J Glob Optim (2009) 43:1–22
DOI 10.1007/s10898-008-9280-3

Low Order-Value Optimization and applications

R. Andreani · J. M. Martínez · L. Martínez · F. S. Yano

Received: 10 December 2007 / Accepted: 28 January 2008 / Published online: 12 February 2008
© Springer Science+Business Media, LLC. 2008

Abstract Given r real functions $F_1(x), \dots, F_r(x)$ and an integer p between 1 and r, the Low Order-Value Optimization problem (LOVO) consists of minimizing the sum of the functions that take the p smaller values. If $(y_1, \dots, y_r)$ is a vector of data and $T(x, t_i)$ is the predicted value of the observation i with the parameters $x \in I\!R^n$, it is natural to define $F_i(x) = (T(x, t_i) - y_i)^2$ (the quadratic error in observation i under the parameters x). When $p = r$ this LOVO problem coincides with the classical nonlinear least-squares problem. However, the interesting situation is when p is smaller than r. In that case, the solution of LOVO allows one to discard the influence of an estimated number of outliers. Thus, the LOVO problem is an interesting tool for robust estimation of parameters of nonlinear models. When $p \ll r$ the LOVO problem may be used to find hidden structures in data sets. One of the most successful applications includes the Protein Alignment problem. Fully documented algorithms for this application are available at www.ime.unicamp.br/~martinez/lovoalign. In this paper optimality conditions are discussed, algorithms for solving the LOVO problem are introduced and convergence theorems are proved. Finally, numerical experiments are presented.

R. Andreani · J. M. Martínez (✉) · F. S. Yano
Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas,
CP 6065, 13081-970 Campinas, SP, Brazil
e-mail: martinez@ime.unicamp.br

R. Andreani
e-mail: andreani@ime.unicamp.br

L. Martínez
Institute of Chemistry, State University of Campinas, Campinas, Brazil
e-mail: lmartinez@iqm.unicamp.br

L. Martínez
Institute Pasteur, Paris, France
e-mail: leandro@pasteur.fr

F. S. Yano
Itaú Bank, São Paulo, Brazil
e-mail: flavio.yano@itau.com.br

Keywords Order-Value Optimization · Algorithms · Convergence · Robust estimation of parameters · Hidden patterns

1 Introduction

Given r functions $F_1, \dots, F_r$ defined in a domain $\Omega \subset I\!R^n$ and an integer $p \in \{1, \dots, r\}$, we define the Low Order-Value function $S_p : \Omega \to I\!R$ by

$$S_p(x) = \sum_{j=1}^{p} F_{i_j(x)}(x)$$

for all $x \in \Omega$, where $\{i_1(x), \dots, i_r(x)\} = \{1, \dots, r\}$ and

$$F_{i_1(x)}(x) \le F_{i_2(x)}(x) \le \cdots \le F_{i_p(x)}(x) \le \cdots \le F_{i_r(x)}(x).$$

If the functions F_i are continuous, the function S_p is continuous as well, since it is a sum of continuous functions [1,2]. However, even if all the functions F_i are differentiable, the function S_p is, generally, nonsmooth. We define the Low Order-Value Optimization (LOVO) problem in the following way:

$$\text{Minimize } S_p(x) \ \text{ subject to } x \in \Omega. \tag{1}$$

In [1], the Order-Value Optimization problem (OVO) was introduced as the minimization of the Order-Value function $F_{i_p(x)}(x)$ subject to $x \in \Omega$. In [2], a nonlinear programming reformulation was given for OVO and it was proved that, without the necessity of constraint qualifications, local solutions of the reformulation are KKT points. The main applications of OVO are in risk evaluation and robust estimation [18]. When $F_i(x)$ represents the predicted loss under the scenario i and the decision x, the OVO function $F_{i_p(x)}$ corresponds, essentially, to the classical Value-at-Risk (VaR) [19] measurement with a confidence level (p/r) [6]. The Conditional Value-at-Risk (CVaR) measurement with confidence level $(r-p)/r$ corresponds to the High Order-Value function $S^p(x) = \sum_{j=r-p+1}^{r} F_{i_j(x)}(x)$. In this case p is generally small. A more general OVO formulation was presented in [25].

Let us define $m = r!/[p!(r-p)!]$. Clearly, the set $\{1, \dots, r\}$ contains exactly m different subsets $\mathcal{C}_1, \dots, \mathcal{C}_m$ with cardinality p. For all $i = 1, \dots, m$, $x \in \Omega$, we define:

$$f_i(x) = \sum_{j \in \mathcal{C}_i} F_j(x) \tag{2}$$

and

$$f_{\min}(x) = \min\{f_1(x), \dots, f_m(x)\}.$$

It is easy to see that $f_{\min}(x) = S_p(x)$ for all $x \in \Omega$ and, thus, the LOVO problem is:

$$\text{Minimize } f_{\min}(x) \ \text{ subject to } x \in \Omega. \tag{3}$$

Of course, the problem (3) is, at the same time, a particular case of (1), which corresponds to take $p = 1$ and $F_i(x) = f_i(x)$, $i = 1, \dots, r$.

The characterization (3) of the LOVO problem will be used throughout this paper for theoretical purposes and for some relevant applications. However, it should be observed that, for computing $f_{\min}$ in the case (2) it is not necessary to compute all the functions f_i.

The High Order-Value function (that corresponds to CVaR) is $S^p(x) = f_{\max}(x)$, where $f_{\max}(x) = \max\{f_1(x), \dots, f_m(x)\}$. So, if the functions f_i are convex, the problem (HOVO) of minimizing CVaR is a convex (minimax) problem and, if the f_i's are affine functions, this problem is reduced to Linear Programming [32].

The OVO problem [minimizing $F_{i_p(x)}(x)$] may be applied to robust estimation of parameters because it generalizes the classical Minimax regression which, as it is well known, is very sensitive to the presence of outliers. However, LOVO is more adequate for robust estimation purposes, with the proper definitions of $F_i(x)$. If $y_1, \dots, y_r \in I\!R$ are observations of a given phenomenon which, theoretically, corresponds to the physical law $y = T(x, t)$, we may define $F_i(x)$ as the quadratic error at the ith observation ($F_i(x) = (T(x, t_i) - y_i)^2$). The least-squares estimation of the parameters x comes from solving

$$\text{Minimize } \sum_{i=1}^{r} F_i(x) \text{ subject to } x \in \Omega.$$

If we estimate that approximately $r - p$ observations come from (probably systematic) observation errors, it is natural to estimate the parameters by means of solving the LOVO problem

$$\text{Minimize } S_p(x) \text{ subject to } x \in \Omega.$$

Therefore, this LOVO problem is a generalization of the nonlinear least-squares problem which is able to eliminate the influence of outliers.

Unlike OVO and HOVO, the LOVO problem is not applicable to risk evaluation. The reason is that, if we define $F_i(x)$ as the predicted loss under the decision x, the LOVO function discards the larger losses (as OVO and VaR) but does not discard the smaller ones. So, the decisions under LOVO would be always unreasonably optimistic and risky.

On the other hand, in the case that $p \ll r$, the LOVO problem is a tool for finding Hidden Patterns in situations where a lot of wrong observations are mixed with a small number of correct data [5].

LOVO is a *global* nonsmooth optimization problem. In this paper, it will be shown that, in spite of nonsmoothness, essentially smooth methods may be applied to its resolution preserving convergence to reasonable stationary points. The main difficulty for solving LOVO is the fact that the problem possesses many local minimizers. Therefore, in most applications, global optimization tools are necessary in order to obtain suitable initial points. In [5] a space-filling curve method was suggested for obtaining initial approximations when the original OVO is applied to Hidden Pattern problems. Specific problems need particular heuristics for finding initial points. A particular heuristic for Protein Alignment is described in [26].

This paper is organized as follows. In Sect. 2 we define two types of optimality conditions for the LOVO problem. In Sect. 3 we define an algorithm for unconstrained LOVO problems, that converges to weakly critical points. In Sect. 4 we introduce a method that converges to strongly critical points. In Sect. 5 we introduce an algorithm for constrained LOVO problems. Hidden Pattern and Protein Alignment problems are discussed in Sect. 6. Numerical examples are given in Sect. 7 and conclusions in Sect. 8.

