Low Order-Value Approach for Solving VaR-Constrained Optimization Problems*

E. G. Birgin † L. F. Bueno ‡ N. Krejić § J. M. Martínez ‡

August 26, 2010; January 12, 2011 (revised)

Abstract

In Low Order-Value Optimization (LOVO) problems the sum of the r smallest values of a finite sequence of q functions is involved as the objective to be minimized or as a constraint. The latter case is considered in the present paper. Portfolio optimization problems with a constraint on the admissible Value at Risk (VaR) can be modeled in terms of a LOVO problem with constraints given by Low Order-Value functions. Different algorithms for practical solution of this problem will be presented. Using these techniques, portfolio optimization problems with transaction costs will be solved.

Key words: Optimization, Augmented Lagrangian, Order-Value Optimization, Low Order-Value Optimization, Value at Risk, Numerical algorithms.

MSC2000 classification codes: 65Kxx, 47N10.

1 Introduction

In the present contribution we consider optimization problems where Low Order-Value functions appear as constraints. We will report an application to portfolio optimization with transaction costs and Value-at-Risk constraints. A related problem in which a Low Order-Value function with smooth constraints is minimized was considered in a recent paper [11], where suitable algorithms were given and several applications were surveyed.

Portfolio optimization problems deal with allocation of wealth among different assets, in general, risky assets or combination of one risk-free asset and a number of risky ones. The objective is to select a combination that maximizes the expected future gain with a tolerable level of risk or to find a portfolio with the smallest risk among all portfolios that have at least some specified value of future expected gain. Modeling and measuring risk, as well as estimating future gains, are nontrivial tasks and considerable amount of research has been dedicated to these

*This work was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.510/2006 - APQ1), FAPESP (Grants 2006/53768-0 and 2005/57684-2) and CNPq.

†Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090 São Paulo SP, Brazil. e-mail: egbirgin@ime.usp.br

‡Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing (IMECC), University of Campinas, 13083-859 Campinas SP, Brazil. e-mail: {lfelipe|martinez}@ime.unicamp.br

§Department of Mathematics and Informatics, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia. e-mail: natasak@uns.ac.rs

topics (see, for example, [37]). Three mainstream risk measures are currently in use: portfolio variance, dating back to the pioneering work of Markowitz [29], Value at Risk (VaR) [25] and Conditional Value at Risk (CVaR) [36]. Their mutual relationships are analyzed in [22] and all three have advantages and disadvantages. More general risk measures are discussed in [15].

In this paper we will deal with VaR and portfolios composed of stocks. Current regulations for financial industry formulate risk requirements in terms of VaR and, therefore, VaR is a standard tool for risk management. By definition, VaR is the percentile of loss distribution given a confidence level α. The α-VaR of a financial instrument is the lowest amount such that the loss is less than or equal to it with probability α. Regulations usually require that the available capital should be a multiple of VaR. A comprehensive overview of VaR properties and its applications in risk management is given in [25].

The optimization problem which yields a portfolio with maximal gain and satisfies constraints on VaR is considered in [20, 21], where some algorithms are suggested. The problem is difficult due to the complicated geometry of the feasible set. In this paper we will include transaction costs in the portfolio optimization problem. These costs are inevitably present in real life and can significantly decrease portfolio yield. Several papers deal with portfolio optimization with transaction costs using different risk measures. In [16, 28, 30, 32] transaction costs are modeled as linear or piecewise linear functions.

Transaction costs can be divided into two types - fixed costs and impact costs. Fixed costs are fees and taxes, being in general proportional to the transaction value. Due to the rapid development of electronic trading in the last two decades and the large number of participants, fixed costs may not be a dominant part of the total costs, especially for large institutional investors. On the other hand they are still significant for small investors and thus need to be included into a realistic model. Describing impact costs is a far more complicated issue and there is no general agreement about the proper model in the literature. An impact cost is a deviation from the equilibrium price caused by one's own trading activity. If we are buying a large amount of one particular stock then we are obviously increasing the demand and thus increasing the price of that stock. It is difficult to distinguish between the price changes caused by one's trading activity and changes caused by noise. Large financial institutions use proprietary models to measure the impact. Although based on academic work, these models are not publicly available. Different model approaches were published in [4, 5, 18, 27]. In this work we will adopt the market impact model proposed in [4, 5].

Let us describe now the main features of the Order-Value Optimization tool. Assume that the functions $f_i : I\!R^n \to I\!R$, $i = 1, \ldots, q$, are given. For all $x \in I\!R^n$ let $i_1(x), \ldots, i_q(x)$ be such that:

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \ldots \leq f_{i_q(x)}(x) \tag{1}$$

and

$$\{i_1(x), \ldots, i_q(x)\} = \{1, \ldots, q\}.$$

See [7, 8, 10, 11, 12]. Let J be a non-empty subset of $\{1, \ldots, q\}$. The Order-Value function [31] associated with J is defined by

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

Let us give a few examples:

If $J = \{q\}$, we have that $f_J(x) = \sum_{j \in J} f_{i_j(x)}(x) = f_{i_q(x)}(x)$. Therefore, by (1), $f_J(x) = \max\{f_1(x), \ldots, f_q(x)\}$ in this case.

If $J = \{1\}$, we have $f_J(x) = f_{i_1(x)}(x)$. Then, by (1), $f_J(x) = \min\{f_1(x), \ldots, f_q(x)\}$.

When $J = \{r\}$, $f_J(x)$ may be interpreted as the Value-at-Risk (VaR) Order-Value function associated with "portfolio" x, scenarios $\{1, \ldots q\}$, loss functions $f_1, \ldots, f_q$ and confidence level $\alpha = r/q$ (see [7, 8, 12]). (In general $r < q$ and $r \approx q$.) Similarly, when $J = \{r+1, \ldots, q\}$, the function $f_J(x)/(q-r)$ corresponds to the "coherent" risk function CVaR (Conditional Value-at-risk) [36].

The Low Order-Value function introduced in [10, 11] corresponds to $J = \{1, \ldots, r\}$. The problem of minimizing f_J in this case has interesting applications to parameter estimation, hidden pattern problems, protein alignment and general structure alignment [10, 11, 31].

In this paper we are concerned with optimization problems in which there is a "VaR-constraint" of the form

$$f_{i_r(x)}(x) \leq c. \tag{2}$$

In the most typical situation one wishes to minimize the average loss associated with some investment subject to the condition that the VaR that corresponds to the optimal decision must not exceed the tolerance c. By (1), we have that (2) is equivalent to the "LOVO (Low Order-Value Optimization) constraints":

$$f_{i_j(x)}(x) \leq c, \; j = 1, \ldots, r. \tag{3}$$

Without loss of generality (re-defining $f_i(x) \leftarrow f_i(x) - c$) we will assume that $c = 0$ in (2) and (3). This means that, given $r \in \{1, \ldots, q\}$, our problem will be:

$$\text{Minimize } f(x) \text{ subject to } f_{i_r(x)}(x) \leq 0, \; h(x) = 0, \; g(x) \leq 0, \; x \in \Omega. \tag{4}$$

The set Ω will be given by *lower-level constraints* of the form

$$\underline{h}(x) = 0, \; \underline{g}(x) \leq 0.$$

Many times, Ω will take the form of an n-dimensional box:

$$\Omega = \{x \in I\!R^n \mid \ell \leq x \leq u\}.$$

We will assume that the functions $f : I\!R^n \to I\!R$, $f_i : I\!R^n \to I\!R$, $h : I\!R^n \to I\!R^m$, $g : I\!R^n \to I\!R^p$, $\underline{h} : I\!R^n \to I\!R^{\underline{m}}$, $\underline{g} : I\!R^n \to I\!R^{\underline{p}}$ have continuous first derivatives on $I\!R^n$.

By (1), problem (4) is equivalent to:

$$\text{Minimize } f(x) \text{ subject to } f_{i_j(x)}(x) \leq 0, \; j = 1, \ldots, r, \; h(x) = 0, \; g(x) \leq 0, \; x \in \Omega. \tag{5}$$

The present paper deals with the practical solution of problem (5). We are especially interested in large-scale cases, in which the number of assets n, the number of scenarios q, or both, are large. Problem (4) is nonlinear if transaction costs are incorporated into the objective function. Moreover, even in the linear case, its structure is not standard, so that its resolution needs the invention of modern optimization methods.

Our contributions in the present research are the following:

1. We introduce Augmented Lagrangian methods for solving the main problem (4) employing its equivalent formulation (5).

2. Algorithms based on local subproblem solvers will be proved to converge to stationary points of (5) and algorithms based on global optimization subproblem solvers will be shown to possess global minimization properties.

3. Computer implementations of the introduced Augmented Lagrangian algorithms will be provided and numerical experiments will be given.

4. The new algorithms will be applied to the optimization of portfolios with VaR constraints in the presence of transaction costs. We will show that the Augmented Lagrangian approach is a suitable tool for dealing with such problems. This approach deals well with large number of variables and, especially, with large number of inequality constraints.

This paper is organized as follows: An outline of the new algorithms will be given in Section 2. The smooth Augmented Lagrangian with lower-level constraints [6] will be recalled in Section 3. The novel algorithms will be defined in Sections 4, 5 and 6. In Section 7 we will present numerical experiments. Conclusions will be stated in Section 8.

Notation. We write $K_1 \underset{\infty}{\subset} K_2$ to indicate that K_1 is an infinite subsequence of indices contained in K_2. The symbol $\|\cdot\|$ denotes the Euclidean norm, although many times it may be replaced by an arbitrary norm on $I\!R^n$. For all $v \in I\!R^n$, we denote $v_+ = (\max\{0, v_1\}, \ldots, \max\{0, v_n\})^T$. For all $a \in I\!R$ we denote $a_+^2 = (a_+)^2$. We denote $\#I$ the number of elements of the set I.

2 Outline of algorithms

Section 3 of the present paper will be devoted to reviewing convergence results of the Augmented Lagrangian algorithms given in [6] and [17] for solving nonlinear programming problems. The standard constrained optimization problem will be defined in (6). Algorithm 3.1 is Algorithm 2.1 of [6] with minor modifications and Algorithm 3.2 corresponds to the global optimization Augmented Lagrangian method given in [17]. The difference between Algorithm 3.1 and Algorithm 2.1 of [6] is that, in Algorithm 3.1 we update $\rho_{k+1} \geq \rho_k$ when enough feasibility-complementarity progress has been obtained at the last outer iteration, while in Algorithm 2.1 of [6] one choses $\rho_{k+1} = \rho_k$ in that case. Convergence results for the modified algorithm follow exactly in the same way as in [6]. Both algorithms in Section 3 will be used as auxiliary tools in the remaining sections of the paper. The difference between Algorithm 3.1 and Algorithm 3.2 corresponds to the difference between [6] and [17]. In the case of Algorithm 3.1 one employs a local optimization algorithm for solving the Augmented Lagrangian subproblems, whereas in Algorithm 3.2 one uses global subproblem optimization.

