value functions are continuous functions such that, for all x in the domain, the value of $f(x)$ depends on order relations in a set of the form $\{f_i(x)\}_{i\in I}$.

Let us give a more formal and precise definition. Assume that, for all $k = 1, \ldots, m, j = 1, \ldots, \kappa(k)$, we have $f_{kj} : \Omega \to I\!R$, where Ω is an arbitrary set. For all $k = 1, \ldots, m$ we define

$$I_k = \{1, 2, \ldots, \kappa(k)\}.$$

For all $k = 1, \ldots, m$, $x \in \Omega$, we define $i_{k,1}(x), \ldots, i_{k,\kappa(k)}(x)$ by

$$\{i_{k,1}(x), \ldots, i_{k,\kappa(k)}(x)\} = I_k$$

and

$$f_{k,i_{k,1}(x)}(x) \leq \ldots \leq f_{k,i_{k,\kappa(k)}(x)}(x).$$

*Institute of Mathematics, Statistics and Scientific Computing, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This work was supported by PRONEX, FAPESP (PT 2006-53768-0), CNPq, FAEP-UNICAMP.

We say that $f : \Omega \to I\!R$ is a Generalized Order-Value function if there exists $\Phi : I\!R^n \to I\!R$, $n = \sum_{i=1}^{n} \kappa(i)$ such that, for all $x \in \Omega$,

$$f(x) = \Phi(f_{1,i_{1,1}(x)}(x), \ldots, f_{1,i_{1,\kappa(1)}(x)}(x), \ldots\ldots, f_{m,i_{m,1}(x)}(x), \ldots, f_{m,i_{m,\kappa(m)}(x)}(x)).$$

The most obvious example is the *maximum* function given by $f(x) = \max\{f_1(x), \ldots, f_\kappa(x)\}$. In this case, one has $m = 1$, $\kappa(1) = \kappa$, $f_{1,j} = f_j$ for $j = 1, \ldots, \kappa$, $n = \kappa$ and $\Phi(z_1, \ldots, z_n) = z_n$.

Order-Value functions began to be considered with this denomination in [2], where it was observed that the maximization of the function that represents an almost-worst return under a finite number of scenarios may be useful for risky decisions and for fitting models. In the first situation, this function corresponds to the well-known Value-at-Risk measure [3, 27, 40]. In [3] and [40], using different arguments, smooth equivalent reformulations to that optimization problem were introduced. Newton and quasi-Newton primal methods for the same problem were defined and tested in [7].

The consideration of Low Order-Value Optimization (LOVO) problems in [4, 5, 6] was accompanied by the development of interesting applications. On one hand, it was pointed out that the LOVO approach is a natural generalization of classical nonlinear least-squares fitting with automatic detection of outliers in the sense of Robust Statistics [26]. On the other hand, LOVO proved to be useful for detecting common structures in different spatial objects. The application to chemical structural alignment problems motivated the development of LovoAlign [36], a free-software package for alignment of proteins with a solid mathematical basement which exhibits satisfactory practical performance. LOVO is a representative of a class of Piecewise-Smooth optimization problems for which the concept of Clarke-stationarity [14] is not satisfactory enough. As a consequence, one is forced to employ methods that provably converge to points that satisfy stronger optimality conditions.

This paper is organized as follows. In Section 2 we provide some useful definitions and we give examples of GOVO problems. In Section 3 we discuss Clarke-like optimality conditions for Piecewise-Smooth optimization. In Section 4 we generalize to the case of general Piecewise-Smooth Optimization some algorithms that proved to be useful for the first OVO problem [2]. In Section 5 we show how some GOVO problems can be reformulated as smooth optimization problems. Low Order-Value Optimization is discussed in Section 6. In Section 7 we describe a way to deal with OVO constraints of VaR type. The Protein Alignment problem and several of its variants are addressed in Section 8. Conclusions are given in Section 9.

2 Definitions and examples

Assume that I is a finite set and $\{f_i\}_{i \in I}$ is a set of continuous real-valued functions defined on $\Omega \subseteq I\!R^n$. We say that $f : \Omega \to I\!R$ is a Continuous Choice function (CCF) if f is continuous

and, for all $x \in \Omega$, there exists $i \in I$ such that $f(x) = f_i(x)$. By the continuity of f and f_i, if a sequence $\{x^k\} \subseteq \Omega$ converges to $x \in \Omega$ and there exists $i \in I$ such that $f(x^k) = f_i(x^k)$ for all k, then $f(x) = f_i(x)$.

Continuous choice functions such that, for all $i \in I$, f_i has continuous first derivatives on an open set that contains Ω are called Piecewise-Smooth functions [42]. The set of functions $\{f_i\}_{i \in I}$ is called a "representation" of f.

2.1 Examples

1. **Order-Value function**

   The simplest Generalized Order-Value function is the Order-Value function defined in [2]. Let $I = \{1, \ldots, m\}$ and, for all $x \in \Omega$, define $i_1(x), \ldots, i_m(x)$ by

   $$f_{i_1(x)}(x) \leq \ldots \leq f_{i_m(x)}(x) \tag{1}$$

   and $\{i_1(x), \ldots, i_m(x)\} = I$. Given $p \in \{1, \ldots, m\}$, we define

   $$f_o(x) = f_{i_p(x)}(x). \tag{2}$$

   The function defined by (2) was employed in [2, 3] and other papers in the context of Risk Analysis. Assume that $f_1(x), \ldots, f_m(x)$ are the predicted losses associated with the decision x under different scenarios having identical probabilities. The assumption of equal probabilities is sensible when the scenarios are computed by simulation. Then, $f_{i_p(x)}(x)$ is the maximal predicted loss corresponding to x, when we discard the $m-p$ worst scenarios. This definition evokes the classical Value-at-Risk (VaR) risk measure [27].

2. **Order-Value with weights**

   If we admit different probabilities w_i for the scenarios $1, 2, \ldots, m$ we obtain a Continuous Choice function that is even closer to VaR [40]. Let $w_1, \ldots, w_m > 0$ be such that $\sum_{i=1}^m w_i = 1$. Let $\alpha \in (0, 1)$ be a "confidence level". We define $i_1(x), \ldots, i_m(x)$ as in (1). Let $p(x) \in \{1, \ldots, m\}$ be such that

   $$w_{i_1(x)} + \ldots + w_{i_{p(x)}(x)-1}(x) \leq \alpha < w_{i_1(x)} + \ldots + w_{i_{p(x)}(x)}(x).$$

   Then, we define the order-value function with weights $f_{wo}(x) = f_{i_{p(x)}(x)}(x)$.

3. **Conditional Value-at-risk**

   Defining $i_1(x), \ldots, i_m(x)$ as in (1), and given the confidence level $\alpha = p/m$, we define

   $$f_{o+}(x) = \frac{1}{m-p} \sum_{j=p+1}^{m} f_{i_j(x)}(x). \tag{3}$$

When $f_1(x), \ldots, f_m(x)$ represent the predicted losses under the equally probable scenarios $\{1, \ldots, m\}$, $f(x)$ approximates the Conditional Value-at-risk measure (CVaR) introduced by Rockafellar and Uryasev [41] for the confidence level α. The weighted discrete form of CVaR was also considered in [40].

4. **Low Order-Value functions**

Given $p \in \{1, \ldots, m\}$ and defining $f_{low}(x) = \sum_{j=1}^{p} f_{i_j(x)}(x)$ we obtain the Low-Order Value function considered in [5, 6, 36]. Low Order-Value Optimization has been successfully used for the construction of the Protein Alignment package LovoAlign. See www.ime.unicamp.br/∼martinez/lovoalign.

5. **Clustering**

Assume that $P_1, \ldots, P_m$ are points in $\mathbb{R}^n$ which we wish to classify into q groups [9, 12]. We wish to determine $C_1, \ldots, C_q$, the "centers" of the groups, in an optimal way. The point P_i will be assigned to the group whose center is C_j if a distance-like continuous function $\varphi(P_i, C)$ takes its minimum value at C_j among the discrete set of centers $C_1, \ldots, C_q$. We assume that $\varphi(P, C) \geq 0$ for all $P, C \in \mathbb{R}^n$. Therefore, the goal is to find $C_1, \ldots, C_q$ that solves the optimization problem

$$\text{Minimize } f(C_1, \ldots, C_q) \equiv \sum_{i=1}^{m} \text{ minimum } \{\varphi(P_i, C_j), j = 1, \ldots, q\}. \tag{4}$$

6. **Multiple Order-Value functions**

Assume that $I = I_1 \cup \ldots \cup I_q$ and that the functions f_i^j are defined for all $j = 1, \ldots, q$, $i \in I_j$. Moreover, for all $j = 1, \ldots, q$, $x \in \Omega$ we define $i_\nu^j(x)$ by:

$$f^j_{i_1^j(x)}(x) \leq \ldots \leq f^j_{i^j_{m(j)}(x)}(x),$$

where $m(j)$ is the number of elements of I_j. Therefore, a partial order is introduced in the set $\{f_i^j(x), j = 1, \ldots, q, i = 1, \ldots, m(j)\}$. Function values corresponding to the same set I_j are comparable, but the ones corresponding to different sets are not. Multiple Order-Value functions are continuous functions whose evaluation at each x depends on this partial order. Consider, for example, that, in a Chemical Engineering environment, q experiments are performed, which are reported as q empirical extraction curves, where y_{j,t_k^j} is the j-th extraction observed at time t_k^j during experiment j. Suppose that we know that, for each experiment, approximately one tenth of the observations (outliers) should be discarded because of measurement errors or phenomenological reasons. However, we do not know which are the observations that should be eliminated. The theoretical curve that corresponds to all the experiments is of the form $y = \varphi(x, t)$ and the parameters x

should be the same for all the experiments. Under these conditions, a suitable objective function for fitting the parameters $x_1, \ldots, x_n$ may be:

$$f_{mul}(x) = \sum_{j=1}^{q} \sum_{k=1}^{p_j} f^j_{i^j_k(x)}(x)$$

where $p_j \approx 0.9 \times m(j)$ and, for all i, j, $f^j_i(x)$ is the squared difference between the empirical and the theoretical value of the extraction corresponding to experiment j and time t_i. Namely, $f^j_k(x) = [y_{j,t^j_k} - \varphi(x, t^j_k)]^2$.