Notation.

- The symbol $\|\cdot\|$ will denote the Euclidean norm of vectors and matrices, although often it may be replaced by an arbitrary norm.
- $B(x_*, \epsilon) = \{x \in I\!R^n \mid \|x - x_*\| \le \epsilon\}$.
- We denote $I\!N = \{0, 1, 2, \dots\}$.
- We denote $I\!R_+ = \{t \in I\!R \mid t \ge 0\}$ and $I\!R_{++} = \{t \in I\!R \mid t > 0\}$.

- Given $K = \{k_0, k_1, k_2, \ldots\}$ such that $k_j < k_{j+1}$ and $k_j \in I\!N$ for all $j \in I\!N$, we denote
$$\lim_{k \in K} z_k = \lim_{j \to \infty} z_{k_j}.$$
- If $B \in I\!R^{n \times n}$, $B > 0$ means that B is positive definite.
- $[v]_i$ denotes the ith component of the vector i. If there is no place for confusion, we also denote $v_i = [v]_i$.
- If $v \in I\!R^n$, we denote $v_+ = (\max\{0, v_1\}, \ldots, \max\{0, v_n\})^T$.

2 Optimality conditions

In this section we use formulation (3). For all $x \in \Omega$ we define
$$I_{\min}(x) = \{i \in \{1, \ldots, m\} \mid f_i(x) = f_{\min}(x)\}.$$

In Lemma 2.1, we prove that a global minimizer x_* of (3) is, necessarily, a global minimizer of $f_i(x)$ for all $i \in I_{\min}(x_*)$. In Theorem 2.1 we show that the same property holds for local minimizers.

Lemma 2.1 *Let $A \subset \Omega$, $x_* \in A$. If the point x_* is a global minimizer of $f_{\min}(x)$ subject to $x \in A$, then x_* is a global minimizer of $f_i(x)$ subject to $x \in A$ for all $i \in I_{\min}(x_*)$. In particular (taking $A = \Omega$), if x_* is a global minimizer of* (3) *then x_* is a global minimizer of $f_i(x)$ for all $i \in I_{\min}(x_*)$*

Proof Assume that, for some $i \in I_{\min}(x_*)$, x_* is not a global minimizer of $f_i(x)$ subject to $x \in A$. Then, there exists $y \in A$ such that $f_i(y) < f_i(x_*)$. So, by the definitions of $f_{\min}$ and $I_{\min}(x_*)$,
$$f_{\min}(y) \le f_i(y) < f_i(x_*) = f_{\min}(x_*).$$
Therefore, x_* is not a global minimizer of $f_{\min}(x)$ subject to $x \in A$. □

Theorem 2.1 *If $x_* \in \Omega$ is a local minimizer of* (3) *then, for all $i \in I_{\min}(x_*)$, x_* is a local minimizer of $f_i(x)$ subject to $x \in \Omega$.*

Proof Let $\epsilon > 0$ such that x_* is a global minimizer of $f_{\min}(x)$ subject to $x \in A$, where
$$A = \{x \in \Omega \mid \|x - x_*\| \le \epsilon\}.$$
By Lemma 1 we obtain that x_* is a global minimizer of $f_i(x)$ subject to $x \in A$ for all $i \in I_{\min}(x_*)$. Therefore, x_* is local minimizer of $f_i(x)$ subject to $x \in \Omega$ for all $i \in I_{\min}(x_*)$. □

Remark The reciprocal of Lemma 2.1 is not true, even if the functions are continuous. Take $A = \Omega = I\!R$, $f_1(x) = (x-1)^2$, $f_2(x) = x$. Although $x_* = 1$ is a global minimizer of $f_i(x)$ for all $i \in I_{\min}(x_*) = \{1\}$, this point is not a global minimizer of $f_{\min}$. However, as we will see below, the reciprocal of Theorem 2.1 is true if the functions f_i are continuous.

Proposition 2.1 *Assume that x_* is a local minimizer of f_i for all $i \in I_{min}(x_*)$ and that f_i is continuous at x_* for all $i \notin I_{\min}(x_*)$. Then x_* is a local minimizer of* (3).

Proof Let $\epsilon > 0$ be such that $f_i(x_*) > f_{\min}(x_*) + \epsilon$ for all $i \notin I_{\min}(x_*)$. Since f_i is continuous for all $i \notin I_{\min}(x_*)$, there exists $\delta_1 > 0$ such that

$$f_i(x) > f_{\min}(x_*) \text{ for all } i \notin I_{\min}(x_*) \tag{4}$$

whenever $\|x - x_*\| \le \delta_1$.

By the hypothesis, there exists $\delta_2 > 0$ such that for all $i \in I_{\min}(x_*)$,

$$f_i(x) \ge f_i(x_*) = f_{\min}(x_*) \tag{5}$$

whenever $\|x - x_*\| \le \delta_2$.

Define $\delta = \min\{\delta_1, \delta_2\}$. By (4) and (5), we have that, for all $x \in \Omega$ such that $\|x - x_*\| \le \delta$, and for all $i = 1, \ldots, m$,

$$f_i(x) \ge f_{\min}(x_*).$$

Therefore,

$$f_{\min}(x) \ge f_{\min}(x_*)$$

for all $x \in \Omega$ such that $\|x - x_*\| \le \delta$. □

Let Φ be differentiable on an open set that contains Ω and consider the nonlinear programming problem

$$\text{Minimize } \Phi(x) \text{ subject to } x \in \Omega. \tag{6}$$

Necessary Optimality Conditions (NOC) are conditions that must be satisfied by local minimizers of (6). For example, if $\Omega = I\!R^n$, the requirement "$\nabla\Phi(x) = 0$" is a NOC. In constrained Optimization, Necessary Optimality Conditions usually take the form: If a constraint qualification is satisfied at x_*, then the KKT conditions hold (see, for example [12]). Constraint qualifications only involve properties of Ω whereas the KKT conditions involve the gradient of f and the gradients of the constraints.

Theorem 2.1 allows us to prove the following corollary.

Corollary 2.1 *Let $x_* \in \Omega$ be a local minimizer of the problem (3), where all the functions f_i are differentiable in an open set that contains Ω. Then, for all $i \in I_{\min}(x_*)$, x_* satisfies the necessary optimality conditions associated with the problem*

$$\textit{Minimize } f_i(x) \textit{ subject to } x \in \Omega. \tag{7}$$

Proof By Theorem 2.1, x_* is a local minimizer of f_i for all $i \in I_{\min}(x_*)$. Therefore, x_* satisfies the necessary optimality conditions associated with this problem. □

Corollary 2.1 motivates the following definitions. Given a NOC for nonlinear programming, we say that $x_* \in \Omega$ is *strongly critical* if, for all $i \in I_{\min}(x_*)$, x_* satisfies NOC, associated with the problem (7).

We say that $x_* \in \Omega$ is *weakly critical* if there exists $i \in I_{\min}(x_*)$ such that x_* satisfies NOC, associated with (7).

3 Unconstrained LOVO algorithm with convergence to weakly critical points

Optimization algorithms for solving nonlinear programming problems (6) are iterative. At each iteration, the functional values, the gradients and, perhaps, the second derivatives of the objective function and the constraints are generally required. Users of computer codes that implement nonlinear programming algorithms must provide subroutines that evaluate these quantities.

In the presence of the problems (1) or (3) one is tempted to use any well established optimization method for smooth problems. Each time the (perhaps non-existent) $\nabla f_{\min}(x)$ is required by the algorithm, one may choose $i \in I_{\min}(x)$ and "define"

$$\nabla f_{\min}(x) \leftarrow \nabla f_i(x). \tag{8}$$

(We may proceed in a similar way if the algorithm also requires Hessians).

The question that we address in this section is: what happens if we proceed in that way? As is well-known, to use such a strategy in many nonsmooth problems may be catastrophic. However, we will show here that, in the case of (1–3), the consequences are less severe. Essentially, we will show that convergence to weakly critical points necessarily occurs. It is easy to see that weakly critical points are Clarke-stationary points [10,15] of the problem of minimizing $f_{\min}$ (see [16, Sect. 2.5.1]).