Sections 4, 5 and 6 will be dedicated to the solution of the main problem (5). In Section 4 two algorithms will be introduced for that purpose, namely, Algorithms 4.1 and 4.2. Both incorporate the constraints $f_{i_j(x)}(x) \leq 0$, $j = 1, \ldots, r$, as "penalized" constraints in the definition of the Augmented Lagrangian function. As a consequence, the objective function of each subproblem is of LOVO type. This means that, for solving the non-smooth subproblems in Algorithm 4.1 we may use the method for minimizing LOVO functions introduced in [11]. On the other hand, for solving the subproblems in Algorithm 4.2, we must use a global optimization algorithm. In this case, convergence to global minimizers is guaranteed by the theory presented in [17].

In Section 5 we will introduce two different methods of Augmented Lagrangian type, namely, Algorithms 5.1 and 5.2. Unlike Section 4, the algorithms introduced in Section 5 use smooth

Augmented Lagrangian functions instead of LOVO-like Augmented Lagrangians as objective functions of the subproblems. At the beginning of each outer iteration the indices $i_1(x), \dots, i_r(x)$ that define penalized constraints for the current subproblem are defined. These indices are fixed throughout the execution of the outer iteration. The difference between Algorithm 5.1 and Algorithm 5.2 is that, in the first, we require approximate local minimizers of the subproblems whereas in Algorithm 5.2, ε-global optimizers are employed. As expected, there is a price to be paid for using smooth problems: The convergence results proved in Section 5 are weaker than the ones proved for Algorithms 4.1 and 4.2 in Section 4.

Finally, in Section 6 we will define a simple fixed-point algorithm that, at each iteration, solves a smooth constrained optimization problem of type (6). This seems to be the most obvious approach to the solution of (5) but its convergence properties are weaker than the ones proved in Sections 4 and 5. The introduction of specific algorithms in Sections 3, 4 and 5 will be preceded by the definition of Conceptual Algorithms. In the Conceptual Algorithms we require that each iterate x^k should be an "approximate solution" of the Augmented Lagrangian subproblem, without specifying the conditions that such solution should satisfy. Therefore, the specific algorithms introduced in each section may be considered practical realizations of the respective conceptual algorithms.

In other words, we will present two algorithms in each section, the first of which converges to stationary points of the main problem (5) and the second has global minimization properties. In Section 4 the objective functions of the subproblems are nonsmooth, since a new ordering of the functions $f_i(x)$ is computed at each call to the Augmented Lagrangian function. In the algorithms of Section 5 the quantities $f_1(x), \dots, f_q(x)$ are sorted only at the beginning of each outer iteration. Therefore, the objective functions of the subproblems are smooth. Finally, the algorithms considered in Section 6 are of fixed-point type. At each fixed-point iteration a whole smooth Nonlinear Programming solver is solved using Augmented Lagrangians and the same set $i_1(x), \dots, i_q(x)$ is used all along the fixed-point iteration.

3 Smooth Augmented Lagrangian algorithm

For completeness, in this section we recall the main algorithms presented in [6] and [17]. These algorithms are based on the Powell-Hestenes-Rockafellar (PHR) Augmented Lagrangian approach [23, 33, 35]. Let us assume that $f : I\!R^n \to I\!R, h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p, \underline{h} : I\!R^n \to I\!R^{\underline{m}}, \underline{g} : I\!R^n \to I\!R^{\underline{p}}$ are continuous functions. Algorithm 3.1 requires continuity of the gradients, whereas Algorithm 3.2 does not. We wish to solve the problem

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{6}$$

The constraints $\underline{h}(x) = 0$ and $\underline{g}(x) \le 0$ are called "lower-level constraints". In the case that the lower-level constraints are given by an n-dimensional box, the algorithm defined in this section corresponds to Algencan (the optimization method introduced in [6] and available at the Tango project web page [38])

Given $\rho > 0$, $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$, $x \in I\!R^n$, we define the PHR Augmented Lagrangian:

$$\mathcal{L}_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2\right].$$

Conceptual Algorithm 3.0. Let $\varepsilon_k \downarrow 0$, $\bar{\lambda}^k \in [\lambda_{\min}, \lambda_{\max}]^m$, $\bar{\mu}^k \in [0, \mu_{\max}]^p$ for all $k \in I\!N$, $\rho_1 > 0$, $\tau \in (0, 1)$, $\eta > 1$. For all $k = 1, 2, \dots$ we compute $x^k \in I\!R^n$ as an approximate solution of

$$\text{Minimize } \mathcal{L}_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \leq 0. \tag{7}$$

We define, for all $i = 1, \dots, p$,

$$V_i^k = \max\left\{g_i(x^k), \frac{-\bar{\mu}_i^k}{\rho_k}\right\}.$$

If $k = 1$ or

$$\max\{\|h(x^k)\|, \|V^k\|\} \leq \tau \max\{\|h(x^{k-1})\|, \|V^{k-1}\|\} \tag{8}$$

we define $\rho_{k+1} \geq \rho_k$. Else, we define $\rho_{k+1} \geq \eta\rho_k$.

Algorithm 3.1. Proceed as in the Conceptual Algorithm 3.0, with x^k defined in such a way that there exist $v^k \in I\!R^{\underline{m}}$ and $w^k \in I\!R_+^{\underline{p}}$ satisfying:

$$\|\nabla\mathcal{L}_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \nabla\underline{h}(x^k)v^k + \nabla\underline{g}(x^k)w^k\| \leq \varepsilon_k,$$

$$\|\underline{h}(x^k)\| \leq \varepsilon_k,\ \|\underline{g}(x^k)_+\| \leq \varepsilon_k,$$

and

$$w_i^k = 0 \text{ whenever } \underline{g}_i(x^k) < -\varepsilon_k.$$

Remark. In [6], one defines $\rho_{k+1} = \rho_k$ if (8) holds and $\rho_{k+1} = \eta\rho_k$ otherwise. Here we adopt a more general form, in which

$$\rho_{k+1} \geq \rho_k \ \text{ and } \ \rho_{k+1} \geq \eta\rho_k, \tag{9}$$

respectively. In this way, it will be easier to interpret the forthcoming methods in terms of Algorithm 3.1.

The convergence properties of Algorithm 3.1 are stated in the following theorem.

Theorem 3.1. *Assume that x^* is a limit point of a sequence generated by Algorithm 3.1. Then, one of the following three possibilities hold:*

1. *x^* is a feasible point of (6).*
2. *x^* is a Karush-Kuhn-Tucker (KKT) point of the problem*

$$\textit{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 \textit{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \leq 0.$$

3. *The constraints $\underline{h}(x) = 0$ and $\underline{g}(x) \leq 0$ do not satisfy the Constant Positive Linear Dependence (CPLD) constraint qualification [13, 34] at x^** [1]*.*

If x^* is a feasible point of (6) then one of the following two possibilities hold:

1. *x^* is a KKT point of problem (6).*
2. *The constraints $h(x) = 0$, $g(x) \leq 0$, $\underline{h}(x) = 0$, $\underline{g}(x) \leq 0$ do not satisfy the CPLD constraint qualification at x^*.*

[1] We say that CPLD condition is satisfied at x^* if the linear dependence of gradients of active constraints with non-negative coefficients corresponding to inequalities implies the linear dependendence of the same gradients in a neighborhood of x^*

Proof. See the proofs of Theorems 4.1 and 4.2 of [6]. Observe that the modifications (9) do not interfere in the proof.

The global optimization counterpart of Algorithm 3.1 was defined in [17]. We state the global Augmented Lagrangian method of [17] as Algorithm 3.2 below. The difference between these two algorithms is that in Algorithm 3.2 the iterate x^k is obtained as an ε_k-global minimizer of the Augmented Lagrangian.

Algorithm 3.2. Proceed as in Algorithm 3.0, where $x^k \in \Omega$ is such that

$$\mathcal{L}_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq \mathcal{L}_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k) + \varepsilon_k$$

for all $x \in \Omega$.

The following theorem was proved in [17], where a comprehensive set of numerical experiments using the α-BB algorithm [1, 2, 3, 14] for solving the subproblems were given.

Theorem 3.2. *Assume that the feasible region of problem (6) is non-empty. Then, every limit point of a sequence generated by Algorithm 3.2 is a global minimizer of (4).*

4 Algorithm with LOVO subproblems

In [11] an Augmented Lagrangian method (C-LOVO) was defined to minimize a Low Order-Value function with smooth constraints. Essentially, C-LOVO consists of applying Algencan to the problem, "ignoring" the fact that first derivatives may not be defined. Every cluster point of a sequence generated by C-LOVO is feasible or stationary for the sum of squares of infeasibilities. Moreover, any feasible cluster point satisfies Karush-Kuhn-Tucker (KKT) conditions, provided that the constraints fulfill the Constant Positive Linear Dependence (CPLD) constraint qualification [13, 34]. This state of facts makes it desirable to solve LOVO-constrained problems like (5) by means of a sequence of constrained problems with Low Order-Value objective function. In this section we introduce an algorithm with those characteristics. The algorithm will be, as C-LOVO, of Augmented Lagrangian type. Moreover, at each outer iteration, a subproblem will be approximately solved using C-LOVO.

Given $\rho > 0$, $\lambda \in I\!R^m$, $\mu \in I\!R^p_+, \nu \in I\!R_+$, $x \in I\!R^n$, we define the LOVO Augmented Lagrangian function $L_\rho(x, \lambda, \mu, \nu)$ by:

$$L_\rho(x, \lambda, \mu, \nu) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2 + \sum_{j=1}^{r}\left(f_{i_j(x)}(x) + \frac{\nu}{\rho}\right)_+^2\right]. \qquad (10)$$

For computing (10), given $x \in I\!R^n$ we need to compute the r smallest values of $\{f_1(x), \ldots, f_q(x)\}$. Therefore, this sorting procedure must be performed each time we need to evaluate the Augmented Lagrangian (10). This is the main difference between the algorithms given in this section and the ones given in Sections 5 and 6, in which sorting of $f_1(x), \ldots, f_q(x)$ is less frequent.