Therefore, the minimization of $f(x)$ is a least-squares fitting problem that involves, simultaneously, all the experiments performed and eliminates automatically one tenth of (presumably wrong) observations for each extraction curve.

7. **Gini Coefficient**

Consider $f_i, i \in I$ satisfying (1) for all $x \in \Omega$. If $f_i(x)$ represents the income of an individual (or an homogeneous group of individuals) under the conditions given by $x \in \Omega$, the Gini Coefficient [20], that measures the inequality of the wealth distribution, is given by

$$f_{Gini}(x) = 1 - \frac{2}{m-1}\left(m - \frac{\sum_{j=1}^{m} j f_{i_j(x)}(x)}{\sum_{j=1}^{m} f_j(x)}\right).$$

Clearly f_{Gini} is a Generalized Order-Value function. The Gini Coefficient varies between 0 and 1. The 0 value corresponds to total equality, whereas maximal inequality is represented by Gini=1. The minimization of $f(x)$ under constrains $x \in \Omega$ corresponds to seeking political or economical decisions x that aim to reduce income inequality.

8. **Approximating the extremes**

Another popular measure of income distribution is the quotient between the income of the richest part and the poorest part of the population. As in the case of the Gini function, equalitarian policies may seek to approximate the incomes of these groups. In terms of (1) this would lead to minimize $f_Q(x) \equiv \frac{\sum_{j=m-p+1}^{m} f_{i_j(x)}(x)}{\sum_{j=1}^{p} f_{i_j(x)}(x)}$, where p is small with respect to m.

9. **Discarding extreme evaluations**

Suppose that $f_i(x)$ represents the evaluation of some product by individuals $1, \ldots, m$. The variables $x \in \Omega$ could be the possible proportions of ingredients used for its fabrication. In principle, one would like to maximize all the evaluations $f_i(x)$ or, at least, an average of them. However, extreme evaluations may affect the decisions in an undesirable way, leading to biased solutions that satisfy improbable requirements. By this reason, the

extreme values of the set $\{f_i(x)\}_{i\in I}$ should be discarded. This motivates the Generalized Order-Value function given by:

$$f_D(x) = \sum_{j=p+1}^{m-p} f_{i_j(x)}(x),$$

where $p < (m-1)/2$.

3 Optimality conditions

We consider the problem

$$\text{Minimize } f(x) \quad \text{subject to} \quad x \in \Omega, \tag{5}$$

where $\Omega \subseteq I\!R^n$ is closed and convex and f is a Piecewise-Smooth function with representation $\{f_i\}_{i\in I}$. For all $x \in \Omega$, we say that $d \in I\!R^n$ is a feasible direction if $x + d \in \Omega$.

The presence of the constraints $x \in \Omega$ is crucial in problems in which x represents a portfolio. In these cases, the present investment profile is given by a fixed portfolio $\bar{x}$ and one is interested in the computation of a new one that should not differ excessively from $\bar{x}$. Moreover, one wishes to change as few positions as possible, which makes it desirable to employ convex linear constraints defined by $\|x - \bar{x}\|_1 \leq c$.

For all $x \in \Omega$ we define

$$C(x) = \{i \in I \mid f_i(x) = f(x)\}.$$

With some abuse of notation we will write, sometimes, $f_i \in C(x)$ to indicate that $i \in C(x)$. Moreover, we will define:

$$C_-(x) = \{i \in I \mid \text{ there exists } \{x^k\} \subseteq \Omega - x \text{ such that } x^k \to x \text{ and } f(x^k) = f_i(x^k)\}.$$

By continuity of f and f_i we have that $C_-(x) \subseteq C(x)$. The reciprocal is not true. For example, take $f(x) = \max\{f_1(x), f_2(x), f_3(x)\} \equiv \max\{x, 2x, 3x\}$. If $x^* = 0$ we have that $C(x^*) = \{1, 2, 3\}$ but $C_-(x^*) = \{1, 3\}$.

Theorem 3.1 *Assume that x^* is a local minimizer of (5). Then, the set of feasible directions d such that*

$$\nabla f_i(x^*)^T d < 0 \ \text{ for all } \ i \in C_-(x^*) \tag{6}$$

is empty. The set of feasible directions d such that

$$\nabla f_i(x^*)^T d < 0 \ \text{ for all } \ i \in C(x^*) \tag{7}$$

is also empty.

Proof. Assume that d is a feasible direction such that $\nabla f_i(x^*)^T d < 0$ for all $i \in C_-(x^*)$. Therefore, there exists $t_0 > 0$ such that

$$f_i(x^* + td) < f_i(x^*) \text{ for all } t \in (0, t_0], i \in C_-(x^*).$$

Since $C_-(x^*) \subset C(x^*)$, it follows that

$$f_i(x^* + td) < f(x^*) \text{ for all } t \in (0, t_0], i \in C_-(x^*). \tag{8}$$

For all $k = 1, 2, \ldots$, define $t_k = t_0/k$. Let $i(k)$ be such that $f(x^* + t_k d) = f_{i(k)}(x^* + t_k d)$ for all $k = 1, 2, \ldots$. Since I is finite, there exists $j \in I$ such that $j = i(k)$ infinitely many times. Then, by (8), $f(x^* + t_k d) < f(x^*)$ infinitely many times. Thus, x^* is not a local minimizer. This proves (6). The proof of (7) follows using that $C_-(x^*) \subseteq C(x^*)$. □

The optimality condition based on (6) is strictly stronger than the one based on (7). To see this, take $f_1(x) \equiv x, f_2(x) \equiv 0$ and define $f(x) \equiv x$, $x^* = 0$. Since $\nabla f_2(x^*) = 0$, x^* satisfies the optimality condition based on $C(x)$. However, since $f_2 \notin C_-(x^*)$, the optimality condition based on $C_-(x)$ is not satisfied. Nevertheless, we will generally use the optimality condition based on $C(x)$ because, in practical algorithmic terms, given a point $x \in \Omega$, it could be very difficult, if not impossible, to discard choices $f_i \in C(x)$ in the cases in which f_i does not belong to the more restrictive set $C_-(x)$.

4 Basic algorithms

In this section we define three algorithms for solving (5). These algorithms aim to find points $x^* \in \Omega$ that are stationary in the sense that

$$\{d \in I\!R^n \mid x^* + d \in \Omega \text{ and } \nabla f_i(x^*)^T d < 0 \mid \text{ for all } i \in C(x^*)\} = \emptyset. \tag{9}$$

The algorithms presented here are generalizations of methods that proved to be effective for minimizing the Order-Value function (2). See [2]–[7]. Given $\delta, \varepsilon > 0$, the first two algorithms find, in finite time, a point $\bar{x}$ such that

$$\{d \in I\!R^n \mid \bar{x} + d \in \Omega \text{ and } \nabla f_i(\bar{x})^T d < -\varepsilon \text{ for all } i \in C_\delta(\bar{x})\} = \emptyset,$$

where, for all $x \in \Omega$,

$$C_\delta(x) = \{i \in I \text{ such that } |f_i(x) - f(x)| \leq \delta\}. \tag{10}$$

The limit points of sequences generated by the third algorithm are stationary in the sense of (9). The algorithms share many of the characteristics of bundle methods for nonsmooth nonconvex

problems. See [18, 19, 22, 23, 24, 29, 28, 34, 39, 44, 46] and many others. In practical Generalized Order-Value Optimization problems the strategy of using (perhaps all) the gradient information at the current approximation, discarding gradients at previous iterations, seems to be affordable and effective.

From now on we will assume that Ω is convex and compact.

Algorithm 4.1

Assume that $\varepsilon > 0$, $\delta > 0$, $\Delta > 0$, $\alpha \in (0,1)$, $0 < \sigma_{min} < \sigma_{max} < 1$, $x^0 \in \Omega$. Initialize $k \leftarrow 0$.