Algorithm **U1**, defined below, applies to the unconstrained minimization ($\Omega = I\!R^n$) of $f_{\min}(x)$. We assume that the functions f_i are continuously differentiable for all $x \in I\!R^n$. This algorithm may be interpreted as a straightforward application of a smooth unconstrained minimization method to the unconstrained LOVO problem with the "wrong evaluation" (8).

Algorithm U1 Let $\theta \in (0,1), \alpha \in (0,1), M > 1, \beta > 0, t_{\text{one}} > 0$ be algorithmic parameters. Let $x_0 \in I\!R^n$ be the initial approximation. Given $x_k \in I\!R^n$, the steps for computing x_{k+1} are:

Step 1 Choose $\nu(k) \in I_{\min}(x_k)$. If $\|\nabla f_{\nu(k)}(x_k)\| = 0$, terminate.
Step 2 Compute $d_k \in I\!R^n$ such that $\nabla f_{\nu(k)}(x_k)^T d_k \leq -\theta \|d_k\| \|\nabla f_{\nu(k)}(x_k)\|$ and $\|d_k\| \geq \beta \|\nabla f_{\nu(k)}(x_k)\|$.
Step 3 Compute $t_k > 0$, $x_{k+1} \in I\!R^n$, such that

$$f_{\min}(x_{k+1}) \leq f_{\min}(x_k) + \alpha t_k \nabla f_{\nu(k)}(x_k)^T d_k \tag{9}$$

and

$$[t_k \geq t_{\text{one}}] \quad \text{or} \quad \left[f_{\min}(x_k + \bar{t}_k d_k) > f_{\min}(x_k) + \alpha \bar{t}_k \nabla f_{\nu(k)}(x_k)^T d_k \ \text{ for some } \bar{t}_k \leq M t_k \right]. \tag{10}$$

The line-search strategy (9, 10) admits different implementations. The most straightforward one is backtracking. In this case, t_k is chosen as the first number of the sequence $\{1, 2^{-1}, 2^{-2}, \ldots\}$ that satisfies (9) and $x_{k+1} = x_k + t_k d_k$. In this case $t_{\text{one}} = 1$ and $M = 2$. However, the choice based on (9, 10) admits more sophisticated and efficient line-search procedures (see, for example, [13]).

Recall that, in the unconstrained LOVO problem, a weakly critical point is a point where $\nabla f_i(x) = 0$ for some $i \in I_{\min}(x)$. In the following theorems we prove that the algorithm stops at x_k only if x_k is weakly critical and that limit points of sequences generated by Algorithm **U1** are weakly critical. Theorems 3.1 and 3.2 may be proved using similar techniques to the ones used in the unconstrained minimization algorithms presented in [9] and [26]. Complete proofs may be found in the expanded report [3].

Theorem 3.1 *Algorithm* **U1** *is well-defined and terminates at x_k only if x_k is weakly critical.*

Theorem 3.2 *If x_* is a limit point of a sequence generated by Algorithm* **U1** *then x_* is weakly critical. Moreover, if $\lim_{k\in K} x_k = x_*$ and the same $i = \nu(k) \in I_{\min}(x_k)$ is chosen at Step 1 of the algorithm for infinitely many indices $k \in K$, then $i \in I_{\min}(x_*)$ and $\nabla f_i(x_*) = 0$. Finally, $\lim_{k\in K} \|\nabla f_{\nu(k)}(x_k)\| = 0$.*

In [3] we addressed the local convergence of Algorithm **U1**. The choice of x_{k+1} in this algorithm imposes that $f_{\min}(x_{k+1}) \le f_{\min}(x_k) + \alpha t_k \nabla f_{\nu(k)}(x_k)^T d_k$. This property is obviously satisfied if $x_{k+1} = x_k + t_k d_k$ but, for enhancing the probability of convergence to global minimizers, other accelerated definitions for x_{k+1} are possible and, possibly, desirable. For local convergence, however, the distance between x_{k+1} and x_k must be small if x_k is close to being critical. Using quasi-Newton choices of the search direction the convergence is shown to be superlinear and using Newtonian choices the convergence is quadratic [3].

4 Unconstrained LOVO algorithm with convergence to strongly critical points

In Sect. 3 we introduced Algorithm **U1** which, briefly speaking, converges to weakly critical points. In principle, Algorithm **U1** may converge to points that are not strongly critical. For example, consider the problem defined by $f_1(x) = x$, $f_2(x) = x^2$, $m = 2$. For all $x \in (0, 1)$ we have that $f_{\min}(x) = x^2$. Therefore, it is easy to define a sequence $x_k \in (0, 1)$ generated by Algorithm **U1** and converging to 0. Of course, 0 is a weakly critical point, but it is not strongly critical. The objective of this section is to introduce and analyze an unconstrained algorithm that converges to strongly critical points. The idea is to work, at each iteration, with a set of search directions corresponding to the indices such that $f_j(x_k) = f_{\min}(x_k)$. In this way, essentially, the new point will exhibit sufficient descent with respect to all these directions and, in the limit, this property will guarantee strong criticality.

Algorithm U2 Let $\theta \in (0, 1), \alpha \in (0, 1), M > 1, \beta > 0, t_{\text{one}} > 0, \varepsilon > 0, \delta > 0$ be algorithmic parameters. Let $x_0 \in I\!R^n$ be the initial approximation. Given $x_k \in I\!R^n$, the steps for computing x_{k+1} are:

Step 1 If $\|\nabla f_i(x_k)\| = 0$ for all $i \in I_{\min}(x_k)$, terminate the execution of the algorithm.
If $\|\nabla f_i(x_k)\| > \delta$ for all $i \in I_{\min}(x_k)$, choose $i \in I_{\min}(x_k)$ and define $J_k = \{i\}$. Otherwise, define

$$J_k = \{j \in \{1, \dots, m\} \mid f_j(x_k) \le f_{\min}(x_k) + \varepsilon \ \text{ and } \ \nabla f_j(x_k) \ne 0\}.$$

Step 2 For all $i \in J_k$, compute $d_k^i \in I\!R^n$ such that

$$\nabla f_i(x_k)^T d_k^i \le -\theta \|d_k^i\| \|\nabla f_i(x_k)\| \quad \text{and} \quad \|d_k^i\| \ge \beta \|\nabla f_i(x_k)\|. \tag{11}$$

Step 3 For all $i \in J_k$, compute $t_k^i > 0$ such that

$$f_i(x_k + t_k^i d_k^i) \le f_i(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i \tag{12}$$

and

$$\left[t_k^i \ge t_{\text{one}}\right] \quad \text{or} \quad \left[f_i(x_k+\bar{t}_k d_k^i) > f_i(x_k) + \alpha \bar{t}_k^i \nabla f_i(x_k)^T d_k^i \ \text{ for some } \ \bar{t}_k^i \le M t_k^i\right]. \tag{13}$$

Step 4 Compute $x_{k+1} \in I\!R^n$ such that

$$f_{\min}(x_{k+1}) \leq \min_{i \in J_k}\{f_i(x_k + t_k^i d_k^i)\}. \tag{14}$$

In Algorithm **U2**, if $\|\nabla f_i(x_k)\| > \delta$ for all $i \in I_{\min}(x_k)$ the iteration is identical to the one of Algorithm **U1**. If, for some $i \in I_{\min}(x_k)$ the gradient norm is smaller than δ we compute descent directions for all the functions f_i such that $f_i(x_k) = f_{\min}(x_k)$ (with precision ε). Then, we perform line searches along all these directions and we finish taking x_{k+1} such that this point is at least as good as all the points obtained in the line searches. The most obvious way to choose x_{k+1} is to set $x_{k+1} = x_k + t_k^j d_k^j$, where $j \in J_k$ and

$$f_j(x_k + t_k^j d_k^j) \leq f_i(x_k + t_k^i d_k^i) \quad \forall\, i \in J_k.$$

However, the choice (14) allows one to use extrapolation steps to enhance the chance of convergence to global minimizers.

Below we show that the algorithm is well defined and can stop only at strongly critical points.

Theorem 4.1 *Algorithm* **U2** *is well-defined and terminates at x_k if, and only if, x_k is strongly critical. Moreover, if the algorithm does not terminate at x_k,*

$$f_{\min}(x_{k+1}) < f_{\min}(x_k) \tag{15}$$

for all $k = 0, 1, 2, \ldots$.