At each (outer) iteration, the algorithms introduced in this section (approximately) minimize $L_\rho(x, \lambda, \mu, \nu)$ subject to $x \in \Omega$. Let us justify why we employ the objective function L_ρ, given by (10), instead of the (perhaps more intuitive) "Augmented Lagrangian" $\tilde{L}_\rho$ associated with problem (4). In this case, we would have:

$$\tilde{L}_\rho(x,\lambda,\mu,\nu) = f(x) + \frac{\rho}{2}\left[\left\|h(x)+\frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x)+\frac{\mu}{\rho}\right)_+\right\|^2 + \left(f_{i_r(x)}(x)+\frac{\nu}{\rho}\right)_+^2\right]. \tag{11}$$

The key point is that minimizing the nonsmooth function L_ρ is easier than minimizing $\tilde{L}_\rho$. In fact, if , given x, one finds a trial point z such that

$$\Phi_x(z) \equiv f(z) + \frac{\rho}{2}\left[\left\|h(z)+\frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(z)+\frac{\mu}{\rho}\right)_+\right\|^2 + \sum_{j=1}^r \left(f_{i_j(x)}(z)+\frac{\nu}{\rho}\right)_+^2\right] < L_\rho(x,\lambda,\mu,\nu), \tag{12}$$

then, automatically,

$$L_\rho(z,\lambda,\mu,\nu) < L_\rho(x,\lambda,\mu,\nu).$$

Note that indices $i_j(x)$, $i = 1,\dots,r$, are fixed in the definition of $\Phi_x(z)$, in contrast with the indices in the definition of $L_\rho(x,\lambda,\mu,\nu)$ in (10), that depend on the variable x. Hence, obtaining (12) is not difficult because it amounts to find a descent direction for the smooth function $\Phi_x(z)$. As a consequence, suitable global convergence properties are obtained [11]. On the other hand,

$$f(z) + \frac{\rho}{2}\left[\left\|h(z)+\frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(z)+\frac{\mu}{\rho}\right)_+\right\|^2 + \left(f_{i_r(x)}(z)+\frac{\nu}{\rho}\right)_+^2\right] < \tilde{L}_\rho(x,\lambda,\mu,\nu)$$

does not imply

$$\tilde{L}_\rho(z,\lambda,\mu,\nu) < \tilde{L}_\rho(x,\lambda,\mu,\nu). \tag{13}$$

In this case, the fact that $i_r(z)$ is, in general, different than $i_r(x)$ inhibits the fulfillment of the desirable property (13). These observations also support the use of (5) instead of (4). The fact that LOVO problems are easier to solve than OVO problems [7, 8] is exploited in [11].

The derivatives of L_ρ with respect to x may not exist at the points x in which the set of indices that define the r smallest values of $f_i(x)$ is not univocally defined. However, to simplify the notation, we write:

$$\nabla L_\rho(x,\lambda,\mu,\nu) = \nabla f(x) + \frac{\rho}{2}\left\{\nabla\left(\left\|h(x)+\frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x)+\frac{\mu}{\rho}\right)_+\right\|^2\right) + \sum_{j=1}^r \nabla\left[\left(f_{i_j(x)}(x)+\frac{\nu}{\rho}\right)_+^2\right]\right\}.$$

Conceptual Algorithm 4.0. The parameters that define the algorithm are: $\tau \in [0,1)$, $\eta > 1$, $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$. At the first outer iteration we use a penalty parameter $\rho_1 > 0$ and safeguarded Lagrange multipliers estimates $\bar{\lambda}^1 \in I\!R^m$, $\bar{\mu}^1 \in I\!R^p_+$, $\bar{\nu}_1 \in I\!R_+$ such that $\bar{\lambda}^1_i \in [\lambda_{\min}, \lambda_{\max}]$, $i = 1,\dots,m$, $\|\bar{\mu}^1\|_\infty \le \mu_{\max}$ and $\bar{\nu}_1 \le \nu_{\max}$. We assume that $x^0 \in I\!R^n$ is an arbitrary initial point and $\{\varepsilon_k\}$ is a sequence of positive numbers that satisfies $\lim_{k\to\infty} \varepsilon_k = 0$.

Step 1. *Initialization.*
Set $k \leftarrow 1$.

Step 2. *Solve the subproblem.*
Compute $x^k \in I\!R^n$ as an approximate solution of

$$\text{Minimize } L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{14}$$

Step 3. *Update penalty parameter.*

For all $i = 1, \dots, p$, compute $V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}$.

For all $i = 1, \dots, q$, compute $W_i^k = \begin{cases} \max\left\{f_i(x^k), -\frac{\bar{\nu}_k}{\rho_k}\right\}, & \text{if } i \in \{i_1(x^k), \dots i_r(x^k)\}, \\ 0, & \text{otherwise.} \end{cases}$

Compute

$$R_k = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty, \|W^k\|_\infty\}.$$

If $k > 1$ and $R_k > \tau R_{k-1}$, define $\rho_{k+1} = \eta\rho_k$. Else, define $\rho_{k+1} = \rho_k$.

Step 4. *Estimate multipliers.*

Compute $\bar{\lambda}_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}]$ for all $i = 1, \dots, m$, $\bar{\mu}_i^{k+1} \in [0, \mu_{\max}]$ for all $i = 1, \dots, p$, and $\bar{\nu}_{k+1} \in [0, \nu_{\max}]$. Set $k \leftarrow k + 1$ and go to Step 2.

Algorithm 4.1. Proceed as in the Conceptual Algorithm 4.0, where $x^k \in \mathbb{R}^n$ is such that there exist $v^k \in \mathbb{R}^{\underline{m}}$ and $w^k \in \mathbb{R}^{\underline{p}}$ satisfying

$$\|\nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) + \textstyle\sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} w_i^k \nabla \underline{g}_i(x^k))]\| \le \varepsilon_k, \tag{15}$$

$$w^k \ge 0, \ \underline{g}(x^k) \le \varepsilon_k, \tag{16}$$

$$\underline{g}_i(x^k) < -\varepsilon_k \Rightarrow w_i^k = 0 \ \text{ for all } \ i = 1, \dots, \underline{p}, \tag{17}$$

$$\|\underline{h}(x^k)\| \le \varepsilon_k. \tag{18}$$

4.1 Solvability of the subproblems

At each iteration of Algorithm 4.1 we minimize, approximately, $L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k)$ with respect to x on the set defined by $\underline{h}(x) = 0$ and $\underline{g}(x) \le 0$. The stopping criterion for the corresponding iterative process is reflected in the conditions (15–18). We want to show that obtaining (15–18) is possible in finite time using a computable algorithm.

Let us define:

$$F_{\min}(x) = \min_{I \mid \#I = r} \left\{ f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2 + \sum_{i \in I}\left(f_i(x) + \frac{\nu}{\rho}\right)_+^2\right]\right\}.$$

The subproblem

$$\text{Minimize } L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) \ \text{ subject to } \underline{h}(x) = 0, \ \underline{g}(x) \le 0$$

is equivalent to

$$\text{Minimize } F_{\min}(x) \text{ subject to } \underline{h}(x) = 0, \ \underline{g}(x) \le 0.$$

This is a LOVO problem with constraints as defined in [11]. Therefore, one can employ Algorithm C-LOVO of [11] for its resolution. Thus, assuming that the feasible set $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ is non-empty and that the set defined by $\underline{g}(x) \le \varepsilon$ is bounded for some $\varepsilon > 0$, we have that the conditions (15–18) are fulfilled by some iterate of C-LOVO in finite time.

4.2 Convergence of Algorithm 4.1

In the previous section we saw that Algorithm 4.1 is well defined. Here we wish to analyze its convergence properties. From now on we will assume that $K \underset{\infty}{\subset} I\!N$ is such that

$$\lim_{k \in K} = x^*. \tag{19}$$

Since the number of subsets of $\{1, \ldots, q\}$ is finite, there exist $K_1 \underset{\infty}{\subset} K$, $I \subset \{1, \ldots, q\}$, $\#I = r$, such that, for all $k \in K_1$,

$$\{i_1(x^k), \ldots, i_r(x^k)\} = I. \tag{20}$$

For all $k \in K_1$, $i \in I, j \notin I$ one has that:

$$f_i(x^k) \le f_j(x^k). \tag{21}$$

Taking limits in (21) we see that for all $i \in I, j \notin I$,

$$f_i(x^*) \le f_j(x^*). \tag{22}$$

With these definitions, thanks to (9), the sequence satisfied by Algorithm 4.1 may be thought as being generated by Algorithm 3.1, applied to the problem:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0 \ \forall\, i \in I,\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{23}$$

Therefore, the convergence result given in Theorem 3.1 holds for this sequence. The global convergence result is condensed in the theorem below.

Theorem 4.1. *Assume that x^*, K, K_1 and I satisfy (19,20). Then, one of the following three possibilities hold:*

1. *x^* is a feasible point of (5).*
2. *x^* is a KKT point of*

$$\textit{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 + \sum_{i \in I} f_i(x)_+^2 \textit{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0.$$

3. *The constraints $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ do not satisfy the Constant Positive Linear Dependence (CPLD) constraint qualification [13, 34] at x^*.*

If x^* is a feasible point of (5) then one of the following two possibilities hold:

1. *x^* is a KKT point of problem (23).*
2. *The constraints $h(x) = 0$, $g(x) \le 0$, $f_i(x) \le 0 \ \forall\, i \in I$, $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ do not satisfy the CPLD constraint qualification at x^*.*

Remark. A nice practical consequence of Theorem 4.1 is that, in general, the limit point x^* generated by the algorithm satisfies optimality conditions of the nonlinear programming problem (23), where the LOVO constraints $f_i(x) \le 0$, $i \in I$, correspond to the r smallest values of $f_1(x^*), \ldots, f_q(x^*)$ (22). The importance of this property will become apparent when we introduce the algorithms of the following sections.

4.3 Global optimization with LOVO subproblems

As in the case of Algorithm 3.0, we are going to see that a version of Algorithm 4.0 converges to global minimizers of (5). As in Section 3 we only need to find approximate global minimizers of the subproblems.

Algorithm 4.2. Proceed as in the Conceptual Algorithm 4.0, with $x^k \in \Omega$ being such that

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \nu_k) \leq L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \nu_k) + \varepsilon_k$$

for all $x \in \Omega$.