Step 1. Compute $d^k \in I\!R^n$ as a solution of the problem

$$\text{Minimize } \max\{\nabla f_i(x^k)^T d, \mid i \in C_\delta(x^k)\} \quad \text{subject to} \quad x^k + d \in \Omega, \ \|d\| \leq \Delta.$$

If $\max\{\nabla f_i(x^k)^T d^k, \mid i \in C_\delta(x^k)\} \geq -\varepsilon$ stop.

Step 2. Set $t = 1$.

Step 2.1. If

$$f(x^k + td^k) \leq f(x^k) - t\alpha\varepsilon, \tag{11}$$

set $t_k = t$, compute $x^{k+1} \in \Omega$ such that $f(x^{k+1}) \leq f(x^k + t_k d^k)$, set $k \leftarrow k+1$ and go to Step 1.

Step 2.2. If (11) does not hold, choose

$$t_{new} \in [\sigma_{min} t, \sigma_{max} t], \tag{12}$$

set $t \leftarrow t_{new}$ and go to Step 2.1.

Let us prove that Algorithm 4.1 is well defined.

Theorem 4.1 *Assume that $x^k \in \Omega$ is computed by Algorithm 4.1 and $\max\{\nabla f_i(x^k)^T d^k, i \in C_\delta(x^k)\} < -\varepsilon$. Then, the loop that aims to compute x^{k+1} at Step 2 finishes in finite time.*

Proof. Assume that $\max\{\nabla f_i(x^k)^T d, i \in C_\delta(x^k)\} < 0$. Then,

$$\nabla f_i(x^k)^T d^k < 0 \ \text{ for all } \ i \in C_\delta(x^k).$$

Therefore, there exists $\bar{t}_1 > 0$ such that, for all $t \in [0, \bar{t}_1]$ and $i \in C(x^k) \subseteq C_\delta(x^k)$,

$$f_i(x^k + td^k) \leq f_i(x^k) + t\alpha\nabla f_i(x^k)^T d^k \leq f(x^k) + t\alpha \max\{\nabla f_i(x^k)^T d, i \in C_\delta(x^k)\} \leq f(x^k) - t\alpha\varepsilon. \tag{13}$$

Define

$$c = \min\{|f_i(x^k) - f(x^k)| \mid i \notin C(x^k)\} > 0.$$

By the continuity of f and f_i, there exists $\bar{t}_2 \in (0, \bar{t}_1]$ such that for all $i \notin C(x^k)$ and $t \in [0, \bar{t}_2]$,

$$|f_i(x^k + td^k) - f(x^k + td^k)| \geq \frac{c}{2}.$$

Therefore, for all $i \notin C(x^k)$, $t \in [0, \bar{t}_2]$ we have that $i \notin C(x^k + td^k)$. In other words, $C(x^k + td^k) \subseteq C(x^k)$ for all $t \in [0, \bar{t}_2]$. Thus, by (13), we have that for $t \in [0, \bar{t}_2]$,

$$f(x^k + td^k) \leq f(x^k) - t\alpha\varepsilon.$$

Then, (11) holds after a finite number of reductions of the steplength. □

In the following theorem we prove that Algorithm 4.1 stops, in a finite number of iterations, at a point that satisfies $\max\{\nabla f_i(x^k)^T d, i \in C_\delta(x^k)\} \geq -\varepsilon$.

Theorem 4.2 *Assume that the sequence $\{x^k\}$ is generated by Algorithm 4.1. Then, there exists k such that* $\max\{\nabla f_i(x^k)^T d, i \in C_\delta(x^k)\} \geq -\varepsilon$.

Proof. Assume, by contradiction, that the sequence $\{x^k\}$, generated by the algorithm, has infinitely many terms. By the definition of the algorithm, we have that $\max\{\nabla f_i(x^k)^T d^k, i \in C_\delta(x^k)\} < -\varepsilon$ for all $k = 0, 1, 2, \ldots$.

Since Ω is compact, there exists $x^* \in \Omega$ such that $\lim_{k \in K} x^k = x^*$ for some subsequence K. If $f_i(x^*) = f(x^*)$ then, by continuity, $|f_i(x^k) - f(x^k)| \leq \delta$ for $k \in K$ large enough. Therefore, $C(x^*) \subseteq C_\delta(x^k)$ for $k \in K$ large enough.

If $i \notin C(x^*)$ we have that $|f_i(x^*) - f(x^*)| > 0$. Thus, there exists $c_+ > 0$ such that, for all $i \notin C(x^*)$, one has that $|f_i(x^*) - f(x^*)| > c_+ > 0$. Therefore, by the continuity of f_i and f, $|f_i(x) - f(x)| > c_+ > 0$ for all x in a neighborhood of x^*. This implies that $i \notin C(x)$ for all x in that neighborhood. Thus, for $k \in K$ large enough and $i \notin C(x^*)$, we have that $i \notin C(x^k)$. So, for $k \in K$ large enough we have that $C(x^k) \subseteq C(x^*)$.

Since $\max\{\nabla f_i(x^k)^T d^k, i \in C_\delta(x^k)\} < -\varepsilon$, we have that

$$\nabla f_i(x^k)^T d^k \leq -\varepsilon \text{ for all } i \in C_\delta(x^k).$$

In particular, since $C(x^*) \subseteq C_\delta(x^k)$,

$$\nabla f_i(x^k)^T d^k \leq -\varepsilon \text{ for all } i \in C(x^*).$$

Taking limits for an appropriate subsequence $K_1 \subseteq K$, we obtain that there exists a feasible direction d such that

$$\nabla f_i(x^*)^T d \leq -\varepsilon \text{ for all } i \in C(x^*).$$

Take $\alpha' = (1+\alpha)/2$. Let $\bar{t}_1 > 0$ be such that, for all $t \in [0, \bar{t}_1]$, $C(x^* + td) \subseteq C(x^*)$. Since $\nabla f_i(x^*)^T d < 0$, there exists $\bar{t}_2 \in (0, \bar{t}_1]$ such that for all $t \in [0, \bar{t}_2]$, $i \in C(x^*)$,

$$f_i(x^* + td) \leq f_i(x^*) + \alpha' t \nabla f_i(x^*)^T d \leq f_i(x^*) - \alpha' t\varepsilon = f(x^*) - \alpha' t\varepsilon.$$

Since this inequality holds for all $i \in C(x^*)$ and $C(x^* + td) \subseteq C(x^*)$, it turns out that

$$f(x^* + td) \leq f(x^*) - \alpha' t\varepsilon \tag{14}$$

for all $t \in [0, \bar{t}_2]$. Now, suppose that for infinitely many indices $k \in K_2 \subseteq K$ one has that there exists $t'_k \in [\sigma_{min}\bar{t}_2, \bar{t}_2]$ such that

$$f(x^k + t'_k d^k) > f(x^k) - \alpha t'_k \varepsilon.$$

Taking limits we obtain that (14) does not hold. Therefore, for $k \in K_2$ large enough, steplengths t in the interval $[\sigma_{min}\bar{t}_2, \bar{t}_2]$ cannot be rejected and, so, necessarily satisfy (11). By (12), this means that for all $k \in K$ large enough, one has that $t_k \geq \sigma_{min}\bar{t}_2$. This is impossible because it would imply that $f(x^k) \to -\infty$. The contradiction came from assuming, at the beginning of the proof, that the sequence generated by Algorithm 4.2 could be infinity. □

Many heuristics can be developed in order to alleviate the computational cost of the main iteration of Algorithm 4.1. The direction d^k is usually computed solving the optimization problem

$$\text{Minimize } z$$

subject to

$$z \geq \nabla f_i(x^k) d \quad \text{for all} \quad i \in C_\delta(x^k) \tag{15}$$
$$x^k + d \in \Omega, \|d\| \leq \Delta.$$

The number of constraints (15) could be large, which makes it desirable to obtain a suitable descent direction employing only a part of them. The solution of the subproblem with a smaller number of constraints provides a good starting point for solving the full-constraints subproblem. A suitable management of the steplength makes it possible to preserve the convergence properties of the original algorithm. Some of these ideas are systematized in the following algorithm.

Algorithm 4.2

Assume that $\varepsilon > 0$, $\delta > 0$, $\Delta > 0$, $\alpha \in (0,1)$, $0 < \sigma_{min} < \sigma_{max} < 1$, $x^0 \in \Omega$. Initialize $k \leftarrow 0$.

Step 1. Set $J \leftarrow \emptyset$.

Step 1.1. Take $j \in C_\delta(x^k) - J$ and update $J \leftarrow J \cup \{j\}$. Compute d^+, a solution of

$$\text{Minimize } \max\{\nabla f_i(x^k)^T d,\ i \in J\} \quad \text{subject to} \quad x^k + d \in \Omega,\ \|d\| \leq \Delta. \tag{16}$$

Step 1.2. If $\max\{\nabla f_i(x^k)^T d^+,\ i \in J\} \geq -\varepsilon$, stop.

Step 2. Set $t \leftarrow 1$, $\nu \leftarrow 0$.

Step 2.1. If

$$f(x^k + td^+) \leq f(x^k) - t\alpha\varepsilon, \tag{17}$$

set $t_k = t$, $d^k = d^+$, compute $x^{k+1} \in \Omega$ such that

$$f(x^{k+1}) \leq f(x^k + t_k d^k), \tag{18}$$

set $k \leftarrow k + 1$ and go to Step 1.