Proof If x_k is strongly critical, Step 1 guarantees that the algorithm terminates at x_k.

Let us show now that, if x_k is not strongly critical, the iteration that defines Algorithm **U2** can be completed in finite time and that x_{k+1} satisfies (15).

If x_k is not strongly critical, there exists $i \in I_{\min}(x_k)$ such that $\|\nabla f_i(x_k)\| \neq 0$. Therefore, the set J_k is nonempty and, by construction, for all $i \in J_k$, $\nabla f_i(x_k) \neq 0$. Therefore, as in the proof of Theorem 3.1, for all $i \in J_k$ and t small enough, the sufficient descent condition

$$f_i(x_k + t d_k^i) \leq f_i(x_k) + \alpha t \nabla f_i(x_k)^T d_k^i$$

is verified. Therefore, choosing t_k^i as the first number in the sequence $\{t_{\text{one}}, t_{\text{one}}/M, t_{\text{one}}/M^2, \ldots\}$ that satisfies (12), the conditions (12) and (13) are satisfied. So, the algorithm is well defined.

Now, let $i \in I_{\min}(x_k)$ be such that $\nabla f_i(x_k) \neq 0$. Since $i \in J_k$ we have that:

$$f_i(x_k + t_k^i d_k^i) \leq f_i(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i = f_{\min}(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i < f_{\min}(x_k).$$

Therefore, (15) follows from (14). □

In Lemma 4.1 we prove that, in a convergent subsequence generated by Algorithm **U2**, at most finitely many iterations are of type **U1**.

Lemma 4.1 *Assume that $\{x_k\}$ is an infinite sequence generated by Algorithm* **U2** *and K is an infinite sequence of indices such that $\lim_{k \in K} x_k = x_*$. Then, for all $k \in K$ large enough,*

$$\min_{i \in I_{\min}(x_k)}\{\|\nabla f_i(x_k)\|\} \leq \delta.$$

Proof Assume that the thesis is not true. Then, there exists K_1, an infinite subsequence of K, such that

$$\|\nabla f_i(x_k)\| > \delta \quad \text{for all } \; i \in I_{\min}(x_k), k \in K_1. \tag{16}$$

Define $K_1 = \{k_0, k_1, k_2, k_3, \ldots\}$, $k_j < k_{j+1}$ for all j and

$$y_j = x_{k_j} \quad \text{for all} \quad j = 0, 1, 2, \ldots.$$

By (16) and the choice of J_k in this case, the sequence $\{y_j\}$ is generated as in Algorithm **U1**. Therefore, there exists $i \in \{1, \ldots, m\}$ such that $J_{k_j} = \{i\} \subset I_{\min}(x_{k_j})$ infinitely many times. By Theorem 3.2, $i \in I_{\min}(x_*)$ and $\nabla f_i(x_*) = 0$. Therefore, by the continuity of ∇f_i, $\lim_{j\to\infty} \|\nabla f_i(x_{k_j})\| = 0$. This implies that (16) is false. □

In Theorem 4.2 we prove that Algorithm **U2** necessarily produces strongly critical points.

Theorem 4.2 *If x_* is a limit point of a sequence generated by Algorithm* **U2**, *then x_* is strongly critical. Moreover, given $\epsilon > 0$, there exists $k \in I\!N$ such that*

$$\|\nabla f_i(x_k)\| \le \epsilon \quad \textit{for all} \;\; i \in I_{\min}(x_k).$$

Proof Let $K = \{k_0, k_1, k_2, \ldots\}$ be such that $\lim_{k\in K} x_k = x_*$.

By Lemma 4.1 and the definition of Algorithm **U2**, we may assume, without loss of generality, that

$$J_k = \{j \in \{1, \ldots, m\} \mid f_j(x_k) \le f_{\min}(x_k) + \varepsilon \;\text{ and }\; \nabla f_j(x_k) \ne 0\}$$

for all $k \in K$.

Assume that $i \in I_{\min}(x_*)$. Our aim is to prove that $\nabla f_i(x_*) = 0$.

Clearly, $f_i(x_*) = f_{\min}(x_*)$. So, by the continuity of f_i and $f_{\min}$,

$$f_i(x_k) \le f_{\min}(x_k) + \varepsilon. \tag{17}$$

for $k \in K$ large enough. By continuity, if $\nabla f_i(x_k)$ vanishes infinitely many times for $k \in K$, we are done. Otherwise, we may assume, without loss of generality, that $\nabla f_i(x_k) \ne 0$ for all $k \in K$. Therefore, by (17), $i \in J_k$ for all $k \in K$. Moreover,

$$\lim_{k\in K} f_i(x_k) - f_{\min}(x_k) = f_i(x_*) - f_{\min}(x_*) = 0. \tag{18}$$

By the definition of the algorithm, for j large enough we have:

$$\begin{aligned} f_{\min}(x_{k_{j+1}}) &< f_{\min}(x_{k_j+1}) \le f_i(x_{k_j} + t^i_{k_j} d^i_{k_j}) \le f_i(x_{k_j}) + \alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j} \\ &= f_{\min}(x_{k_j}) + [f_i(x_{k_j}) - f_{\min}(x_{k_j})] + \alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j}. \end{aligned} \tag{19}$$

By (11), $\alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j} < 0$. Assume, for a moment, that there exists $c > 0$, $j_0 \in I\!N$, such that

$$\alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j} < -c \tag{20}$$

for all $j \ge j_0$. But, by (18), there exists $j_1 \ge j_0$ such that

$$f_i(x_{k_j}) - f_{\min}(x_{k_j}) < c/2 \tag{21}$$

for all $j \ge j_1$. So, by (19), (20) and (21), we have that $f_{\min}(x_{k_{j+1}}) \le f_{\min}(x_{k_j}) - c/2$ for all $j \ge j_1$. This implies that $\lim_{j\to\infty} f_{\min}(x_{k_j}) = -\infty$ and contradicts the fact that, by continuity, $f_{\min}(x_{k_j}) \to f_{\min}(x_*)$. Therefore, the existence of c and j_0 with the property (20) is impossible. This implies that there exists K_1, an infinite subsequence of K, such that $\lim_{k\in K_1} \alpha t^i_k \nabla f_i(x_k)^T d^i_k = 0$. Therefore, by (11), $\lim_{k\in K_1} t^i_k \|\nabla f_i(x_k)\| \|d^i_k\| = 0$. If, for some subsequence $K_2 \subset K_1$, $\lim_{k \in K_2} \nabla f_i(x_k) = 0$, we are done. So, let us assume that $\|\nabla f_i(x_k)\|$ is bounded away from zero for $k \in K_1$. In this case,

$$\lim_{k \in K} t_k^i \|d_k^i\| = 0. \tag{22}$$

If, for some subsequence $K_3 \subset K_1$, $\lim_{k \in K_3} \|d_k^i\| = 0$, then, by (11), $\lim_{k \in K_3} \|\nabla f_i(x_k)\| = 0$ and, thus, $\nabla f_i(x_*) = 0$. So, we only need to consider the case in which $\|d_k^i\|$ is bounded away from zero for $k \in K_1$. In this case, by (22), $\lim_{k \in K_1} t_k^i = 0$.

Therefore, without loss of generality, we may assume that $t_k^i < t_{\text{one}}$ for all $k \in K_1$. Then, by (13), there exist $\bar{t}_k^i \le M t_k^i$, $s_k = \bar{t}_k^i d_k^i$ such that

$$f_i(x_k + s_k) > f_i(x_k) + \alpha \nabla f_i(x_k)^T s_k \quad \text{for all} \quad k \in K_1 \tag{23}$$

and, by (22),

$$\lim_{k \in K_1} \|s_k\| = 0. \tag{24}$$

So, by (23) and the Mean Value Theorem, there exists $\xi_k \in [0, 1]$ such that

$$\nabla f_i(x_k + \xi_k s_k)^T s_k = f_i(x_k + s_k) - f_i(x_k) > \alpha \nabla f_i(x_k)^T s_k \tag{25}$$

for all $k \in K_1$. Moreover, by (11),

$$\frac{\nabla f_i(x_k)^T s_k}{\|s_k\|} \le -\theta \|\nabla f_i(x_k)\| \tag{26}$$

for all $k \in K_1$. Let K_4 be a subsequence of K_1 such that $\lim_{k \in K_4} \frac{s_k}{\|s_k\|} = s$. By (24), dividing both sides of (25) by $\|s_k\|$ and taking limits for $k \in K_4$, we obtain:

$$\nabla f_i(x_*)^T s \ge \alpha \nabla f_i(x_*)^T s.$$

Since $\alpha < 1$ and $\nabla f_i(x_k)^T d_k < 0$ for all k, this implies that $\nabla f_i(x_*)^T s = 0$. Taking limits on both sides of (26) we obtain that $\|\nabla f_i(x_*)\| = 0$.