Observe that the functions $\bar{f}_j$ defined by $\bar{f}_j(x) = f_{i_j(x)}(x)$ are continuous and that the LOVO constraints of the problem are $\bar{f}_j(x) \leq 0$, $j = 1, \ldots, r$. Therefore, Algorithm 4.2 is a particular case of the main algorithm of [17]. As a consequence, by Theorem 2 of [17], we may prove the following theorem.

Theorem 4.2. *Assume that the feasible region of problem (5) is non-empty. Then, every limit point of a sequence generated by Algorithm 4.2 is a global minimizer of (5).*

5 Algorithm with smooth subproblems

At each iteration of Algorithm 4.0 one solves an optimization problem with a Low Order-Value objective function. Conditions (15–18) are the approximate optimality conditions for this subproblem. This nonsmooth subproblem may be attacked using C-LOVO [11], but the alternative of using smooth subproblems deserves careful consideration. The idea consists of defining, at the beginning of each outer iteration k,

$$I(x^{k-1}) = \{i_1(x^{k-1}), \ldots, i_r(x^{k-1})\} \tag{24}$$

and to fix this set of indices in the Augmented Lagrangian definition. Namely, instead of (10), we define:

$$L_\rho^k(x, \lambda, \mu, \nu) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2 + \sum_{j \in I(x^{k-1})} \left(f_j(x) + \frac{\nu}{\rho}\right)_+^2\right]. \tag{25}$$

Again, it is pertinent to ask why we use the Augmented Lagrangian L_ρ^k instead of its counterpart $\tilde{L}_\rho^k$, associated with problem (4) and defined by:

$$\tilde{L}_\rho^k(x, \lambda, \mu, \nu) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2 + \left(f_{i_r(x^{k-1})}(x) + \frac{\nu}{\rho}\right)_+^2\right].$$

The reason is the following: After minimizing L_ρ^k, we obtain, hopefully, a point x such that $f_j(x) \leq 0$ for at least r indices j (those belonging to $I(x^{k-1})$). This implies that $f_{i_r(x)}(x) \leq 0$. On the other hand, the best we can expect from minimizing $\tilde{L}_\rho^k$ is a point x such that $f_{i_r(x^{k-1})}(x) \leq 0$. Since, very likely, the order of the $f_i's$ changes from one iteration to another, this property does not imply that $f_{i_r(x)}(x) \leq 0$.

Function (25) has continuous first derivatives, so that its gradient does not need a special definition. The Conceptual Algorithm 5.0 is identical to Algorithm 4.0 except that L_{ρ_k} is replaced by $L^k_{\rho_k}$ in (15).

Conceptual Algorithm 5.0. Define τ, η, $\lambda_{\min}$, $\lambda_{\max}$, $\mu_{\max}$, ρ_1, $\bar{\lambda}^1$, $\bar{\mu}^1, \bar{\nu}_1$ and ε_k as in the Conceptual Algorithm 4.0. Steps 1, 3 and 4 are the same as in that algorithm. At Step 2, the point $x^k \in \mathbb{R}^n$ is computed as an approximate solution of

$$\text{Minimize } L^k_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{26}$$

Algorithm 5.1. Proceed as in the Conceptual Algorithm 5.0. For computing an approximate solution of (26), proceed as in Algorithm 4.1, replacing condition (15) by:

$$\|\nabla L^k_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) + \sum_{i=1}^{m} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{p} w_i^k \nabla \underline{g}_i(x^k))]\| \le \varepsilon_k. \tag{27}$$

Conditions (16–18) remain as in Algorithm 4.1.

5.1 Solvability of the subproblems

At each iteration of Algorithm 5.1 we minimize, approximately, $L^k_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k)$ with respect to x on the set defined by $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$. The stopping criterion for the corresponding iterative process is given by the conditions (27) and (16–18).

As in the case of Algorithm 4.1, obtaining the stopping criterion is possible in finite time using a computable algorithm. In this case, if we define:

$$F_{\min}(x) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2 + \sum_{j \in I(x^{k-1})} \left(f_j(x) + \frac{\nu}{\rho}\right)_+^2\right],$$

the subproblem

$$\text{Minimize } L^k_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0$$

is equivalent to

$$\text{Minimize } F_{\min}(x) \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0.$$

This is a smooth optimization problem with constraints. Therefore, one can employ Algorithm 3.1 for its resolution. Assuming that the feasible set $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ is non-empty and that the set defined by $\underline{g}(x) \le \varepsilon$ is bounded for some $\varepsilon > 0$, we know that conditions (25) and (16–18) are fulfilled by some Augmented Lagrangian iteration [6].

5.2 Convergence

In this section we analyze the convergence properties of Algorithm 5.1. Assume, as in Section 4.2, that $K \underset{\infty}{\subset} \mathbb{N}$ is such that

$$\lim_{k \in K} x^k = x^*. \tag{28}$$

Taking an appropriate subsequence and re-labeling we assume:

$$\lim_{k\in K} x^{k-1} = x^{**}. \tag{29}$$

Since the number of subsets of $\{1,\ldots,q\}$ is finite, there exist $K_1 \underset{\infty}{\subset} K$, $I \subset \{1,\ldots,q\}$, $\#I = r$, such that, for all $k \in K_1$,

$$\{i_1(x^{k-1}),\ldots,i_r(x^{k-1})\} = I. \tag{30}$$

For all $k \in K_1$, $i \in I, j \notin I$ one has that:

$$f_i(x^{k-1}) \le f_j(x^{k-1}). \tag{31}$$

Taking limits in (31) we see that for all $i \in I, j \notin I$,

$$f_i(x^{**}) \le f_j(x^{**}). \tag{32}$$

With these definitions, the sequence generated by Algorithm 5.1 may be thought as being generated by Algorithm 3.1, applied to the problem:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0\ \forall\, i \in I,\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{33}$$

Therefore, Theorem 3.1 can be applied. The global convergence result is stated in the theorem below.

Theorem 5.1. *Assume that x^*, x^{**}, K, K_1 and I satisfy (28–30). Then, one of the following three possibilities hold:*

1. *x^* is a feasible point of (5).*
2. *x^* is a KKT point of*

$$\text{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 + \sum_{i\in I} f_i(x)_+^2 \text{ subject to } \underline{h}(x) = 0,\ \underline{g}(x) \le 0.$$

3. *The constraints $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ do not satisfy the Constant Positive Linear Dependence (CPLD) constraint qualification [13, 34] at x^*.*

If x^* is a feasible point of (33) then one of the following two possibilities hold:

1. *x^* is a KKT point of problem (33).*
2. *The constraints $h(x) = 0$, $g(x) \le 0$, $f_i(x) \le 0\ \forall\, i \in I$, $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$ do not satisfy the CPLD constraint qualification at x^*.*

Remark. In the case that

$$x^{**} = \lim_{k\in K_1} x^{k-1} = \lim_{k\in K_1} x^k = x^*, \tag{34}$$

the result of Theorem 5.1 is the same as the one of Theorem 4.1. However, (34) is a property of the algorithmic sequence (not of the problem) and, therefore, might not be verified by the sequence generated by Algorithm 5.1. In any case, the limit point x^* generally satisfies all the constraints $f_i(x) \le 0, i \in I$. Thus, $f_i(x^*) \le 0$ for at least r indices i. This means that the constraint $f_{i_r(x)}(x) \le 0$ is certainly satisfied by x^* but the possibility exists that $f_{i_{r+s}}(x^*) \le 0$ also for some $s > 0$. Since x^* possibly satisfies more constraints than necessary, the point x^* is, perhaps, merely suboptimal.

To see the practical counterpart of the observation above we need to discuss first the adequate stopping criterion for Algorithm 5.1. By Theorem 5.1, we may expect that limit points of the algorithm solve problem (33). Therefore, the algorithm should stop at the iterate x^k when x^k approximately fulfills a KKT condition for this problem. (For a discussion on approximate KKT conditions, see [9].) In other words, the algorithm should stop at x^k when x^k is, presumably, an approximate solution of

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0\ \forall\, i \in I(x^{k-1}),\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{35}$$

Since (with some small tolerance ε) $f_i(x^k) \le 0$ for all $i \in I(x^{k-1})$ and $I(x^{k-1})$ contains r indices, it turns out that $f_i(x^k) \le 0$ for all $i \in I(x^k)$ (with tolerance ε). Therefore, x^k is a probable minimizer of $f(x)$ on a set that, disregarding the tolerance, is (perhaps strictly) contained in the feasible set of the following problem:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0\ \forall\, i \in I(x^k),\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{36}$$

This means that minima of (35) could be greater than minima of (36). Then, albeit Algorithm 5.1 generally computes (almost) feasible points of problem (5), its iterates could be worse than the ones generated by Algorithm 4.1.

5.3 Global optimization properties

Following the ideas of previous sections, it is natural to ask for the properties of Algorithm 5.1 when, instead of (27), (16), (17) and (18), we require that x^k should be an approximate global minimizer of $L^k_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k)$ with respect to $x \in \Omega$.

Algorithm 5.2. Proceed as in Algorithm 5.1 except that $x^k \in \Omega$ is such that

$$L^k_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) \le L^k_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k, \bar{\nu}_k) + \varepsilon_k$$

for all $x \in \Omega$.

The global optimization properties of Algorithm 5.2 are given in the following theorem.

Theorem 5.2. *Assume that the feasible region of problem (5) is non-empty. Let $\{x^k\}$ be generated by Algorithm 5.2 and let x^* be a limit point, x^{**}, K, K_1 and I satisfy (28–30). Then, x^* is a global solution of problem (33).*

Proof. As in Section 5.2, we may assume that the subsequence that converges to x^* is generated by Algorithm 3.2 applied to problem (33). Then, by Theorem 3.2, x^* is a global solution of (33), as we wanted to prove.

Note that, even in the case that $x^* = x^{**}$ this result is weaker than the one obtained in Theorem 4.2 for Algorithm 4.2. In fact, if $x^* = x^{**}$, it follows that x^* is a global minimizer of $f(x)$

subject to $h(x) = 0$, $g(x) \leq 0$, $x \in \Omega$ and $f_i(x) \leq 0$ for all $i \in I$ where $I = \{i_1(x^*), \dots, i_r(x^*)$, but this does not imply that x^* is a solution of (5). For example, take the problem

$$\text{Minimize } x \text{ subject to } \min\{f_1(x), f_2(x)\} \leq 0,$$

where $f_1(x) = (x-1)^2 - 1, f_2(x) = (x+1)^2 - 1$. Take $x^* = 0$. Then, we may consider that $r = 1$, $i_1(x^*) = 1, i_2(x^*) = 2$. Clearly, x^* is a global minimizer of the objective function subject to $f_{i_1(x^*)}(x) \leq 0$ but the global minimizer of the problem is -2.