Step 2.2. Set $\nu \leftarrow \nu + 1$. If $\nu \leq \#J$ or $J = C_\delta(x^k)$, choose

$$t_{new} \in [\sigma_{min} t, \sigma_{max} t],$$

set $t \leftarrow t_{new}$ and go to Step 2.1. Else, go to Step 1.1.

Remarks. The difference between Algorithm 4.2 and Algorithm 4.1 is that in Algorithm 4.2 we try to obtain the search direction using subproblems with a smaller number of constraints. (We assume that (16) is solved by means of the minimization of the auxiliary variable z subject to $z \geq \nabla f_i(x^k)^T d, i \in J,\ x^k + d \in \Omega, \|d\| \leq \Delta$.) If the descent condition (17) does not hold, and $J \neq C_\delta(x^k)$ we admit as many backtracking reductions as the number of elements of J. If $J = C_\delta(x^k)$ we admit as many reductions of t as necessary. Only the steps in which $J = C_\delta(x^k)$ force the termination of the algorithm which, in turn, follows trivially as in Theorem 4.2. However, subproblems in which $J \neq C_\delta(x^k)$ may be very successful in nonconvex Piecewise-Smooth problems. Consider, for example, the problem of minimizing the minimum between x and $-x$ subject to $x \in [-1, 1]$. Start with $x^0 = 0$, which is a stationary point. If one uses Algorithm 4.1, the method will stop at x^0. However, if one employs Algorithm 4.2, with $\Delta > 1$ the method will find a global minimizer $x^1 \in \{-1, 1\}$ in just one iteration.

Clearly, in (18) we may choose $x^{k+1} = x^k + t_k d^k$. However, different choices of the new current point such that $f(x^{k+1}) < f(x^k + t_k d^k)$ may provide opportunistic accelerations of the basic procedure. For example, starting from $x^k + t_k d^k$ we may try a Newton step based on just one of the functions $f_i \in C(x^k + t_k d^k)$. We will see that this device has interesting theoretical properties in the case of Low Order-Value Optimization problems. It is pertinent to mention here that nonconvex nonsmooth optimization using the BFGS method has recently be considered in [32, 43].

Algorithm 4.3. Assume that $\Delta > 0$ Let $\{\varepsilon_k\}, \{\delta_k\}$ be sequences of positive numbers such that $\lim_{k\to\infty} \varepsilon_k = \lim_{k\to\infty} \delta_k = 0$.

Step 1. Compute $x^k \in \Omega$ in such a way that

$$\max\{\nabla f_i(x^k)^T d \mid i \in C_{\delta_k}(x^k)\} \geq -\varepsilon_k$$

for all $d \in I\!R^n$ such that $x^k + d \in \Omega$ and $\|d\| \leq \Delta$.

Step 2. If

$$\max\{\nabla f_i(x^k)^T d \mid i \in C(x^k)\} \geq 0$$

for all $d \in I\!R^n$ such that $x^k + d \in \Omega$, stop. Else, set $k \leftarrow k + 1$ and go to Step 1.

Theorem 4.3 *Algorithm 4.3 is well defined. Moreover, if the algorithm does not stop at Step 2, every limit point x^* of $\{x^k\}$ satisfies*

$$\max\{\nabla f_i(x^*)^T d \mid i \in C(x^*)\} \geq 0$$

for all feasible direction d.

Proof. According to Theorem 4.2, the iterate x^k at Step 1 can be computed using Algorithms 4.1 or 4.2 in finite time. Now, assume that Algorithm 4.3 generates an infinite sequence $\{x^k\}$ and the subsequence $\{x^k\}_{k\in K}$ converges to $x^* \in \Omega$. Assume, by contradiction, that there exists $d \in I\!R^n$ such that $x^* + d \in \Omega$, $\|d\| \leq \Delta$, and

$$\nabla f_i(x^*)^T d < -\varepsilon < 0$$

for all $i \in C(x^*)$.

Define $\delta = \min\{|f_i(x) - f(x^*)| \mid i \notin C(x^*)\}/2$. If $j \notin C(x^*)$ we have that $|f_j(x^*) - f(x^*)| \geq 2\delta$ and, so, $j \notin C_\delta(x^k)$ for k large enough. Therefore, $C_\delta(x^k) \subseteq C(x^*)$ for k large enough.

Define, for all $k \in K$, $\hat{d}^k = d + x^* - x^k$. Clearly, $x^k + \hat{d}^k = x^* + d \in \Omega$. If $\|\hat{d}^k\| \leq \Delta$, define $d^k = \hat{d}^k$. Otherwise, define $d^k = \hat{d}^k \Delta / \|\hat{d}^k\|$. Since $\|d\| \leq \Delta$, we have that $\lim_{k\in K} d^k = d$. Let k_0 be such that, for all $k \in K$ such that $k \geq k_0$ and $i \in I$,

- $C_{\delta_k}(x^k) \subseteq C(x^*)$;
- $\varepsilon_k < \varepsilon/2$;
- $|\nabla f_i(x^k)^T d^k - \nabla f_i(x^*)^T d| < \varepsilon/2$.

Then, for $k \in K$ such that $k \geq k_0$ and $i \in C_{\delta_k}(x^k)$, one has that $i \in C(x^*)$ and:

$$\nabla f_i(x^k)^T d^k = [\nabla f_i(x^k)^T d^k - \nabla f_i(x^*)^T d] + \nabla f_i(x^*)^T d < \varepsilon/2 + \nabla f_i(x^*)^T d.$$

Therefore, since $i \in C(x^*)$,

$$\nabla f_i(x^k)^T d^k < \varepsilon/2 - \varepsilon = -\varepsilon/2 < -\varepsilon_k.$$

This contradicts the definition of x^k. □

Remarks

The algorithms presented in this section are based on the linear first order models of the functions f_i, namely, $f_i(x^k + d) \approx f_i(x^k) + \nabla f_i(x^k)^T d$. If Ω is represented by linear constraints and $\|\cdot\| = \|\cdot\|_1$ or $\|\cdot\| = \|\cdot\|_\infty$, d^k is given by the solution of a Linear Programming problem. If we adopt convex quadratic models $f_i(x + d) \approx f_i(x) + \nabla f_i(x)^T d + \frac{1}{2} d^T B_k d^k$, with bounded Hessians B_k, we obtain similar algorithms and results but the subproblems that define d^k are more complicated. The corresponding theory has been developed in [7] for the case in which $f(x)$ is given by (2).

The requirements on the directions d^k can be relaxed, imposing only that they should be approximate solutions of the subproblems. See [2] for the case (2) with linear models and [7] for convex quadratic models.

5 Smooth reformulations

Some Generalized Order-Value Optimization problems admit reformulations in terms of standard smooth optimization problems. In this section we consider the cases in which one wishes to minimize $f_o(x)$, given by (2), and $f_{o+}(x)$, given by (3).

Assume, firstly, that we want to minimize $f_{o+}(x)$ subject to $x \in \Omega$.

Clearly, one has that $f_{o+}(x)$ is the maximum of $\frac{1}{m-p} \sum_{j \in J} f_j(x)$, taken over all the subsets $J \subset \{1, \ldots, m\}$ such that $\#J = m - p$. Therefore, minimizing $f_{o+}(x)$ subject to $x \in \Omega$ is equivalent to the following smooth optimization problem:

$$\text{Minimize } z \tag{19}$$

$$\text{subject to } x \in \Omega \text{ and } z \geq \sum_{j \in J} f_j(x) \text{ for all } J \subseteq \{1, \ldots, m\} \mid \#J = m - p. \tag{20}$$

This is a smooth reformulation of the original problem. If the functions f_j are linear and Ω defines a polytope, then (19-20) is a Linear Programming problem. If Ω is convex and the functions f_j are convex we are in presence of a convex programming problem. The linear case is common when $-f_j(x)$ represents the predicted return of an investment under the scenario j. In this case the problem consists of finding the portfolio that minimizes CVaR.

The number of constraints (20) may be very large ($\frac{m!}{p!(m-p)!}$). In what follows we deduce a reformulation with a smaller number of constraints, which, essentially, corresponds to the approach described in [41].

We begin with a simple lemma on finite sequences of real numbers. Let $a_1, \ldots, a_m \in I\!R$ be such that

$$a_1 \leq \ldots \leq a_m$$

and $p \in \{1, \ldots, m\}$. We define, for all $\xi \in I\!R$,

$$F(\xi) = (m-p)\xi + \sum_{a_i \geq \xi} (a_i - \xi).$$

Lemma 5.1 *If $p < m$, the set of minimizers of $F(\xi)$ is the interval $[a_p, a_{p+1}]$ with minimum $\sum_{p+1}^{m} a_i$. If $p = m$ the set of minimizers is $[a_m, \infty)$ and the minimum is zero.*

Proof. Note that the slope is zero in the interval of minimizers, greater than zero on the right and smaller than zero on the left. □

Now, assume that $f_i : \Omega \to I\!R$ and $i_1, \ldots, i_p$ are defined as in (1). We consider the problem of minimizing $f_{o+}(x)$ subject to $x \in \Omega$:

$$\text{Minimize } \frac{1}{m-p}[f_{i_{p+1}(x)}(x) + \ldots + f_{i_m(x)}(x)] \text{ subject to } \ x \in \Omega. \tag{21}$$

By Lemma 5.1, for solving this problem it is enough to minimize, with respect to $x \in \Omega$, the minimum with respect to ξ of $[(m-p)\xi + \sum_{f_i(x) \geq \xi}(f_i(x) - \xi)]$.