Let us prove the second part of the thesis. If it is not true, then there exists K_5, an infinite subset of K and $\epsilon > 0$ such that for all $k \in K_5$ there exists $i \in I_{\min}(x_k)$ such that $\|\nabla f_i(x_k)\| > \epsilon$. Clearly, the same index i must be repeated infinitely many times, and, taking limits, we obtain that $i \in I_{\min}(x_*)$ and $\|\nabla f_i(x_*)\| \ge \epsilon$. This contradicts the first part of the thesis. □

Local and superlinear convergence results may be found in [3].

Remark Both Algorithms **U1** and **U2** aim convergence to local minimizers. Strong criticality and weak criticality are generalizations of the condition $\nabla f(x) = 0$ of smooth unconstrained optimization. Strong criticality is stronger than weak criticality so that, in principle, Algorithm **U2** should be preferred. However (a) The iterations of Algorithm **U2** are more expensive than the ones of Algorithm **U1**; (b) In all our experiments, the critical points obtained by Algorithm **U1** were strongly critical. Of course, the existence of relevant applications in which the use of Algorithm **U2** could be essential for obtaining practical solutions cannot be discarded.

5 Constrained LOVO problems

In this section we consider LOVO problems with constraints and we define a natural extension of the Augmented Lagrangian Algorithm (in the version described in [7]) to these problems.

We consider the problem

$$\text{Minimize } f_{\min}(x) \text{ subject to } h(x) = 0, \quad g(x) \le 0, \tag{27}$$

where $f_i : I\!R^n \to I\!R$ for all $i = 1, \dots, m$, $h : I\!R^n \to I\!R^{n_h}$, $g : I\!R^n \to I\!R^{n_g}$ and all these functions are smooth.

For all $x \in I\!R^n$, $\rho \in I\!R_{++}$, $\lambda \in I\!R^{n_h}$, $\mu \in I\!R_+^{n_g}$ we define the Augmented Lagrangian associated with f_i by:

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

The Augmented Lagrangian associated with $f_{\min}$ is defined by

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

Let us define, for all $x \in I\!R^n$,

$$I_{\min}(x) = \{i \in \{1, \dots, m\} \mid f_i(x) = f_{\min}(x)\}.$$

Observe that

$$I_{\min}(x) = \{i \in \{1, \dots, m\} \mid L_i(x, \lambda, \mu, \rho) = L_{\min}(x, \lambda, \mu, \rho)\}$$

for all $\lambda \in I\!R^m$, $\mu \in I\!R_+^p$, $\rho > 0$.

5.1 Algorithm C-LOVO

Let $x_0 \in I\!R^n$ be an arbitrary initial point.

The parameters for the execution of the algorithm are: $\tau \in [0, 1)$, $\gamma > 1$, $-\infty < \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty$, $0 \le \bar{\mu}_{\max} < \infty$, $\rho_1 \in I\!R_{++}$, $[\bar{\lambda}_1]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ $\forall j = 1, \dots, n_h$, $[\bar{\mu}_1]_j \in [0, \bar{\mu}_{\max}]$ $\forall j = 1, \dots, n_g$, $\varepsilon_1 > 0$.

Step 1 *Initialization*
Set $k \leftarrow 1$. For $j = 1, \dots, n_g$, compute $[\sigma_0]_j = \max\{g_j(x_0), 0\}$.
Step 2 *Solving the subproblem*
Compute $x_k \in I\!R^n$ such that

$$\|\nabla L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)\|_\infty \le \varepsilon_k \tag{28}$$

for some $i \in I_{\min}(x_k)$.
Step 3 *Estimate multipliers*
For all $j=1, \dots, n_h$, compute $[\lambda_{k+1}]_j=[\bar{\lambda}_k]_j + \rho_k h_j(x_k)$ and $[\bar{\lambda}_{k+1}]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$.
For all $j=1, \dots, n_g$, compute $[\mu_{k+1}]_j=\max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\}$, $[\sigma_k]_j=\max\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\}$, and $[\bar{\mu}_{k+1}]_j \in [0, \bar{\mu}_{\max}]$.
Step 4 *Update the penalty parameters*
If $\max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \le \tau \max\{\|h(x_{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\}$, define $\rho_{k+1} = \rho_k$.
Else, define

$$\rho_{k+1} = \gamma \rho_k.$$

Step 5 *Begin a new outer iteration*
Compute $\varepsilon_{k+1} > 0$. Set $k \leftarrow k + 1$. Go to Step 2.
The obvious way to solve (28) is to apply Algoritm **U1** or Algorithm **U2** to

$$\text{Minimize } L_{\min}(x, \bar{\lambda}_k, \bar{\mu}_k, \rho_k).$$

Both algorithms guarantee that a point satisfying (28) can be found, provided that the generated sequence is bounded.

In Theorem 5.1 we prove that Algorithm **C-LOVO** finds stationary points of the constraint infeasibility.

Theorem 5.1 *Assume that* $\{x_k\}$ *is an infinite sequence generated by Algorithm* **C-LOVO** *with* $\varepsilon_k \to 0$ *and that* x_* *is a limit point. Then,* x_* *is a stationary point of*

$$\textit{Minimize} \sum_{j=1}^{n_h} h_j(x)^2 + \sum_{j=1}^{n_g} \max\{0, g_j(x)\}^2.$$

Proof Since $\{x_k\}$ is infinite, there exists $i \in \{1, \dots, m\}$ such that (28) holds for f_i infinitely many times. Taking the corresponding subsequence of $\{x_k\}$, it turns out that this subsequence may be thought to be generated by the Augmented Lagrangian Algorithm [7] applied to the minimization of f_i. Therefore, the thesis follows as in Theorem 4.1 of [7]. □

In the next theorem we prove that feasible limit points are stationary provided that they satisfy the Constant Positive Linear Dependence (CPLD) constraint qualification. The CPLD condition was introduced in [30] and its status as a constraint qualification was elucidated in [4]. This condition says that, if a subset of gradients of active constraints are positive linear dependent (linearly dependent with nonnegative coefficients corresponding to the inequalities) at a feasible point $\bar{x}$, then the same set of gradients is linearly dependent for all points (feasible or not) in a neighborhood of $\bar{x}$. The CPLD condition is strictly weaker than the Mangasarian-Fromovitz [24,31] constraint qualification.

Theorem 5.2 *Assume that* $\{x_k\}$ *is an infinite sequence generated by Algorithm* **C-LOVO** *with* $\varepsilon_k \to 0$, x_* *is a limit point and the CPLD constraint qualification is fulfilled at* x_**. Then, there exists* $i \in I_{\min}(x_*)$ *such that* x_* *is a KKT point of*

$$\textit{Minimize } f_i(x) \ \textit{ subject to } \ h(x) = 0, g(x) \le 0.$$

Proof As in Theorem 5.1, consider an infinite subsequence of $\{x_k\}$ such that (28) holds with the same index i for all the terms of this subsequence. Again, this subsequence may be thought as having been generated by the smooth Augmented Lagrangian Algorithm applied to the minimization of f_i. By Theorem 4.2 of [7], there exists x_* satisfying the thesis of the present theorem. The fact that $i \in I_{\min}(x_*)$ follows trivially from $L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k) \le L_j(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)$ for all j. □

Remark In (28) we assume that, at each outer iteration of Algorithm **C-LOVO** we obtain an approximate weak critical point of the unconstrained Augmented Lagrangian. With this assumption, we obtain, in Theorem 5.2, a weak critical point of the constrained LOVO problem. Let us show that the strong-criticality of x_k would not guarantee strong criticality at the solution of the constrained problem. Take $n = 1, p = 2, n_h = 0, n_g = 1, f_1(x) = (x-1)^2/2$, $f_2(x) = (x + 1)^2/2$, $g_1(x) = x$. Define $\bar{\mu}_k = 0$ for all k. Then:

$$L_1(x) = f_1(x) + \rho(x_+)^2/2, \ L_2(x) = f_2(x) + \rho(x_+)^2/2$$

For all k we have that $x_k \equiv 1/(1+\rho_k)$ is a strong critical point of $L_{\min}$. (x_k is a minimizer of L_1 and $L_2(x_k) > L_1(x_k)$ for all k.) Clearly, $x_* = 0$ is a minimizer of $f_1(x)$ subject to $g_1(x) \leq 0$ but is not a KKT point of f_2 subject to the same constraint. However, $f_1(x_*) = f_2(x_*)$, so x_* is not a strong critical point of the constrained problem. This example shows the assumption of strong criticality at x_k would be useless in terms of the solutions that can be obtained by **C-LOVO**.