6 Fixed-point LOVO algorithms

The algorithms presented in this section for solving (5) will be of fixed-point type. The idea is to solve the problem by means of a small number (perhaps only one) smooth constrained optimization problem. For all $k = 1, 2, \dots$ we define $I(x^{k-1})$ as in (24). The definition of the Conceptual Algorithm is as follows.

Algorithm 6.0. We assume that $x^0 \in I\!R^n$ is an arbitrary initial point.

Step 1. *Initialization.*
Set $k \leftarrow 1$.

Step 2. *Solve the subproblem.*
Compute x^k as a (possible) solution of

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \leq 0,\ f_i(x) \leq 0\ \forall\, i \in I(x^{k-1}),\ \underline{h}(x) = 0,\ \underline{g}(x) \leq 0. \tag{37}$$

Step 3. If $I(x^{k-1}) = I(x^k)$, define $x^{\text{final}} = x^k$ and stop. Else, set $k \leftarrow k+1$ and go to Step 1.

An alternative to problem (37) could be:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \leq 0,\ f_{i_r(x^{k-1})}(x) \leq 0,\ \underline{h}(x) = 0,\ \underline{g}(x) \leq 0. \tag{38}$$

However, in the case of solving successfully (38), we would only obtain $f_{i_r(x^{k-1})}(x^k) \leq 0$. That is, the fulfillment of $f_j(x) \leq 0$ would be guaranteed for only one index j. On the other hand, solving successfully (37) leads to a point x^k that satisfies $f_j(x) \leq 0$ for at least r indices. Therefore, $f_{i_r(x^k)}(x^k) \leq 0$.

By a "possible" solution of problem (37) we understand a limit point of a sequence generated by a smooth constrained optimization algorithm. We may use Algorithm 3.1 or Algorithm 3.2 for that purpose. If

$$I(x^k) = I(x^{k-1}) \tag{39}$$

then, very likely, x^{k+s} would be identical to x^k for all $s = 1, 2, \dots$. Therefore, the identity between $I(x^{k-1})$ and $I(x^k)$ is a sensible practical stopping criterion for Algorithm 6.0.

Algorithm 6.0 resembles the third algorithm proposed in the paper of Gaivoronski and Pflug [21]. These authors considered the case of linear objective function, returns, and additional constraints, observing that, in this case, (37) reduces to a Linear Programming problem. Instead of using always $I(x^{k-1})$ for risk constraints, they suggest to select a suitable set of r scenarios using adequate heuristics.

Algorithms 6.1 and 6.2 are defined and the local and global counterparts of Algorithm 6.0. Namely, in Algorithm 6.1 we are supposed to use Algorithm 3.1 for computing the fixed-point iteration, whereas in Algorithm 6.2 we use Algorithm 3.2 for the same purpose.

Theorem 6.1. *Assume that we use Algorithm 3.1 and the sequence $\{x^k\}$ stops at the feasible point x^{final}. Then, at least one of the two following possibilities hold:*

1. $x^{\rm final}$ *is a KKT point of the problem*

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0\ i \in I(x^{\rm final}),\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0. \tag{40}$$

2. *The constraints of problem (40) do not satisfy the CPLD constraint qualification at x^{final}.*

Proof. The proof follows from Theorem 3.1 using the identity (39).

The proof of the final theorem of this section follows directly from the definition of Algorithm 6.2 and the fact that $I(x^k) = I(x^{k-1})$ at the final iterate.

Theorem 6.2. *Assume that we use Algorithm 6.2 and the sequence $\{x^k\}$ stops at the feasible point x^{final}. Then, x^{final} is a global solution of*

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0,\ f_i(x) \le 0\ \forall\, i \in I(x^k),\ \underline{h}(x) = 0,\ \underline{g}(x) \le 0.$$

Note that the thesis of Theorem 6.2 does not imply that $x^{\rm final}$ is a solution of the original problem (5). Similarly to Section 5, consider the problem:

$$\text{Minimize } x \text{ subject to } \min\{f_1(x), f_2(x)\} \le 0,$$

where $f_1(x) = (x-1)^2-1, f_2(x) = (x+2)^2-1$. Taking $r = 1$ this problem has the form (5). Take $x^* = 0$. Then, $i_1(x^*) = 1, i_2(x^*) = 2$. Clearly, x^* is a global minimizer of the objective function subject to $f_{i_1(x^*)}(x) \le 0$, and, consequently, x^* is a fixed-point of the algorithm. However, the global minimizer of the problem is -3.

This means that the only algorithm presented in this paper that is completely satisfactory from the point of view of global minimization is Algorithm 4.2. This is because Algorithm 4.2 fits completely the global theory of [17] whereas Algorithms 5.2 and 6.2 do not.

7 Numerical experiments

We implemented the algorithms introduced in Sections 4, 5 and 6 for solving problem (5), making suitable modifications of Algencan and using all the Algencan default parameters. More specifically, we implemented Algorithms 4.1, 5.1 and 6.1 using the framework of Algencan, which implies that global optimization properties mentioned in Theorems 4.2, 5.2 and 6.2 cannot be guaranteed for these implementations. However, Algencan implementations are designed in such a way that global minimizers of subproblems are actively pursued, independently of the fulfillment of approximate stationarity conditions in the subproblems. In other words, our subproblem solvers try always to find the lowest possible function values, even if this is not necessary for obtaining approximate local minimizers. As a consequence, practical behavior of Algencan-like methods is usually well explained by the properties of their global-optimization counterparts. The "preference for global minimizers" of the original smooth Algencan method has been discussed in [6].

For solving problem (5) with Algorithms 4.1, 5.1 and 6.1, the evaluations of the objective function, the constraints and their derivatives are the most time consuming tasks. In particular, among them, the most time consuming task is the evaluation of the VaR constraints $f_{i_j(x)}(x) \leq 0, j = 1, \ldots, r$ that involves, for a given x, the selection of the r smallest values among $f_i(x), i = 1, \ldots, q$. We implemented a divide-and-conquer type algorithm called Randomized-Select (see [19] pp. 185–192) whose expected complexity is linear on the number of scenarios q (for any value of r).

Codes are in Fortran 77 and, together with some problem data that makes it possible reproduction of the numerical experiments, are available for download in [39]. Codes were compiled with gfortran (GNU Fortran version 4.2.1) and the compiler option -O4 was adopted. All the experiments were run on a 2.4GHz Intel Core2 Quad Q6600 processor, 4Gb of RAM memory and Linux operating system.

7.1 Preliminary test

Before going to our main application, in this subsection we will compare Algorithms 4.1, 5.1 and 6.1 using a simple portfolio risk problem. Assume that we have a history of percentage returns for n assets and, using this information, we simulate a list of q equally probable n-uples of prices at the end of some period of time (equal to 10 business days in the experiments below). Each n-uple represents a different scenario and defines a loss function. We want to minimize the average loss subject to $f_{i_r(x)}(x) \leq 0$, where f_j is the difference between the loss under scenario j and the tolerated loss, for $j = 1, \ldots, q$. We assume that x_j is the amount invested in asset j, $j = 1, \ldots, n$. Let θ_{ij} be the quotient between the final price of asset j and the initial price of this asset, under scenario i. Therefore, at the end of the period we have $\sum_{j=1}^n \theta_{ij} x_j$ monetary units. On average, our final amount of money will be: $\sum_{j=1}^n \widehat{\theta}_j x_j$, where, for $j = 1, \ldots, n$, $\widehat{\theta}_j = \frac{1}{q} \sum_{i=1}^q \theta_{ij}$. The natural function to be minimized is, therefore,

$$-\text{Average final money} \equiv f(x) = -\sum_{j=1}^n \widehat{\theta}_j x_j.$$

We have a natural budget restriction given by:

$$\sum_{j=1}^n x_j = M \tag{41}$$

and, in order to avoid negative investments, we include the non-negativity constraints

$$x_j \geq 0, \ j = 1, \ldots, n. \tag{42}$$

The problem of minimizing $f(x)$ subject to (41) and (42) has at least one trivial solution x^*, given by

$$x_j^* = M \text{ if } \widehat{\theta}_j = \max\{\widehat{\theta}_1, \ldots, \widehat{\theta}_n\}, \ x_j^* = 0, \text{otherwise}.$$

The additional VaR constraint imposes that, under at least r scenarios, the loss must not exceed the tolerance T_{loss}. This means that:

$$\text{loss} \equiv M - \sum_{j=1}^{n} \theta_{ij} x_j \leq T_{\text{loss}} \tag{43}$$

for at least r indices i. In our experiments we take $T_{\text{loss}} = 0.05M$ and $r = 0.99q$. This means that the tolerated loss is required to be smaller than $(T_{\text{loss}}/M)100\% = 5\%$ of the invested capital with a probability $\alpha = r/q = 0.99$.

We assume that one of the assets (corresponding to $j = n$) is risk-free with rate equal to zero. This fact is expressed stating that $\theta_{in} = 1$ for all $i = 1, \dots, q$. As a consequence, the portfolio given by:

$$x_n = M, x_j = 0 \text{ for all } j = 1, \dots, n-1, \tag{44}$$

necessarily satisfies the constraints (41–43). The portfolio x given by (44) is attractive as initial approximation but it has the drawback that $f_i(x)$ is the same for all scenarios i. This makes the choice of $i_1(x), \dots, i_r(x)$ quite ambiguous. If a poor choice is made this could lead to poor local minimizers of the problem. Therefore, the initial choice

$$x_n = 0.5M, x_j = 0.5M/(n-1) \text{ for all } j = 1, \dots, n-1 \tag{45}$$

is more attractive. Observe that the portfolio (45) might not satisfy the constraint (43), but this is not a serious inconvenience for Augmented Lagrangian approaches.