This corresponds to:

$$\underset{x \in \Omega}{\text{Minimize}} \ \underset{\xi}{\text{Minimum}} \ (m-p)\xi + \sum_{i=1}^{m} \max\{0, f_i(x) - \xi\}.$$

Equivalently,

$$\underset{x \in \Omega, \xi \in I\!R}{\text{Minimize}} \ (m-p)\xi + \sum_{i=1}^{m} \max\{0, f_i(x) - \xi\}.$$

This problem can be written as:

$$\underset{x, z, \xi}{\text{Minimize}} \ (m-p)\xi + \sum_{i=1}^{m} z_i$$

subject to $x \in \Omega, \ z_i \geq 0, \ z_i \geq f_i(x) - \xi, i = 1, \ldots, m.$

If the functions $f_i(x)$ are linear and Ω is a polytope, this is a Linear Programming problem, as (19,20). However, the number of constraints is substantially smaller than in the former problem.

Let us now consider the problem of minimizing VaR:

$$\text{Minimize } f_{i_p(x)}(x) \ \text{ subject to } x \in \Omega. \tag{22}$$

By Lemma 5.1, we know that $f_{i_p(x)}(x)$ is the smallest value of ξ that minimizes $(m-p)\xi + \sum_{i=1}^{m} \max\{0, f_i(x) - \xi\}$. Therefore, the problem (22) is:

$$\text{Minimize } \xi$$

subject to $x \in \Omega$ and to the condition that ξ is a minimizer of $(m-p)\xi + \sum_{i=1}^{m} \max\{0, f_i(x) - \xi\}$.

In other words, (22) may be written as follows:

$$\text{Minimize } \xi$$

subject to $x \in \Omega$ and to the condition that ξ solves the optimization problem given by:

$$\underset{\xi,z}{\text{Minimize}} \ (m-p)\xi + \sum_{i=1}^{m} z_i$$

subject to $z_i \geq 0, z_i \geq f_i(x) - \xi, i = 1, \ldots, m$. This formulation was introduced, with different arguments, in [3] and [40].

6 Low Order-Value Optimization

Given $p \in I \equiv \{1, \ldots, m\}$ and the set $\{f_i\}_{i \in I}$, the Low Order-value Optimization (LOVO) problem consists of minimizing $f_{low} : \Omega \to I\!R$, where

$$f_{low}(x) = \sum_{j=1}^{p} f_{i_j(x)}(x),$$

and $i_1, \ldots, i_m$ are defined by (1). If the functions f_i are smooth, the function f_{low} is piecewise-smooth. When $f_i(x)$ is the error of a model with respect to the observation i under the parameters x, minimizing f corresponds to fitting the model with automatic discard of $m-p$ possible outliers. For each subset $J \subseteq I$ such that $\#J = p$, we define

$$f_J(x) = \sum_{j \in J} f_j(x).$$

It is easy to see that minimizing f_{low} is equivalent to minimize (with respect to x) the minimum (with respect to J) of $f_J(x)$. Then, renaming the functions f_i, we may consider that, at least from the formal point of view, the Low Order-Value Optimization problem is of the form

$$\text{Minimize } f_{min}(x) \equiv \min\{f_i(x), i \in I\} \tag{23}$$

subject to $x \in \Omega$. The algorithms presented in Section 4 may be used for solving (23). These algorithms are guaranteed to converge to stationary points in the sense of (9), as many other nonsmooth nonconvex methods do. However, in the case of (23), a stronger optimality condition can be proved [5].

Theorem 6.1 *Let x^* be a local minimizer of (23). Then,*

$$\nabla f_i(x^*)^T d \geq 0 \ \ \textit{for all} \ \ d \ \ \textit{feasible}, i \in C(x^*). \tag{24}$$

A weaker optimality condition than (24) but yet stronger than Clarke stationarity [14] is:

$$\textit{There exists } \; i \in C(x^*) \mid \nabla f_i(x^*)^T d \geq 0 \; \textit{ for all } \; d \; \textit{ feasible.} \tag{25}$$

Conditions (24) and (25) are necessary for optimality because, in this problem, the fact that $f_i(y) < f(x)$ for some $i \in C(x)$ implies that $f(y) < f(x)$. Then, if d is a descent direction for $f_i \in C(x)$, it is a descent direction for f too.

In Section 4.1 we saw that Algorithm 4.2 should be more effective than Algorithm 4.1 for solving (23). Line-search algorithms and trust-region algorithms for solving (23) that converge to points that satisfy (24) or (25) were introduced in [4] and [6]. Algorithms for solving (23) that converge to points that satisfy (25) are, essentially, algorithms for smooth problems in which, whenever $\nabla f(x^k)$ is required, one computes $\nabla f_i(x^k)$ for some $i \in C(x^k)$. Algorithms that converge to stationary points in the sense of (24) are more complicate and do not seem to be more effective, in practice, than the ones that seek the weaker condition (25).

6.1 LOVO problems and Coordinate-Search

The Coordinate-Search (CS) method is a useful tool for Derivative-Free Optimization [15]. It may be used as an independent algorithm or as an auxiliary procedure in the context of more sophisticated methods. In this section we assume that Ω is an $n-$dimensional box ($\Omega = \{x \in I\!R^n \mid \ell \leq x \leq u\}$ with $\ell < u$). For simplicity, we define, in this section, $f = f_{min}$, with $f(x) = \infty$ and $f_j(x) = \infty$ if $j \in I$ and $x \notin \Omega$. The CS method generates a sequence of points $\{x^k\} \subset \Omega$ such that, for all $k = 1, 2, \ldots$, $i = 1, \ldots, n$,

$$f(x^k - \delta_k e^i) \geq f(x^k) \leq f(x^k + \delta_k e^i), \tag{26}$$

where $\lim_{k\to\infty} \delta_k = 0$ and $\{e^1, \ldots, e^n\}$ is the canonical basis.

In general Piecewise-Smooth Optimization, the CS method may be quite ineffective because convergence to a kink that does not satisfy first-order optimality conditions usually occurs. However, in LOVO problems the situation is quite different. We will see that CS-like methods are able to find points that satisfy rather strong optimality conditions in this case.

Theorem 6.2 *Assume that the sequence $\{x^k\}$ satisfies (26) and K is an infinite sequence of indices such that $\lim_{k\in K} x^k = x^* \in \Omega$. Suppose that, for infinitely many indices $k \in K_1 \subseteq K$ we have that $f_j \in C(x^k)$. Then,*

$$\textit{If } \ell_i < x_i^* < u_i, \; \frac{\partial f_j}{\partial x_i}(x^*) = 0; \tag{27}$$

$$\textit{If } x_i^* = \ell_i,\ \frac{\partial f_j}{\partial x_i}(x^*) \geq 0; \tag{28}$$

and

$$\textit{If } \ x_i^* = u_i,\ \frac{\partial f_j}{\partial x_i}(x^*) \leq 0. \tag{29}$$

Proof. By (26), since $f_j(x^k) = f(x^k)$, $f_j(x^k+\delta_k e^i) \geq f(x^k+\delta_k e^i)$ and $f_j(x^k-\delta_k e^i) \geq f(x^k-\delta_k e^i)$, we have that, for all $k \in K_1$,

$$f_j(x^k - \delta_k e^i) \geq f_j(x^k) \leq f_j(x^k + \delta_k e^i). \tag{30}$$

The thesis follows from (30) applying the Mean Value Theorem and taking limits. □

Theorem 6.2 does not ensure that (27, 28, 29) hold for all $f_j \in C(x^*)$. Consider, for example, the following situation: We define $f_1 : I\!R^2 \to I\!R$, $f_2 : I\!R^2 \to I\!R$ by:

$$f_1(x_1, x_2) = \frac{x_1^2 + x_2^2}{3} \ \text{ if } \ x_1 \geq 0 \ \text{ and } \ 2x_1 \geq x_2 \geq x_1/2;$$

$$f_1(x_1, x_2) = x_1^2 + x_2^2 \ \text{ if } x_1 \leq 0 \ \text{ or } \ x_2 \leq 0.$$

In the region of the plane not contemplated by the definitions above, we define f_1 in such a way that first-order smoothness is preserved. We also define:

$$f_2(x_1, x_2) = x_1 + x_2.$$

Take the sequence $x^k = (1/k, 1/k)$ and $\delta_k = 1/k$. Clearly, $f_{min}(x^k) = f_1(x^k)$ for all k and the sequence satisfies (26). However, at the limit $x^* = (0, 0)$ one has that $f_2 \in C(x^*)$ and the gradient of this function does not vanish.