Under additional local assumptions, similarly to [7], it may be proved that penalty parameters remain bounded. See [3] for details.

If one is able to find global minimizers of the unconstrained subproblems addressed in the Augmented Lagrangian Algorithm, then limit points are global minimizers of the constrained problem, since this property depends only on the continuity of the objective function and constraints [7]. We state this important property in the following theorem. A generalization of this theorem may be found in [14].

Theorem 5.3 *Assume that the feasible region of the original constrained problem is nonempty and that, in Algorithm* **C-LOVO**, *x_k is a global minimizer of the subproblem for all k. Then, every limit point of $\{x_k\}$ is a global minimizer of the constrained LOVO problem.*

6 Hidden patterns and Protein Alignment

Let $\mathcal{Q} = \{Q_1, \ldots, Q_N\} \subset I\!R^{n_q}$, $\mathcal{P} = \{P_1, \ldots, P_M\} \subset I\!R^{n_p}$. The goal is to find the structure defined by $\mathcal{Q}$ in the set $\mathcal{P}$. Strictly speaking, we aim to find a transformation operator $D : I\!R^{n_q} \rightarrow I\!R^{n_p}$ such that some subset of $\{D(Q_1), \ldots, D(Q_N)\}$ fits some subset of $\mathcal{P}$. For example, assume that $n_q = 3, n_p = 2$ and that the $\mathcal{P}$ is the set of possible "shadows" of the points in $\mathcal{Q}$. Therefore, we wish to find the rigid-body displacement of $\mathcal{Q}$ such that (say) the two-dimensional points represented by the $x - y$ coordinates of the displaced $\mathcal{Q}$ fit $\mathcal{P}$ in the best possible way. So, D will be the composition of a rigid-body movement with a projection. A lot of applications of this general problem can be given, from medicine tissue recognition to security systems. Let us show here how the problem can be modelled in terms of LOVO.

Define $\mathcal{N}$ the set of $N-$uples $\nu = (\nu(1), \ldots, \nu(N))$, where $\nu(i) \in \{1, \ldots, M\}$ for all $i = 1, \ldots, N$. (In other words, $\mathcal{N} = \{1, \ldots, M\}^N$).

Let D be an admissible transformation. For all $\nu \in \mathcal{N}$ we define $f_\nu(D) = \sum_{i=1}^N \|D(Q_i) - P_{\nu(i)}\|^2$. Finally, $f_{\min}(D) = \min_{\nu \in \mathcal{N}} f_\nu(D)$.

If there exists a set of N points of $\mathcal{P}$ that fits exactly a displacement D of $\mathcal{Q}$ we have that $f_{\min}(D) = 0$. The problem of minimizing $f_{\min}$ falls under the theory introduced in previous sections.

Fortunately, the evaluation of $f_{\min}$ does not need the computation of all the functions f_ν. In fact, given a transformation D, we compute, for all $i = 1, \ldots, N$, $P_{c(i)}(D) \in \mathcal{P}$ such that

$$\|D(Q_i) - P_{c(i)}(D)\| \leq \|D(Q_i) - P\| \ \ \forall\ P \in \mathcal{P}. \tag{29}$$

Then, $f_{\min}(D) = \sum_{i=1}^N \|D(Q_i) - P_{c(i)}(D)\|^2$.

The two most common situations in applications correspond to dim $= 2$ and dim $= 3$. In the first case the displacement may be represented by three parameters: the translation of the center of gravity of $\mathcal{Q}$ and the angle of rotation. In the three-dimensional case, displacements may be represented by the translation vector and three rotations, although other alternatives are possible.

A generalization of this problem is to find a common structure to the sets $\mathcal{P}$ and $\mathcal{Q}$. Suppose that we want to find a displacement D such that there exists $R \leq N$ points of $\mathcal{Q}$ (say, $Q_{j_1}, \ldots, Q_{j_R}$) such that $D(Q_{j_1}), \ldots, D(Q_{j_R})$ fit R points of $\mathcal{P}$. In this case, we define $\mathcal{M}$ as the Cartesian product between the subsets of R elements of $\{1, \ldots, N\}$ and the R-uples of $\{1, \ldots, M\}$. For all $\nu = (\{j_1, \ldots, j_R\}, (i_1, \ldots, i_R)) \in \mathcal{M}$, we define

$$f_\nu(D) = \sum_{\ell=1}^{R} \|D(Q_{j_\ell}) - P_{i_\ell}\|^2$$

and the goal is to minimize $f_{\min}(D) \equiv \min_{\nu \in \mathcal{M}} f_\nu(D)$. Again, the computation of $f_{\min}$ is simple: for all $i = 1, \ldots, N$ compute $P_{c(i)}(D) \in \mathcal{P}$ as in (29). Then, $f_{\min}(D)$ is the sum of the R smaller values of $\|D(Q_i) - P_{c(i)}(D)\|^2$.

Although the most obvious definition of a displacement operator involves only translation, rotations and projections, more general definitions are possible. For example, the introduction of an additional parameter allows one to consider scale variations so that a given form may be recognized in a structure regardless of its size. Moreover, if we replace the Euclidean norm of the difference by a different distance function, we may obtain many alternative case-oriented similarity measures.

Protein Alignment is a particularly important problem related to hidden-pattern identification. The goal is to find similarities between two proteins $\mathcal{P}$ and $\mathcal{Q}$, represented by the coordinates of their Cα atoms. The similarity is measured by a *score*. Several scores have been proposed in the protein literature. One of the most popular ones is the *Structal Score*, the definition of which is given now. Assume that the 3D-coordinates of the Cα atoms of protein $\mathcal{P}$ (in angstroms) are $P_1, \ldots, P_M$ and the coordinates of the Cα atoms or protein $\mathcal{Q}$ are $Q_1, \ldots, Q_N$. Under the rigid-body displacement D, the coordinates of the displaced protein $\mathcal{Q}$ are, therefore, $D(Q_1), \ldots, D(Q_N)$. Assume that Φ is a monotone bijection between a subset of $\{1, \ldots, M\}$ and a subset of $\{1, \ldots, N\}$. (We mean that $i < j \Rightarrow \Phi(i) < \Phi(j)$). The Structal Score associated with the displacement D and the bijection Φ is:

$$StS(D, \Phi) = \sum \frac{20}{1 + \|P_k - D(Q_{\Phi(k)})\|^2/5}, -10 \times \text{gaps}, \tag{30}$$

where the $\sum$ symbol involves the pairs $(k, \Phi(k))$ defined by the bijection and *gaps* is the number of cases in which at least one of the following situations occur:

- $\Phi(k)$ is defined, there exists $\ell > k$ such that $\Phi(\ell)$ is defined, but $\Phi(\ell + 1)$ is not defined;
- $\Phi^{-1}(k)$ is defined, there exists $\ell > k$ such that $\Phi^{-1}(\ell)$ is defined, but $\Phi^{-1}(\ell + 1)$ is not defined.