Summing up, in this section we wish to solve problem (5) stated as:

$$\begin{aligned} \text{Minimize} \quad & -\sum_{j=1}^{n} \widehat{\theta}_j x_j && (46)\\ \text{subject to} \quad & f_{i_k(x)}(x) \leq 0, k = 1, \dots, r, && (47)\\ & \sum_{j=1}^{n} x_j = M, && (48)\\ & x \in \Omega, && (49) \end{aligned}$$

where $f_i(x) = M - \sum_{j=1}^{n} \theta_{ij} x_j - T_{\text{loss}}$, $i = 1, \dots, q$, and $\Omega = \{x \in I\!R^n \mid x \geq 0\}$.

We use $n = 8$ and $q = 1000$ in our experiments. For reproducibility, the scenarios matrix $\theta \in I\!R^{q \times n}$ can be found in [39]. An information that can be useful to analyze the obtained results is that, as a consequence of having

$$\widehat{\theta} \approx (0.9994, 1.0005, 0.9958, 1.0003, 1.0063, 1.0059, 1.0023, 1.0000)^T,$$

the solution of problem (46)–(49) ignoring the VaR constraint (47) is given by $(100/M) \times x_5^* = 100$ and $x_i^* = 0$, $\forall\, i \neq 5$, with $(-100/M) \times f(x^*) \approx 100.63$. Solutions obtained by Algorithms 4.1, 5.1 and 6.1 are shown in Table 1.

7.2 Model with transaction costs

We will now introduce transaction costs to problem (46)–(49). According the model described in [4, 5], impact costs can be temporary or permanent. A temporary impact is a short lived

Method	$(-100/M) \times f(x^*)$	$(100/M) \times x^*$
Algorithm 4.1	100.40	$x_1 = x_2 = x_3 = x_4 = 0$ $x_5 \approx 1.58$ $x_6 \approx 65.79$ $x_7 \approx 1.62$ $x_8 \approx 31.01$
Algorithm 5.1	100.41	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 27.54$ $x_6 \approx 39.92$ $x_8 \approx 32.54$
Algorithm 6.1	100.42	$x_1 = x_2 = x_3 = x_4 = 0$ $x_5 \approx 20.98$ $x_6 \approx 46.60$ $x_7 \approx 4.96$ $x_8 \approx 27.46$

Table 1: Performance of Algorithms 4.1, 5.1 and 6.1 applied to a simple problem without transaction costs.

deviation from the equilibrium price caused by our own trading and it can affect our transaction in progress. Its duration is mainly governed by the liquidity pattern of the stock we are trading. A permanent impact can stay well after the trade is executed but it is significantly smaller (for one order of magnitude) than the temporary impact. Both temporary and permanent impacts are concave functions. Contrary to the fixed costs, impact costs are important for large institutional investors since their trades tend to be of large volume, while their importance for small investors is not that big. Due to all these considerations we will include both fixed and impact costs in our model and consider separately small and large investors in the numerical tests. Due to the presence of impact costs, we will have a nonlinear objective function in our optimization problem. Transaction costs are also divided into fixed and impact costs in [26], where the problem is formulated as VaR minimization with yield and costs constraints.

Let us now describe the model we will consider from now on. Assume that the set of n different shares $\{1, 2, \ldots, n\}$ is available. Denoting by x_j the amount invested in asset j we have:

$$\sum_{j=1}^{n} x_j = M,\ x_j \geq 0, j = 1, \ldots, n. \tag{50}$$

Following [4, 5], the impact is a function of trading intensity that depends of several stock-specific parameters: the spread ε_j, the initial price of the stock π_j, the average daily volume ADV_j and an additional parameter $\beta \in (0, 1]$ that determines the impact function nonlinearity.

We define the temporary impact function [4], relative to asset j, in the following way:

$$H(x_j) = \frac{\varepsilon_j}{2\pi_j} + \frac{\varepsilon_j}{2\pi_j}\left(\frac{100x_j}{ADV_j\pi_j}\right)^{\beta}. \tag{51}$$

The permanent impact function [4] will be given by:

$$G(x_j) = \frac{\varepsilon_j}{\pi_j}\left(\frac{10x_j}{ADV_j\pi_j}\right)^{\beta}. \tag{52}$$

Assuming that $c_j x_j$ is the fixed cost of trading x_j monetary units of stock j, the total cost of the transaction that includes impact and fixed costs will be:

$$t_j(x_j) = x_j(H(x_j) + G(x_j) + c_j).$$

In this way, the rule of thumb given in [4] is taken into account. This rule states that a buy order of 1% of ADV_j of asset j temporarily moves the price for one spread ε_j while the same permanent impact is achieved with volume equal to 10% of ADV_j. Therefore the temporary impact is larger than the permanent for one order of magnitude.

Fixed costs, consisting of all fees and taxes, are in general proportional to the transaction value but with different factors for small and large investors. We will assume that a small investor has a fixed cost equal to 1% of the transaction value for any asset. On the other hand, a large institutional investor will be paying 0.1% for any asset.

Let us assume that the sequence of historical (or simulated) asset returns is available. This means that the matrix $\theta = (\theta_{ij})$, that gives the quotient of the final price and the initial price of the asset j under scenario i, is available. The final value of our portfolio, under scenario i, will be given by $\sum_{j=1}^{n} \theta_{ij} x_j$. Therefore, defining, as in Section 7.1, $\widehat{\theta}_j = \sum_{i=1}^{q} \theta_{ij}/q$, the average final value of the portfolio is $\sum_{j=1}^{n} \widehat{\theta}_j x_j$. We want to maximize this value, discounting transaction costs and considering the VaR constraint. This means that our objective function in (5) should be:

$$f(x) = -\sum_{j=1}^{n} \theta_j x_j + \sum_{j=1}^{n} t_j(x_j).$$

The VaR constraint is determined by the definition of $f_i(x)$ for each scenario i. In the present case, we have:

$$f_i(x) = M - \sum_{j=1}^{n} \theta_{ij} x_j + \sum_{j=1}^{n} t_j(x_j) - T_{\text{loss}},$$

where T_{loss} denotes the tolerated loss.

As in Section 7.1, we take $T_{\text{loss}} = 0.05M$, $r = 0.99q$ and $q = 1000$ in our experiments. Moreover, we also assume that one of the assets (corresponding to $j = n$) is risk-free stating that $\theta_{in} = 1$ for all $i = 1, \ldots, q$, $\varepsilon_n = 0$ and $c_n = 0$. Finally, we also use $x_n = 0.5M$, $x_j = 0.5M/(n-1)$ for $j = 1, \ldots, n-1$ as initial choice.

Summing up, in this section we wish to solve problem (5) stated as:

$$\begin{aligned}
\text{Minimize} \quad & -\sum_{j=1}^{n} \widehat{\theta}_j x_j + \sum_{j=1}^{n} t_j(x_j) && (53)\\
\text{subject to} \quad & f_{i_k(x)} \le 0, k = 1, \ldots, r, && (54)\\
& \sum_{j=1}^{n} x_j = M, && (55)\\
& x \in \Omega, && (56)
\end{aligned}$$

where

$$f_i(x) = M - \sum_{j=1}^{n} \theta_{ij} x_j + \sum_{j=1}^{n} t_j(x_j) - T_{\text{loss}}, i = 1, \ldots, q,$$

and

$$\Omega = \{x \in I\!R^n \mid x \geq 0\}.$$

The proposed algorithms were tested using the problems described below. The set of $n = 8$ available shares consists of the seven FTSE shares AZN.L, BARC.L, KGF.L, LLOY.L, MKS.L, TSCO.L and VOD.L, plus a risk-free asset. We consider $q = 1000$ scenarios. The matrices of scenarios θ were generated using the historical data on daily returns from March 19 (2004) to February 12 (2008). For the small investor case we consider $M = 10,000$ (pounds) and portfolios with a time life of 120 and 240 business days. For the large investor case we consider $M = 100,000,000$ and a 10-days life time portfolio. The matrices θ as well as the transaction costs related constants ε_j, π_j and $ADV_j, j = 1, \ldots, n$ can be found in [39]. Table 2 shows the solutions found by Algorithms 4.1, 5.1 and 6.1 in the large investor case using $\beta = 1$ and $\beta = 0.6$ in the transaction costs formulae, respectively. The solutions in the small investor case for $\beta = 1$ are given in Table 3. As expected, since impact factors are negligible for small investors, tests made with $\beta = 0.6$ yield almost identical results.

In Table 2 we observe that:

1. As expected, returns with $\beta = 0.6$ are slightly smaller than returns with $\beta = 1$ since in the first case the impact factor is bigger.

2. Qualitatively, the suggested portfolios are the same in all the cases except in the case $\beta = 0.6$ with Algorithm 5.1. Since the return is slightly smaller in this case, it may be deduced that Algorithm 5.1 converged to a local minimizer when $\beta = 0.6$.

3. The identity between portfolios with $\beta = 1$ and $\beta = 0.6$ is more evident in the case of Algorithm 6.1. However, it can be observed that the solution obtained by Algorithm 6.1 is also identical to the one obtained by Algorithm 4.1 with $\beta = 0.6$. On the other hand, the solution obtained by Algorithms 4.1 and 5.1 are identical when $\beta = 0.6$. This seems to indicate that the algorithms find slightly different local minimizers, the choice among which is not meaningful from the practical point of view.

The conclusions in the case of the small investor with short and long life time scenarios (Table 3) are significantly different. (Recall that we only report the case $\beta = 1$ because the results for $\beta = 0.6$ are almost identical.)

1. The three algorithms recommend, for the 120 days scenario, to keep 74 % of the budget on the risk-free asset (x_8), around 19 % on the asset x_6 and 6 % on x_5. In the case of the 240-days scenario the decision is even more conservative, since the investment on asset x_5 migrates to the risk-free asset.

2. The reason for that progressively conservative behavior is the VaR constraint. Clearly, the chance of loosing 5 % of the capital is very small for a short-time investment and increases with time. Although short-time investments are not encouraged for small investors, the conservative effect of transaction costs is less important than effect caused by the VaR constraint.

Method		$(-100/M) \times f(x^*)$	$(100/M) \times x^*$
$\beta = 1.0$	Algorithm 4.1	100.31	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 16.20$ $x_6 \approx 53.03$ $x_8 \approx 30.77$
	Algorithm 5.1	100.31	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 16.20$ $x_6 \approx 53.03$ $x_8 \approx 30.77$
	Algorithm 6.1	100.31	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 10.17$ $x_6 \approx 58.82$ $x_8 \approx 31.01$
$\beta = 0.6$	Algorithm 4.1	100.30	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 10.15$ $x_6 \approx 58.71$ $x_8 \approx 31.13$
	Algorithm 5.1	100.29	$x_1 = x_2 = x_3 = x_4 = 0$ $x_5 \approx 1.55$ $x_6 \approx 64.36$ $x_7 \approx 1.58$ $x_8 \approx 32.51$
	Algorithm 6.1	100.30	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 10.15$ $x_6 \approx 58.71$ $x_8 \approx 31.13$

Table 2: Performance of Algorithms 4.1, 5.1 and 6.1 using $\beta \in \{0.6, 1.0\}$ in the large investor case considering a 10-days life time scenario.