To ensure that (27, 28, 29) take place for all $j \in C(x^*)$ we need a stronger form of the CS method. Assume that $\delta > 0$ and the positive sequence $\{\delta_k\}$ tends to zero. Each iterate x^k generated by the SCS (Strong Coordinate Search) Method will be such that, for all $i = 1, \ldots, n$,

$$f(x^k + \nu\delta_k e^i) \geq f(x^k) \text{ for all } \nu = 1, -1, 2, -2, \ldots \ \text{ such that } \ |\nu\delta_k| \leq \delta. \tag{31}$$

In the following theorem, we prove that, at every limit point x^* of a sequence generated by the SCS method, the projected gradient of f_j is zero, for all $j \in C(x^*)$.

Theorem 6.3 *Assume that the sequence $\{x^k\}$ satisfies (31) and K is an infinite sequence of indices such that $\lim_{k\in K} x^k = x^* \in \Omega$. Then, for all $j \in C(x^*)$, (27), (28) and (29) hold.*

Proof. Assume that $x_i^* < u_i$ and

$$\frac{\partial f_j}{\partial x_i}(x^*) < 0 \tag{32}$$

for some $j \in C(x^*)$. Therefore, there exists $t_0 \in (0, \delta]$ such that $x_i^* + t_0 < u_i$ and

$$f_j(x^* + te^i) < f_j(x^*) \tag{33}$$

for all $t \in (0, t_0]$. Thus,

$$f(x^* + te^i) \leq f_j(x^* + te^i) < f_j(x^*) = f(x^*) \text{ for all } t \in (0, t_0]. \tag{34}$$

Let k_0 be such that for all $k \in K_1$ and $k \geq k_0$, one has that $\delta_k < t_0$. Therefore, by (31),

$$f(x^k + \nu\delta^k e^i) \geq f(x^k) \tag{35}$$

for all $\nu = 1, 2, \ldots$ such that $\nu\delta_k < t_0$. Define, for all $k \geq k_0$,

$$\delta'_k = \max\{\nu\delta_k \mid \nu\delta_k < t_0\}.$$

Clearly, $\delta'_k \in [t_0/2, t_0]$ for k large enough. Taking limits for an appropriate subsequence we obtain that $\delta'_k \to \delta' \in [t_0/2, t_0]$. So, taking limits in (35), we get that

$$f(x^* + \delta' e^i) \geq f(x^*).$$

This contradicts (34). Therefore, (32) is false. Similarly, we prove that, if $j \in C(x^*)$ and $x_i^* > \ell_i$ we have that $\frac{\partial f_j}{\partial x_i}(x^*) \leq 0$. This completes the proof. □

Assume that one applies an algorithm that converges to weak stationary points to the minimization of f_{min} with box constraints. The algorithms presented in [6] may be used for that purpose. Usually, one stops the iterative procedure when an iterate x^k is found such that the norm of the projected gradient of one of the functions $f_j \in C(x^k)$ is close to zero. When this happens, it is recommendable to switch to the SCS method which, without explicit enumeration of $C(x^k)$, will check whether the projected gradients of *all* the functions $f_j \in C(x^k)$ are close to zero. If this is not the case, we continue with the SCS method so that, in the limit, we will guarantee that limit points verify the strong optimality condition (24).

6.2 LOVO problems with general smooth constraints

Here we consider the problem

$$\text{Minimize } f_{low}(x) \text{ subject to } h(x) = 0, g(x) \leq 0, x \in \Omega, \tag{36}$$

where $h : I\!R^n \to I\!R^m$ and $g : I\!R^n \to I\!R^p$ have continuous first derivatives. A method for solving this problem using the Augmented Lagrangian approach of [1] was introduced in [6]. Assume, to

simplify the exposition, that Ω is a box. Given $x \in \Omega$, a penalty parameter $\rho > 0$ and Lagrange multipliers estimates $\lambda \in I\!R^m, \mu \in I\!R^p_+$ the Augmented Lagrangian is defined by

$$L_\rho(x, \lambda, \mu) = f_{low}(x) + \frac{\rho}{2}\Big[\sum_{i=1}^{m}\Big(h_i(x) + \frac{\lambda_i}{\rho}\Big)^2 + \sum_{i=1}^{p}\max\Big\{0, g_i(x) + \frac{\mu_i}{\rho}\Big\}^2\Big]. \tag{37}$$

At each iterate of the algorithm one finds an approximate minimizer x^k of $L_\rho(x, \lambda, \mu)$ subject to $x \in \Omega$. This subproblem is a LOVO problem in the sense of (23) and, as a consequence, may be solved by algorithms discussed in previous sections. Under appropriate assumptions, it may be proved that feasible limit points x^* of the sequence $\{x^k\}$ whose active constraints satisfy a weak constraint qualification [8] fulfill the KKT conditions of the problem

$$\text{Mimimize } f_j(x) \text{ subject to } h(x) = 0, g(x) \leq 0, x \in \Omega$$

for some $f_j \in C(x^*)$ [6]. If x^k is an approximate global minimizer of (37), one may invoke the results of [11] to prove that x^* is a global minimizer of (36).

6.3 LOVO and MINLP

Since $f_j(x)$ depends on the continuous variable x and the integer variable j, the LOVO problem (23) may be studied in the context of Mixed Integer Nonlinear Programming (MINLP). See [17, 37]. Reciprocally, assume that one has the MINLP problem given by:

$$\text{Minimize } f(x, z) \text{ subject to } g_i(x, z) \leq 0,\ i = 1, \ldots, p, (x, z) \in \Omega \times \Omega_{int}, x \in I\!R^n, z \in Z^q, \tag{38}$$

where Z denotes the set of integers. Assuming that Ω and Ω_{int} are bounded boxes, we may write:

$$\Omega_{int} \cap Z = \{z^i, \ldots z^m\}.$$

We define, for all $i = 1, \ldots, p$, $j = 1, \ldots, m$, $x \in \Omega$,

$$f_j(x) = f(x, z^j) \text{ if } g_i(x, z^j) \leq 0 \ \forall\, i = 1, \ldots, p,$$

$$f_j(x) = \infty \text{ otherwise.}$$

With these definitions, (38) takes the LOVO form (23). The definition of f_j suggests many penalty, smoothing and Augmented Lagrangian approximations.

7 VaR Constraints and Transaction Costs

Assume that, with the definitions (1) and (2), one wants to minimize a smooth function $f(x)$ subject to $f_o(x) \equiv f_{i_p(x)}(x) \leq c$. This problem appears in the financial environment when one wishes to maximize the average return of an investment under a Value-at-Risk constraint.

Assuming that $-f_1(x), -f_2(x), \ldots, -f_m(x)$ represent the returns generated by the decision $x \in \mathbb{R}^n$ under scenarios $1, 2, \ldots, m$ (generated by simulation), the Value-at-Risk with confidence level α is well approximated by $f_o(x)$, if $\alpha \approx p/m$. In that case the objective function is $f(x) \equiv \sum_{i=1}^{m} f_i(x)/m$. By (1), the constraint $f_o(x) \leq c$ is equivalent to the set of constraints

$$f_{i_1(x)}(x) \leq c, \ldots, f_{i_p(x)}(x) \leq c. \tag{39}$$

The formulation (39) facilitates the resolution of the optimization problem using the Augmented Lagrangian approach [10].

Given $\rho > 0$, $\mu \in \mathbb{R}_+$, $x \in \mathbb{R}^n$, we define the LOVO Augmented Lagrangian function $L_\rho(x, \mu)$ by:

$$L_\rho(x, \mu) = f(x) + \frac{\rho}{2}\Big[\sum_{j=1}^{p}\Big(f_{i_j(x)}(x) - c + \frac{\mu}{\rho}\Big)_+^2\Big]. \tag{40}$$

Define, for all $i = 1, \ldots, m$,

$$\bar{f}_i(x) = \frac{1}{p}f(x) + \frac{\rho}{2}\Big[\Big(f_i(x) - c + \frac{\mu}{\rho}\Big)_+^2\Big].$$

Observe that $L_\rho(x, \mu)$ is the sum of the p lowest elements of $\{\bar{f}_1(x), \ldots, \bar{f}_m(x)\}$. This means that the minimization of $L_\rho(x, \mu_k)$ represents a LOVO problem, for which suitable algorithms have been discussed in previous sections. These subproblems may include lower-level subproblem constraints in the sense of [1]. The main algorithm introduced in [10] considers the problem

$$\text{Minimize } f(x) \text{ subject to } f_o(x) \leq c, x \in \Omega, \tag{41}$$

where Ω represents a set of smooth constraints. At each iteration k, given a penalty parameter ρ_k and a Lagrangian shift μ_k/ρ_k, we compute an approximate minimizer of $L_{\rho_k}(x, \mu_k)$ subject to $x \in \Omega$. If x^* is a feasible limit point of $\{x^k\}$ then, under suitable qualifications [8], this point satisfies the KKT conditions of

$$\text{Minimize } f(x) \text{ subject to } f_j(x) \leq c \; \forall j \in J, x \in \Omega,$$

where J is a set of p different indices such that $f_j(x^*) \leq f_i(x^*)$ for all $j \in J, i \notin J$.