The Structal Alignment Problem consists of finding Φ and D such that $StS(D, \Phi)$ is maximal. A global optimization procedure for achieving this objective was given in [22]. However, this method is not computationally affordable (see [22]) and, in practice, an heuristic procedure called *Structal Method* [33] is generally used. In [23], the *Structal Method* was reported as the best available practical algorithm for protein alignment. Each iteration of the Structal Method consists of two steps:

1. Update Φ: Given the positions $P_1, \ldots, P_M$ and $D(Q_1), \ldots, D(Q_N)$, the monotone bijection Φ that maximizes StS (fixing D) is computed using Dynamic Programming.
2. Update D: Assume that the graph of Φ is $\{(k_1, \Phi(k_1)), \ldots, (k_s, \Phi(k_s))\}$. Then, the rigid-body displacements that minimizes $\sum_{\ell=1}^{s} \|P_{k_\ell} - D(Q_{\Phi(k_\ell)}\|^2$ is computed.

The computation of D at the second step of the Structal Method involves the solution of the well known Procrustes problem [20,21]. The main drawback of the Structal Method is that the Update-Φ step aims at the optimization of a function (the Structal Score) with respect to Φ and the Update-D step involves the optimization of a different function (the sum of squared distances) with respect to D. This may lead to oscillation [26]. With the aim of overcoming this problem we suggest a different algorithm (DP-LS), where the Update-Φ phase at each iteration of the Structal Method is maintained but the Update-D iteration is modified according to LOVO principles.

The idea is the following. Assume that $\{\Phi_1, \ldots, \Phi_m\}$ is the set of all the monotone bijections between a subset of $\{1, \ldots, M\}$ and a subset of $\{1, \ldots, N\}$. For each $i = 1, \ldots, m$ and for each rigid-body displacement D, we define:

$$f_i(D) = -StS(D, \Phi_i).$$

Observe that f_i is a smooth function of the displacement vector D. The Update-Φ phase of the Structal Method, in the LOVO terminology, consists of finding $i_1(D)$. Dynamic Programming is a quite efficient algorithm for this purpose. The second (Update-D) phase of the DP-LS method consists of the computation of a search direction in the D-space for f_{i_1} (we used a safeguarded Newton procedure) and the application of the ordinary line-search of Algorithm **U1**. Therefore, DP-LS is Algorithm **U1** applied to the maximization of the Structal Score, both with respect to Φ and D.

The application of DP-LS to the alignment of proteins of the Protein Data Bank (PDB) [11] is fully described in [26]. Using 79,800 individual protein comparisons it has been concluded that DP-LS obtains the best scores in most meaningful cases and that the computer time used by DP-LS is, on average, $2/3$ the computer time employed by the Structal Method on the tests reported in [26].

These facts are quite encouraging and makes the comparison of a single protein to all the proteins of the PDB quite efficient and the all-to-all comparison affordable.

An additional LOVO algorithm for Protein Alignment (NB-Newton) was presented in [26]. With the aim of improving computer time, instead of a monotone bijection, an arbitrary correspondence is used. For good alignments, this algorithm obtained comparable scores to DP-LS and it was 6 times faster than the Structal Method in terms of computer time. Other LOVO methods for different types of chemical structures comparisons were suggested in [9]. Algorithms for Protein Alignment based on LOVO ideas are publicly available in our site www.ime.unicamp.br/~martinez/lovoalign. On-line alignments can be performed using the facilities of this site. Initial approximations for the application of **U1** were obtained using a specific heuristic described in [26]. An algorithm that converges to global solutions of the Protein Alignment problem was introduced in [22]. However, this algorithm is based on evaluation of the objective function (which involves Dynamic Programming) on a grid in the parameter space and, so, it is not practical.

7 Numerical examples

One of the main practical consequences of the theory introduced in Sections 2–5 of this paper is that, in spite of the nonsmoothness of the LOVO problem, if we ignore the multiplicity of gradients at a given point x_k and we use straightforward smooth minimization solvers, the bad consequences are rather mild. Many smooth optimization algorithms, when applied to LOVO, may be considered particular cases of Algorithms **U1** and **C-LOVO**. With this

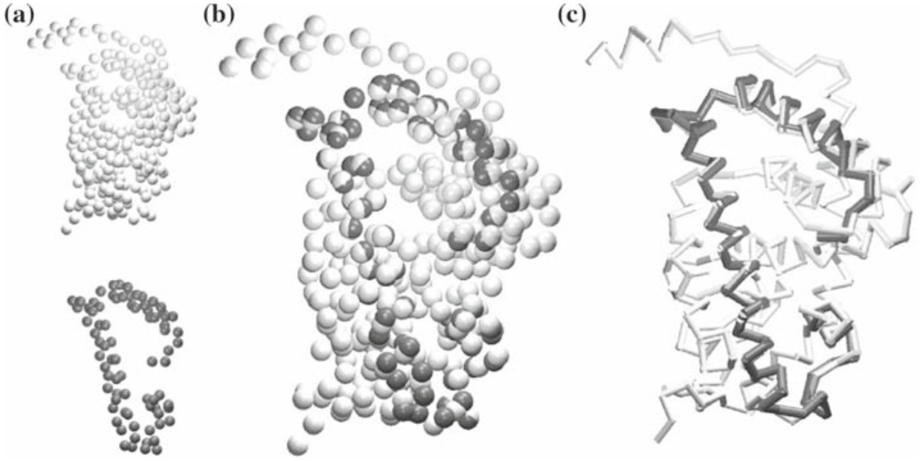

Fig. 1 Finding patterns of protein folding with LOVO

property in mind, we used, in our experiments, the unconstrained and constrained versions of `Algencan`, the nonlinear-programming code available in the `Tango` project web-page (www.ime.usp.br/~egbirgin/tango) with its default algorithmic parameters [7,8,13].

All the experiments were run on a computer with Pentium IV processor, 512 Mb of RAM memory and Linux operating system. Codes are in Fortran77 and the compiler option “-O” was adopted.

7.1 A hidden-pattern example

We consider the application of LOVO described in Sect. 6. The points of $\mathcal{P}$, represented in Fig. 1a in light grey, are the 253 Cα atoms of the thyroid hormone receptor protein bound to a IH5, a synthetic ligand (Protein Data Bank identifier 1NAV). The points of $\mathcal{Q}$, in black in Fig. 1a, are 78 Cα atoms of the C-terminal region of a similar protein, albeit bound to a different ligand (PDB id. 1Q4X), which provides some structural differences. Therefore, there is no set of points in $\mathcal{P}$ which *exactly* match the set $\mathcal{Q}$. However, the proteins are similar. The goal here is to identify which set of points in the target protein best matches the points of the fragment. In other words, we aim to know whether there is a structural pattern of the type defined by $\mathcal{Q}$ in the structure defined by $\mathcal{P}$. This is the general definition of the problem of Protein Fold Recognition, which has fundamental importance for the analysis of protein function and evolution [17].

We used a multistart approach, since this type of problems may have many local minimizers. The variables of the problem are the ones that define the displacement D: three variables for defining the translation and three variables for defining rotations around the coordinate axes. Let $\mathcal{B} \subset I\!R^3$ be the smaller box that contains the protein $\mathcal{P}$. The initial approximation for the translation vector was taken as $\xi - O$ where O is the center of gravity of $\mathcal{Q}$ and ξ is a random point in B. The initial angles were taken uniformly randomly between 0 and 2π.

We ran the algorithm for 1,000 different initial points and obtained the best solution in 32 of these trials. In the best solution, the average distance between displaced points of $\mathcal{Q}$ and the points of $\mathcal{P}$ was 1.07 Å (the best solution found is correct from the point of view of protein function and is, very likely, the global solution). The execution of the unconstrained algorithm used an average of 16 iterations. The total time of execution was 15.5 s, which implies an average of 0.0155 s per execution of the algorithm. The use of designed initial points instead of random ones can greatly improve the tendency of the algorithm to converge to global minimizers, as was shown for the protein alignment problem [26].

In Fig. 1b we show the superposition of the points in the best solution found. We note that even when the alignment is good, its recognition is not obvious. Figure 1c shows the same solution, but now represented as a Cα trace (consecutive points in the structure are connected), and provides a clearer view of the alignment obtained (the fragment is in black and the target protein is in light grey).