3. There are no important practical differences between the tested algorithms. The average returns predicted by them are very similar.

7.3 Large-scale problems

In this subsection we will consider larger scale instances of problem (53)-(56) with $n \in \{500, 1000\}$ assets and $q = 1000$ scenarios. The data $\theta \in I\!R^{q \times n}$ and $\varepsilon, \pi, ADV \in I\!R^n$ can be found in [39]. We use $\beta = 0.6$ to deal with non-linear transaction costs. We deal with the large investor case, using $M = 100,000,000$, $c_j = 0.001$ for all j. The matrix θ simulates 120-days life time scenarios. We also consider that the n-th asset is risk-free and we use $x_n = 0.5M$, $x_j = 0.5M/(n-1)$ for $j = 1, \ldots, n-1$ as initial choice.

In order to evaluate the influence of the VaR constraint in the quality of the objective function value at the solution, we solve the problem for different values of the VaR constraint parameters

$$\text{Confidence level } = r/q \in \{0.90, 0.91, \ldots, 0.99\}$$

Method		$(-100/M) \times f(x^*)$	$(100/M) \times x^*$
120 days	Algorithm 4.1	101.77	$x_2 = x_3 = x_4 = 0$ $x_1 \approx 1.02$ $x_5 \approx 6.04$ $x_6 \approx 18.53$ $x_7 \approx 1.31$ $x_8 \approx 73.11$
	Algorithm 5.1	101.82	$x_2 = x_3 = x_4 = x_7 = 0$ $x_1 \approx 0.13$ $x_5 \approx 6.19$ $x_6 \approx 19.37$ $x_8 \approx 74.31$
	Algorithm 6.1	101.83	$x_2 = x_3 = x_4 = x_7 = 0$ $x_1 \approx 0.19$ $x_5 \approx 6.17$ $x_6 \approx 19.41$ $x_8 \approx 74.23$
240 days	Algorithm 4.1	103.00	$x_1 = x_2 = x_3 = x_4 = x_7 = 0$ $x_5 \approx 2.63$ $x_6 \approx 17.23$ $x_8 \approx 80.14$
	Algorithm 5.1	103.09	$x_3 = x_4 = x_5 = x_7 = 0$ $x_1 \approx 0.15$ $x_2 \approx 0.54$ $x_6 \approx 19.49$ $x_8 \approx 79.82$
	Algorithm 6.1	103.09	$x_3 = x_4 = x_5 = x_7 = 0$ $x_1 \approx 0.15$ $x_2 \approx 0.55$ $x_6 \approx 19.49$ $x_8 \approx 79.81$

Table 3: Performance of Algorithms 4.1, 5.1 and 6.1 using $\beta = 1.0$ in the small investor case considering 120- and 240-days life time scenarios.

and

$$\text{Maximal percentual loss } = (100/M) \times T_{\text{loss}} \in \{1, 2, \ldots, 10\}.$$

Tables 4 and 5 show the optimal values obtained by Algorithms 4.1, 5.1 and

6.1 for each combination of r and T_{loss}. Note that tighter the VaR constraint (bottom-right), smaller the expected gain.

The average CPU times of each method are 64.82, 60.10 and 10.06 seconds, for the problem with $n = 500$, and 324.65, 451.02 and 50.27 seconds, for the problem with $n = 1000$, respectively. On average, optimal values found by Algorithm 4.1 are slightly better than the ones obtained by Algorithm 5.1, and the ones obtained by Algorithm 5.1 are slightly better than the ones obtained by Algorithm 6.1.

It may be observed that, in the solutions given by Algorithms 4.1 and 5.1, the expected return decreases when the confidence level increases and when the tolerated loss decreases, as expected. In the case of Algorithm 6.1 this desired behavior is violated systematically when we go from $\alpha = 0.95$ to $\alpha = 0.96$. In fact, we observe that the expected return increases between those values of α. This may reveal that, as predicted by theory, the global convergence properties of Algorithms 4.1 and 5.1 are stronger than the ones of Algorithm 6.1.

8 Conclusions

We presented algorithms of Augmented Lagrangian type to cope the (large scale) portfolio optimization problem with transaction costs. The structure of this optimization problem is different from the standard smooth nonlinear programming structure. Non-smoothness of our problem appears in a very particular way, allowing us to fully exploit the main characteristics of the Low Order-Value constrained optimization paradigm.

Numerical results seem to show that there are little performance differences between the algorithms presented in this paper. However, these results are preliminary and practical experience with this approach is still incipient. Theoretical results are rather emphatic in the sense of showing that better behavior should be expected from algorithms in Section 4. As mentioned before, the algorithms effectively implemented here do not guarantee global minimization of the problems. The complexity of the problem makes it very improbable that practical guaranteedly global methods could exist, at least with the present development of computer facilities and in the presence of large-scale situations. However, in everyday practice, is is not unreliable to think that good initial approximations to the global minimizers, provided by trained investors, may be available. Moreover, when Algencan-like algorithms are used, the global heuristic procedures merged in the subproblems are known to enhance the probability of convergence to global minimizers.

Other times, investors are ready to make small changes in their portfolios instead of the large ones that could be suggested by a global minimization solver. Real life contains subjective criteria that are difficult to capture in models. Such criteria are usually implicit in initial approximations. Thus, on one hand, the initial approximation motivated by investor feelings is, probably, a good guess for the global solution of the problem. On the other hand, in the case that global minimizers are very far from initial educated guesses, it is very likely that some constraint implicitly considered by the user is not contemplated in the mathematical model. For this reason we believe that even local-minimization algorithms may be quite useful.

	Algorithm 4.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.51	101.48	101.42	101.31	101.25	101.16	101.10	101.06	100.99	100.87	**101.22**
9 %	101.37	101.33	101.27	101.18	101.11	101.07	101.00	100.96	100.86	100.79	**101.09**
8 %	101.21	101.18	101.14	101.05	101.00	100.92	100.87	100.84	100.80	100.70	**100.97**
7 %	101.06	101.02	100.94	100.92	100.88	100.81	100.76	100.75	100.67	100.63	**100.84**
6 %	100.92	100.72	100.85	100.77	100.73	100.69	100.66	100.65	100.58	100.52	**100.71**
5 %	100.77	100.74	100.70	100.66	100.63	100.59	100.56	100.54	100.50	100.45	**100.61**
4 %	100.61	100.59	100.56	100.53	100.50	100.48	100.46	100.43	100.41	100.35	**100.49**
3 %	100.46	100.45	100.42	100.40	100.36	100.36	100.34	100.33	100.29	100.27	**100.37**
2 %	100.31	100.30	100.29	100.27	100.25	100.25	100.23	100.22	100.20	100.18	**100.25**
1 %	100.16	100.16	100.15	100.14	100.13	100.13	100.12	100.11	100.11	100.10	**100.13**
$\bar{f}$	**100.84**	**100.80**	**100.77**	**100.72**	**100.68**	**100.65**	**100.61**	**100.59**	**100.54**	**100.49**	**100.67**

	Algorithm 5.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.51	101.48	101.42	101.31	101.25	101.19	101.11	101.06	100.98	100.89	**101.22**
9 %	101.37	101.33	101.27	101.18	101.11	101.08	101.00	100.96	100.84	100.80	**101.09**
8 %	101.22	101.18	101.12	100.91	100.97	100.87	100.89	100.82	100.77	100.70	**100.95**
7 %	101.03	101.04	100.99	100.90	100.83	100.78	100.78	100.73	100.69	100.59	**100.84**
6 %	100.91	100.89	100.85	100.76	100.74	100.69	100.67	100.62	100.58	100.54	**100.73**
5 %	100.77	100.72	100.71	100.64	100.61	100.60	100.55	100.53	100.50	100.45	**100.61**
4 %	100.55	100.55	100.51	100.51	100.49	100.49	100.46	100.42	100.39	100.36	**100.47**
3 %	100.44	100.43	100.40	100.38	100.38	100.35	100.34	100.32	100.30	100.27	**100.36**
2 %	100.31	100.31	100.28	100.26	100.25	100.23	100.23	100.22	100.21	100.18	**100.25**
1 %	100.15	100.14	100.14	100.13	100.13	100.12	100.11	100.10	100.10	100.10	**100.12**
$\bar{f}$	**100.83**	**100.81**	**100.77**	**100.70**	**100.68**	**100.64**	**100.61**	**100.58**	**100.54**	**100.49**	**100.66**

	Algorithm 6.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.40	101.33	101.13	101.12	101.09	101.01	101.09	101.03	100.93	100.82	**101.10**
9 %	101.26	101.20	101.02	100.99	100.95	100.91	100.98	100.93	100.84	100.74	**100.98**
8 %	101.12	101.07	100.91	100.87	100.88	100.81	100.87	100.83	100.75	100.66	**100.88**
7 %	100.98	100.93	100.80	100.77	100.77	100.71	100.76	100.72	100.65	100.58	**100.77**
6 %	100.84	100.80	100.68	100.66	100.66	100.61	100.65	100.62	100.56	100.49	**100.66**
5 %	100.70	100.67	100.57	100.55	100.55	100.51	100.55	100.52	100.47	100.41	**100.55**
4 %	100.56	100.54	100.46	100.44	100.44	100.41	100.44	100.42	100.37	100.33	**100.44**
3 %	100.42	100.40	100.34	100.33	100.33	100.30	100.33	100.31	100.28	100.25	**100.33**
2 %	100.28	100.27	100.23	100.22	100.22	100.20	100.22	100.21	100.20	100.17	**100.22**
1 %	100.14	100.14	100.12	100.11	100.11	100.10	100.11	100.11	100.10	100.08	**100.11**
$\bar{f}$	**100.77**	**100.74**	**100.63**	**100.61**	**100.60**	**100.56**	**100.60**	**100.57**	**100.52**	**100.45**	**100.60**

Table 4: Performance of Algorithms 4.1, 5.1 and 6.1 on a problem with $n = 500$ assets, varying confidence level and tolerated loss. The VaR constraint imposes that the loss must be smaller than $(100/M) \times T_{\text{loss}}\%$ of the invested capital with a probability $\alpha = r/q$. Each cell of the table shows the value of $-(100/M) \times f(x^*)$. Rows and columns in boldface represent average values.