Finally, if we suppose that each x^k is an approximate global minimizer of $L_{\rho_k}(x, \mu_k)$ subject to $x \in \Omega$, the continuity of $f_{i_j(x)}(x)$ for all j and the results of [11] guarantee that, in the limit, one finds a global minimizer of (41).

In [10] this approach was used to maximize the expected return of a portfolio under VaR constraints. Transaction costs were incorporated within the loss functions which, therefore, turned out to be highly nonlinear. Nevertheless, the LOVO algorithm sketched here found suitable solutions in negligible computer time.

8 Protein Alignment

Let $P = (P_1, \ldots, P_{NP})$ and $Q = (Q_1, \ldots, Q_{NQ})$ be two finite sequences. An Alignment between P and Q is a sequence of pairs $(A_1, B_1), \ldots, (A_N, B_N)$ such that $(A_1, \ldots, A_N)$ is a subsequence of P and $(B_1, \ldots, B_N)$ is a subsequence of Q. The best alignment between two finite sequences is supposed to maximize some score. Here we are concerned with the case in which P and Q are proteins. More precisely, $P_1, \ldots, P_{NP}$ are the 3D coordinates of the $C\alpha$ atoms of a protein P and $Q_1, \ldots, Q_{NQ}$ are the 3D coordinates of the $C\alpha$ atoms of a protein Q. The alignment may be thought as being a monotone bijection between subsequences of P and Q. If, for some i, the atoms A_i, A_{i+1} are not consecutive elements in the sequence P or the atoms B_i, B_{i+1} are not consecutive elements in the sequence Q we say that the alignment presents a gap between (A_i, B_i) and (A_{i+1}, B_{i+1}). The "Structal Score" [45] associated with the alignment $((A_1, B_1), \ldots, (A_N, B_N))$ is defined by

$$Structal = \sum_{i=1}^{N} \frac{20}{1 + \|A_i - B_i\|_2^2/5} - 10N_{gaps},$$

where N_{gaps} is the number of gaps. In the best case one of the sequences $\{A_i\}$ or $\{B_i\}$ is equal to the smaller protein, one has that $B_i = A_i$ for all i and there are no gaps. In this case, the Structal Score is $20N = 20\min\{NP, NQ\}$. Observe that a perfect match between two atoms $(A_i = B_i)$ contributes with 20 units to the score and each gap decreases the score by 10 units. The score associated with a particular alignment may be negative, but the one associated with the best alignment between P and Q is necessarily positive, since any alignment with $N = 1$ have no gaps and its score is $\frac{20}{1+\|A_1-B_1\|_2^2/5} > 0$. Given the coordinates of P and Q, the best alignment can be computed using Dynamic Programming [38]. See [30, 36] and references therein.

Obviously, rigid displacements can be performed on the whole structure of a protein without changing any of its biological properties. Therefore, one is interested in finding, not only the best alignment for fixed spatial positions, but the best one with respect to all possible rigid displacements. Clearly, we only need to consider the displacement of one of the proteins with this purpose, therefore the variable displacement is represented by three translation variables and three angles of rotation. Thus, the rigid displacement is represented by a vector $x \in \mathbb{R}^6$. We write the protein Q as $(Q_1(x), \ldots, Q_{NQ}(x))$. Therefore, our optimization problems consists of maximizing, with respect to x, the Structal score of the best possible alignment between $(P_1, \ldots, P_{NP})$ and $(Q_1(x), \ldots, Q_{NQ}(x))$.

Let I be the set of all monotone bijections between subsequences of P and Q. Define $f_i(x)$ as the Structal score corresponding to the alignment i given the displacement x. Then, our problem consists of maximizing, with respect to x, the maximum, with respect to i, of $f_i(x)$. Clearly, this is a LOVO problem of type (23). The package LovoAlign [36] addresses this problem using a variant of Newton's method. As a consequence, in the limit we obtain the optimality condition

(25).

A slightly different approach is adopted in the PhD Thesis of P. Gouveia [21]. Following [31], Gouveia considers only a finite number of feasible translations. Given a rotation of the structure Q, represented by three angles $x \in I\!R^3$, he considers the translations defined by superimposing $Q_i(x)$ and P_j, for $i = 1, \ldots, NP, j = 1, \ldots, NQ$. Therefore, one has $NP \times NQ$ feasible translations. The rotation x associated with each translation represents a feasible rigid displacement. For each rigid displacement we may compute the best sequence alignment and our problem consists, as before, of finding the rigid displacement that provides the alignment with maximal score. With the proper definitions of the functions f_i (which involve now not only sequence alignments but also translations) we have again a LOVO problem of type (23) that can be solved up to stationarity (25) using some variation of the smooth Newton method [6]. The Alignment problem was formulated in this way in the PNAS paper [31], where a grid strategy for optimization was used in the space of rotations. The resulting method was proved to solve the problem up to any required precision in polynomial time (as a function of the number of $C\alpha$ atoms) but is inviable in practical computations.

Gouveia [21] analyzed the application of global optimization methods for solving the Protein Alignment problem. His general conclusion is that the employment of LovoAlign with a clever multistart strategy generally outperforms the behavior of well established global optimization packages, both in terms of computer time and quality of solutions.

8.1 Multiple Alignments

In Multiple Alignment problems one aims to identify homologous subsequences in a set of different proteins. Suppose that $P^1, \ldots, P^M$ are M finite sequences. A Multiple Alignment of $P^1, \ldots, P^M$ is a sequence of M-uples $((A_1^1, \ldots, A_1^M), \ldots, (A_N^1, \ldots, A_N^M))$ such that, for all $i = 1, \ldots, M$, $(A_1^i, \ldots, A_N^i)$ is a subsequence of P^i.

Assume that, for all $j = 1, \ldots, M$, $P^j(x^j)$ is the sequence of points that represent the $C\alpha$ atoms of Protein j, as dependent of the rigid displacement $x^j \in I\!R^6$. Given M rigid displacements $x^1, \ldots x^M$, we may consider the set I of all possible multiple alignments of $P^1(x^1), \ldots, P^M(x^M)$. To each multiple alignment $i \in I$ a score is assigned that reflects the spatial proximity of different alignments as well as the length N of the alignment and the presence of gaps. For example:

$$Score = \sum_{k=1}^{N} \varphi(A_k^1(x^1), \ldots, A_k^M(x^M)) - \psi(N_{gaps}),$$

where $\varphi(C^1, \ldots, C^M)$ is maximal if $C^1 = \ldots = C^M$ and ψ increases with the number of gaps. Unlike the single alignment case ($M = 2$), finding the alignment that maximizes the score for fixed positions of the proteins is not an easy task. However, from the conceptual point of view, the situation is not different as the one analyzed previously. We need to maximize,

with respect to the displacements $x^1, \ldots, x^M$, the maximal score with respect to the multiple sequence alignments of $P^1(x^1), \ldots, P^M(x^M)$. This is a typical difficult LOVO problem whose dimension can be very large.

8.2 Protein Maps and Quick Alignments

Assume that the pairwise (say, Structal) scores corresponding to the alignments of a set of proteins $P_1, \ldots, P_M$ have been computed. We wish to represent these proteins in the space, in such a way that close spatial representations correspond to high scores, whereas the spatial objects that represent proteins with low similarity (low scores) are distant [25]. Let us call Q_j to the spatial representation of P_j for all $j = 1, \ldots, M$. In his PhD Thesis, R. Lima [33] suggested that each protein can be represented by different types of spatial objects. One of his proposals is that, for all j, Q_j should be a sequence with as many points as the number of $C\alpha$ atoms of the protein P_j. Moreover, he defined a "gap-free" score between the objects Q_i and Q_j which is essentially the best Structal score that can be found considering all the alignments between Q_i and Q_j without gaps. The computation of the best gap-free alignment is simple and can be performed in linear time. (Recall that the cost of Dynamic Programming is quadratic in the number of atoms [38].) Suppose that Q_i has Ni points, Q_j has Nj points and $|Ni - Nj| = Nij$. (Thus, the number of $C\alpha$ atoms of P_i is Ni and the number of $C\alpha$ atoms of P_j is Nj.) Then, for computing the score between Q_i and Q_j we only need to analyze $Nij + 1$ gap-free sequences. Moreover, the best subsequence that realizes the gap-free score will be, not the one for which the score is maximized but the one for which the difference between the gap-free score and the true Structal score between P_i and P_j is minimal. With some abuse of notation let us call $S(P_i, P_j)$ the known Structal score between proteins P_i and P_j and $S(Q_i, Q_j)$ the gap-free score described before. Then, our objective is to find objects $Q_1, \ldots, Q_N$ such that $\sum_{i,j}[S(Q_i, Q_j) - S(P_i, P_j)]^2$ is minimized. Again, this is a typical LOVO problem.

Assume now that the map $\{Q_1, \ldots, Q_M\}$ is complete, representing the proteins $\{P_1, \ldots, P_M\}$ and that we wish to include a new protein P_{new}, with a quick evaluation of its scores with respect to $P_1, \ldots, P_M$. The procedure is as follows. First we select a small subset $S_{small} \subset \{P_1, \ldots, P_M\}$. For all the proteins $P \in S_{small}$ we compute the Structal scores $S(P, P_{new})$. Using only these Structal scores we find the spatial object Q_{new} minimizing the deviation of Structal scores with respect to gap-free scores $\sum_{P_j \in S_{small}}[S(Q_{new}, Q_j) - S(P_{new}, P_j)]^2$. This is a LOVO problem whose unknown is Q_{new}. Finally, the Structal score $S(P_j, P_{new})$ for $P_j \notin S_{small}$, is estimated as the easy-to-evaluate gap-free score $S(Q_j, Q_{new})$. In this case, since the true Structal scores $S(P_j, P_{new})$ are not available for $j \notin S_{small}$, the gap-free score $S(Q_j, Q_{new})$ is computed as the one that maximizes the score of the gap-free sequences.

9 Conclusions

In this paper we surveyed some of the main results and we introduced several new ideas in the field of Order-Value Optimization and its applications. The denomination Generalized Order-Value Optimization (GOVO) is introduced in the present paper to designate problems in which objective function or constraints depend on some (perhaps partial) order involving functional values of some representation set. Some examples presented here have not yet been addressed in practical computations.

We showed that GOVO problems can be studied under the framework of Piecewise-Smooth Optimization. Algorithms introduced in previous papers for solving particular OVO problems were generalized to Piecewise-Smooth Optimization and convergence proofs for these algorithms were given.

The Low Order-Value Optimization problem was revisited. We showed that LOVO provides a good model for the problem of minimization with Value-at-Risk constraints and we sketched the connection of LOVO with Mixed Integer Nonlinear Programming. We emphasized the role of Coordinate Search procedures for obtaining strong optimality conditions in LOVO problems. Concerning the well established applications to Protein Alignment, we introduced two extensions: A LOVO model for Multiple Protein Alignment and a procedure for designing maps using a LOVO approach.

Many of the ideas presented in this report require corroboration, both in terms of computer efficiency and practical usefulness. As such, they are currently being subject of active research [21, 33].

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, On Augmented Lagrangian methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286-1309 (2007).

[2] R. Andreani, C. Dunder and J. M. Martínez, Order-Value Optimization: formulation and solution by means of a primal Cauchy method, *Mathematical Methods of Operations Research* 58, pp. 387-399 (2003) .

[3] R. Andreani, C. Dunder and J. M. Martínez, Nonlinear-Programming reformulation of the Order-Value Optimization Problem, *Mathematical Methods of Operations Research* 61, pp. 365-384 (2005).

[4] R. Andreani, J. M. Martínez and L. Martínez, Trust-Region superposition methods for Protein Alignment. *IMA Journal on Numerical Analysis* 28, pp. 690-710 (2008).

[5] R. Andreani, J. M. Martínez, L. Martínez and F. S. Yano, Continuous Optimization methods for Structural Alignment, *Mathematical Programming* 112, pp. 93-124 (2008).

[6] R. Andreani, J. M. Martínez, L. Martínez and F. S. Yano, Low Order-Value Optimization and applications, *Journal of Global Optimization* 43, pp. 1-10 (2009).

[7] R. Andreani, J. M. Martínez, M. Salvatierra and F. S. Yano, Quasi-Newton methods for Order-Value Optimization and Value-at-Risk calculations, *Pacific Journal of Optimization* 2, pp. 11-33 (2006).

[8] R. Andreani, J. M. Martínez and M. L. Schuverdt, On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications* 125, pp. 473–485 (2005).

[9] J. C. Bezdek, J. Keller, R. Krisnapuram, M. Pal (editors), *Fuzzy models and algorithms for pattern recognition and image processing*, Kluwer Academic Publishers, 1999.

[10] E. G. Birgin, L. F. Bueno, N. Krejić and J. M. Martínez, Low Order-Value Approach for Solving VaR-Constrained Optimization Problems, Technical Report, Department of Applied Mathematics, State University of Campinas, 2010. Available in *Optimization on line*.

[11] E. G. Birgin, C. A. Floudas and J. M. Martínez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, *Mathematical Programming* 125, pp. 139–162 (2010).

[12] E. G. Birgin, J. M. Martínez and D. P. Ronconi, Minimization subproblems and heuristics for an applied clustering problem. *European Journal of Operations Research* 146, pp. 19-34 (2003).

[13] J. V. Burke, A. S. Lewis and M. L. Overton, A robust gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM Journal on Optimization* 15, pp. 751-779 (2005).

[14] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classic in Applied Mathematics, SIAM, Philadelphia, 1990.

[15] A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[16] R. Fletcher and S. Leyffer, A bundle filter method for nonsmooth nonlinear Optimization, University of Dundee Numerical Analysis Report NA 195, December, 1999.

[17] R. Fletcher and S. Leyffer, Solving Mixed-Integer Nonlinear Programs by outer approximation, *Mathematical Programming* 66, pp. 327–349 (1994).

[18] A. Fuduli, M. Gaudioso and G. Gallombardo, Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, *SIAM Journal on Optimization* 14, pp. 743-756 (2004).

[19] A. Fuduli, M. Gaudioso and G. Gallombardo, A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization, *Optimization Methods and Software* 19, pp. 89–102 (2004).

[20] C. Gini, Measurement of Inequality of Incomes, *The Economic Journal* 31, pp. 124–126 (1921).

[21] P. Gouveia, *Global Optimization methods for Protein Alignment*, Tese de Doutorado, Departamento de Matemática Aplicada, Universidade Estadual de Campinas, Brasil, 2011.

[22] A. Grothey, A Second Order Trust Region Bundle Method for Nonconvex Nonsmooth Optimization, Technical Report MS 02-001, University of Edinburgh, 2002.

[23] N. Haakala, K. Miettinen and M. M. Makela, Globally convergent limited memory bundle method for large-scale nonsmooth optimization, *Mathematical Programming* 109, pp. 181-205 (2007).

[24] W. Hare and C. Sagastizábal, A redistributed proximal bundle method for nonconvex optimization, *SIAM Journal on Optimization* 20, pp. 2443-2473 (2010).

[25] L. Holm and C. Sander, Mapping the Protein Universe, *Science* 273, pp. 595–602 (1996).

[26] P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[27] P. Jorion, *Value at risk: the new benchmark for managing financial risk*, 2nd edition, Mc Graw-Hill, New York, 2001.

[28] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, New York, 1985.

[29] K. C. Kiwiel, Restricted step and Levenberg-Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization, *SIAM Journal on Optimization* 6, pp. 227-249 (1996).

[30] R. Kolodny, P. Koehl and M. Levitt, Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures, *Journal of Molecular Biology* 346, pp. 1173-1188 (2005).

[31] R. Kolodny and N. Linial, Approximate protein structural alignment in polynomial time, *Proceedings National Academy of Sciences USA* 101, pp. 12201–12206 (2004).

[32] A. S. Lewis and M. L. Overton, Nonsmooth Optimization via BFGS, submitted to *SIAM Journal on Optimization* (2010).

[33] R. Lima, *Protein Maps and Quick Alignments*, Tese de Doutorado, Departamento de Matemática Aplicada, Universidade Estadual de Campinas, Brasil, 2011.

[34] L. Lukŭsan and J. Vlćek, A bundle method for nonsmooth unconstrained minimization, *Mathematical Programming* 83, pp. 373-391 (1998).

[35] J. M. Martínez, Order-Value Optimization and new applications, in *Sixth International Congress on Industrial and Applied Mathematics: Zürich, Switzerland, July 16-20, 2007 – Invited Lectures*, R. Jeltsch and G. Wanner eds., European Mathematical Society, Zürich, 2009, pp. 279–296.

[36] L. Martínez, R. Andreani and J. M. Martínez. Convergent algorithms for Protein Structural Alignment, *BMC Bioinformatics* 8: 306 (2007).

[37] MINLP World, [www http://www.gamsworld.org/minlp], 2003.

[38] B. Needleman and C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology* 48, pp. 443-453 (1970).

[39] D. Noll, O. Prot and A. Rondepierre, A proximity control algorithm to minimize nonsmooth and nonconvex functions, *Pacific Journal on Optimization* 4, pp. 569-602 (2008).

[40] J. S. Pang and S. Leyffer, On the global minimization of the value-at-risk, *Optimization Methods and Software* 19, pp. 611-631 (2004).

[41] R. T. Rockafellar and S. Uryasev, Conditional value-at-risk for general loss distributions, *Journal of Banking and Finance* 26, pp. 1443-1471 (2002).

[42] L. Qi and P. Tseng, On almost smooth functions and piecewise smooth functions, *Nonlinear Analysis - Theory Methods and Applications* 67, pp. 773-794 (2007).

[43] A. Skajaa, *Limited Memory BFGS method for nonsmooth optimization*, Master's thesis, Courant Institute of Mathematical Science, New York University, 2010.

[44] H. Schramm and J. Zowe, A version of the bundle idea for minimizing a smooth function: Conceptual idea, convergence analysis, numerical results, *SIAM Journal on Optimization* 2, pp. 121–152 (1992).

[45] S. Subbiah, D. V. Laurents and M. Levitt, Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core, *Current Biology* 3, pp. 141-148 (1993).

[46] J. Vlcék and L. Lukšvsan, Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization, *Journal of Optimization Theory and Applications* 111, pp. 407-430 (2001).