Among the algorithmic improvements that we have in mind, we plan to adapt some of the global techniques employed in the subproblems of [17] to the problem introduced in this paper.

Acknowledgements. We are indebted to two anonymous referees for useful remarks.

	Algorithm 4.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.78	101.75	101.72	101.63	101.54	101.41	101.28	101.17	101.12	101.04	**101.44**
9 %	101.63	101.59	101.51	101.47	101.39	101.27	101.16	101.06	101.02	100.94	**101.30**
8 %	101.45	101.39	101.38	101.31	101.24	101.11	100.98	100.94	100.92	100.81	**101.15**
7 %	101.26	101.24	101.19	101.15	101.08	100.97	100.86	100.83	100.81	100.73	**101.01**
6 %	101.09	101.02	101.01	100.97	100.93	100.86	100.78	100.71	100.69	100.63	**100.87**
5 %	100.86	100.83	100.88	100.83	100.78	100.71	100.64	100.60	100.58	100.52	**100.72**
4 %	100.73	100.69	100.65	100.67	100.63	100.57	100.51	100.49	100.46	100.41	**100.58**
3 %	100.56	100.53	100.53	100.49	100.47	100.43	100.39	100.35	100.35	100.31	**100.44**
2 %	100.37	100.36	100.33	100.33	100.32	100.28	100.27	100.24	100.24	100.21	**100.30**
1 %	100.19	100.19	100.18	100.17	100.16	100.14	100.14	100.12	100.12	100.10	**100.15**
$\bar{f}$	**100.99**	**100.96**	**100.94**	**100.90**	**100.85**	**100.78**	**100.70**	**100.65**	**100.63**	**100.57**	**100.80**

	Algorithm 5.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.80	101.74	101.72	101.63	101.55	101.41	101.29	101.19	101.08	101.04	**101.45**
9 %	101.66	101.59	101.53	101.47	101.35	101.27	101.11	101.07	101.02	100.93	**101.30**
8 %	101.45	101.40	101.33	101.30	101.24	101.11	100.97	100.94	100.89	100.84	**101.15**
7 %	101.28	101.26	101.14	101.14	101.07	100.99	100.85	100.82	100.77	100.72	**101.00**
6 %	101.09	101.02	100.98	100.98	100.88	100.85	100.73	100.71	100.69	100.63	**100.86**
5 %	100.90	100.87	100.85	100.82	100.76	100.71	100.60	100.59	100.57	100.51	**100.72**
4 %	100.74	100.67	100.67	100.58	100.63	100.57	100.53	100.46	100.46	100.41	**100.57**
3 %	100.55	100.53	100.49	100.46	100.45	100.43	100.40	100.36	100.34	100.31	**100.43**
2 %	100.36	100.35	100.35	100.33	100.31	100.30	100.26	100.24	100.23	100.20	**100.29**
1 %	100.20	100.18	100.18	100.17	100.16	100.15	100.13	100.12	100.12	100.10	**100.15**
$\bar{f}$	**101.00**	**100.96**	**100.92**	**100.89**	**100.84**	**100.78**	**100.69**	**100.65**	**100.62**	**100.57**	**100.79**

	Algorithm 6.1										
	Confidence level $\alpha = r/q$										$\bar{f}$
% loss $(100/M) \times T_{\text{loss}}$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
10%	101.68	101.64	101.46	101.52	101.31	101.26	101.16	101.08	101.04	100.99	**101.31**
9 %	101.64	101.41	101.31	101.31	101.27	101.14	101.05	100.97	100.93	100.89	**101.19**
8 %	101.35	101.26	101.17	101.16	101.13	101.01	100.93	100.86	100.82	100.80	**101.05**
7 %	101.28	101.10	101.02	101.01	100.92	100.89	100.82	100.75	100.73	100.70	**100.92**
6 %	101.08	100.95	100.87	100.87	100.79	100.76	100.70	100.65	100.65	100.60	**100.79**
5 %	100.91	100.79	100.73	100.73	100.66	100.64	100.58	100.54	100.53	100.50	**100.66**
4 %	100.74	100.66	100.58	100.58	100.53	100.51	100.47	100.43	100.42	100.40	**100.53**
3 %	100.55	100.47	100.44	100.44	100.40	100.38	100.35	100.35	100.32	100.31	**100.40**
2 %	100.37	100.36	100.29	100.29	100.27	100.25	100.23	100.24	100.21	100.20	**100.27**
1 %	100.18	100.18	100.15	100.15	100.13	100.13	100.12	100.11	100.11	100.10	**100.14**
$\bar{f}$	**100.98**	**100.88**	**100.80**	**100.81**	**100.74**	**100.70**	**100.64**	**100.60**	**100.58**	**100.55**	**100.73**

Table 5: Performance of Algorithms 4.1, 5.1 and 6.1 on a problem with $n = 1000$ assets, varying confidence level and tolerated loss. The VaR constraint imposes that the loss must be smaller than $(100/M) \times T_{\text{loss}}$% of the invested capital with a probability $\alpha = r/q$. Each cell of the table shows the value of $-(100/M) \times f(x^*)$. Rows and columns in boldface represent average values.

References

[1] C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas, A global optimization method α-BB for process design, *Computers & Chemical Engineering* 20, pp. S419–424, 1996.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, A global optimization method, α-BB, for general twice-differentiable constrained NLPs – I. Theoretical Advances, *Computers & Chemical Engineering* 22, pp. 1137–1158, 1998.

[3] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, A global optimization method, α-BB, for general twice-differentiable constrained NLPs – II. Implementation and computational results, *Computers & Chemical Engineering* 22, pp. 1159–1179, 1998.

[4] R. F. Almgren, Optimal execution with nonlinear impact functions and trading-enhanced risk, *Applied Mathematical Finance* 10, pp. 1–18, 2003

[5] R. F. Almgren and N. Chriss, Optimal execution of portfolio transaction, *Journal of Risk* 3, pp. 5–39, 2001.

[6] 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.

[7] 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.

[8] 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.

[9] R. Andreani, G. Haeser, and J. M. Martínez, Sequential optimality conditions for constrained optimization, *Optimization*, to appear.

[10] R. Andreani, J. M. Martínez, L. Martínez, and F. Yano, Continuous optimization methods for structure alignments, *Mathematical Programming* 112, pp. 93–124, 2008.

[11] R. Andreani, J. M. Martínez, L. Martínez, and F. Yano, Low Order-value optimization and applications, *Journal of Global Optimization* 43, pp. 1–10, 2009.

[12] R. Andreani, J. M. Martínez, M. Salvatierra, and F. Yano, Quasi-Newton methods for Order-value optimization and value-at-risk calculations, *Pacific Journal of Optimization* 2, pp. 11–33, 2006.

[13] 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.

[14] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, α-BB: A global optimization method for general constrained nonconvex problems, *Journal of Global Optimization* 7, pp. 337–363, 1995.

[15] A. Ben Tal and M. Teboulle, And old-new concept of convex risk measures: the optimized certainty equivalent, *Mathematical Finance* 17, pp. 449–476, 2007.

[16] M. J. Best and J. Hlouskova, Portfolio selection and transactions costs, *Computational Optimization and Applications* 24, pp. 95–116, 2003.

[17] 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.

[18] J. P. Bouchaud, Y. Gefen, M. Potters, and M. Wyart, Fluctuations and response in financial markets: The subtle nature of 'random' price changes, *Quantitative Finance* 4, pp. 57–62, 2004.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition, The MIT Press, 2001.

[20] J. Daníielsson, B. N. Jorgensen, C. G. de Vries, and X. Yang, Optimal portfolio allocation under the probabilistic VaR constraint and incentives for financial innovation, *Annals of Finance* 4, pp. 345–367, 2008.

[21] A. A. Gaivoronski and G. Pflug, Finding optimal portfolios with constraints on value at risk, *Technical Report*, Department of Industrial Economics and Technology Management, NTNU - Norwegian University of Science and Technology, Norway, 1999.

[22] A. A. Gaivoronski and G. Pflug, Value at risk in portfolio Optimization: Properties and computational approach, *Journal of Risk* 7, pp. 1–31, 2005.

[23] M. R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, pp. 303–320, 1969.

[24] R. Horst, P. M. Pardalos, and M. V. Thoai, *Introduction to Global Optimization*, Kluwer Book Series: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[25] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*, McGraw-Hill, 2001.

[26] N. Krejić, M. Kumaresan, and A. Rožnjik, VaR optimal portfolio with transaction costs, University of Novi Sad, Department of Mathematics and Informatics, 2008.

[27] F. Lillo, J. D. Farmer, and R. N. Mantegna, Master curve for price impact function, *Nature* 421, pp. 129–130, 2003.

[28] M. S. Lobo, M. Fazel, and S. Boyd, Portfolio optimization with linear and fixed transaction costs, *Annals of Operation Research* 152, pp. 341–365, 2007.

[29] H. Markowitz, Portfolio Selection, *The Journal of Finance* 7, pp. 77–91, 1952.

[30] R. Mansini and G. Speranza, An exact approach for portfolio selection with transaction costs and rounds, *IIE Transactions* 37, pp. 919–929, 2005.

[31] 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.

[32] M. Potaptchik, L. Tuncel, and H. Wolkowicz, Large scale portfolio Optimization with piecewise linear transaction costs, *Optimization Methods & Software* 23, pp. 929-952, 2008.

[33] M. J. D. Powell, A method for nonlinear constraints in minimization problems, in *Optimization*, R. Fletcher (ed.), Academic Press, New York, NY, pp. 283–298, 1969.

[34] L. Qi and Z. Wei, On the constant positive linear dependence condition and its application to SQP methods, *SIAM Journal on Optimization* 10, pp. 963–981, 2000.

[35] R. T. Rockafellar, Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control and Optimization* 12, pp. 268–285, 1974.

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

[37] A. M. So, J. Zhang, and Y. Ye, Stochastic Combinatorial Optimization with Controllable Risk Aversion Level, *Mathematics of Operations Research* 34, pp. 522-537, 2009.

[38] http://www.ime.usp.br/∼egbirgin/tango/

[39] http://www.ime.usp.br/∼egbirgin/