7.2 Fitting models with outliers

7.2.1 Unconstrained fitting

Assume that $\{(t_1, y_1), \dots, (t_m, y_m)\} \subset I\!R^2$ is a set of data and we know that "some of them are wrong". Assume that $T(x, t_i)$ is the predicted value of the observation i with the parameters $x \in \Omega$. Least-squares fitting of the form $y_i = T(x, t_i)$ leads to unsatisfactory results due to the overwhelming influence of outliers.

The LOVO approach for robust estimation of parameters consists in defining, for each $i = 1, \dots, r$, the error function $F_i(x) = (T(x, t_i) - y_i)^2$. Given $p \in \{1, \dots, r\}$, this set of functions defines a LOVO problem. When $p = r$ this LOVO problem coincides with the classical nonlinear least-squares problem. However, the interesting situation is when p is smaller than r. In that case, the solution of LOVO allows one to discard the influence of an estimated number of outliers. If $p = r$ we expect a large value of the LOVO function in the solution. When p is decreased, the LOVO function at the solution tends to decrease as well.

To illustrate the behavior of the LOVO approach we consider a simple unconstrained problem where $T(x, t_i)$ is defined as

$$\begin{aligned} T(x, t_i) \;=\; & x_1 exp[-t_i x_5] + x_2 exp[-(t_i - x_9)^2 x_6] \;+\; x_3 exp[-(t_i - x_{10})^2 x_7] \\ & + x_4 exp[-(t_i - x_{11})^2 x_8]. \end{aligned}$$

This is the Osborne-2 function (coming from Problem 19 of [27], where $r = 65$). Here we introduced 13 additional data representing systematic errors.

As an initial approximation, we took the point indicated in [27]. For $p = 67$ and $p = 63$ the local solution obtained using this approximation was not satisfactory in the sense that the monotonicity condition $S_p(x_p) \le S_{p+1}(x_{p+1})$ was violated. In these two cases we tried 100 different random (100%) perturbations of the initial point and reported the best solution obtained in Table 1. The number of evaluations and computer time reported corresponds, in these two cases, to the successful execution.

The results are shown in Table 1 and Fig. 2.

Table 1 shows the behavior of the method for different values of p. FE means Function Evaluations and GE stands for Gradient Evaluations. Observe that, from $p = 65$ to $p = 66$ the functional value at the solution obtained increases four times. This is, by far, the biggest percentual increase between two consecutive values of p and supports the decision that 65 should be the "correct" p. Of course, this is as expected, since the remaining data were deliberately generated as outliers. In Fig. 2 one can appreciate that, for $p = 65$, the method identifies correctly the outliers and produces a satisfactory model for the correct data. For greater values of p, the fitted curved is distorted by the outliers.

Table 1 Unconstrained fitting example

p	Four first coordinates of x				$S_p(x)$	FE	GE	Iterations	CPU Time (s)
78	1.27	0.38	0.48	0.59	2.74	58	41	36	0.0574
77	1.32	0.56	0.61	0.60	2.39	56	39	37	0.0602
76	1.34	0.51	0.62	0.61	2.02	33	22	20	0.0555
75	1.33	0.46	0.61	0.62	1.76	29	22	20	0.0552
74	1.31	0.50	0.61	0.81	1.53	36	27	25	0.0587
73	1.32	0.39	0.60	0.80	1.32	30	20	18	0.0583
72	1.32	0.41	0.60	0.70	1.11	29	20	18	0.0547
71	1.32	0.37	0.61	0.68	0.94	30	20	16	0.0540
70	1.34	0.40	0.61	0.66	0.75	24	18	16	0.0558
69	1.34	0.41	0.61	0.61	0.61	22	17	15	0.0552
68	1.33	0.35	0.60	0.61	0.42	25	17	15	0.0543
67	1.34	0.60	0.35	0.56	0.28	115	54	52	0.0646
66	1.29	0.35	0.60	0.57	0.17	21	15	13	0.0580
65	1.31	0.43	0.63	0.60	0.04	37	23	21	0.0565
64	1.26	0.39	0.62	0.58	0.03	26	19	17	0.0576
63	1.26	0.38	0.61	0.58	0.03	28	20	17	0.0555

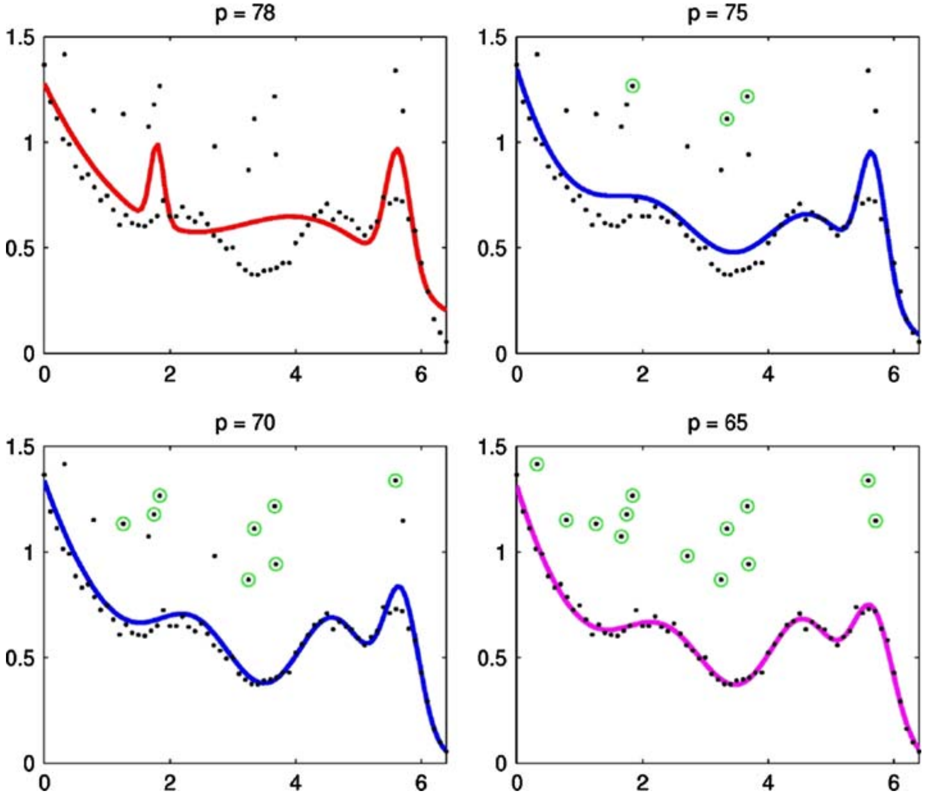

Fig. 2 Unconstrained model fitting

7.2.2 Constrained fitting

Assume that $x_1, \ldots, x_r$ satisfy the difference equations

$$\frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} = \Phi(t_i, x_i, z) \tag{31}$$

for $i = 2, \ldots, r - 1$, where $z \in I\!R^{npar}$ is a vector of unknown parameters, $h = 2/(r - 1)$, $t_i = (i - 1)h$. We want to find the correct values of x and the parameters z. The data of the problem are $y_1, \ldots, y_r$. Assume that approximately $r - p$ data are wrong. So, defining $F_i(x, z) = (x_i - y_i)^2$, the goal is to minimize $S_p(x, z)$ subject to the constraints (31).

In the experiments reported here we took $r = 21$, $npar = 3$ and

$$\Phi(x_i, z) = z_1 e^{x_i} - z_2(x_i^2 + 1)t_i - z_3 \sin(t_i x_i).$$

The data were generated as follows. First, we found the exact solution of (31) that satisfies $\bar{x}_1 = 4$, $\bar{x}_r = 6$ with $z_1 = 0.1$, $z_2 = 1$, $z_3 = 2$. Then, we chose $y_i = \bar{x}_i + \xi_i$, where ξ_i is random between -0.05 and 0.05, for $i = 4, \ldots, r - 2$. The data y_1, y_2, y_3, y_{r-1} and y_r were generated as outliers, much larger than the "correct" y_i (Fig. 3). As initial approximations for x_i we took 100 random (100%) perturbations of the data y_i. The initial approximations of z_1, z_2, z_3 were random between -10 and 10. The results reported in Table 2 correspond to the best solution obtained for each p. In this table, "Outer" means the number of (outer)

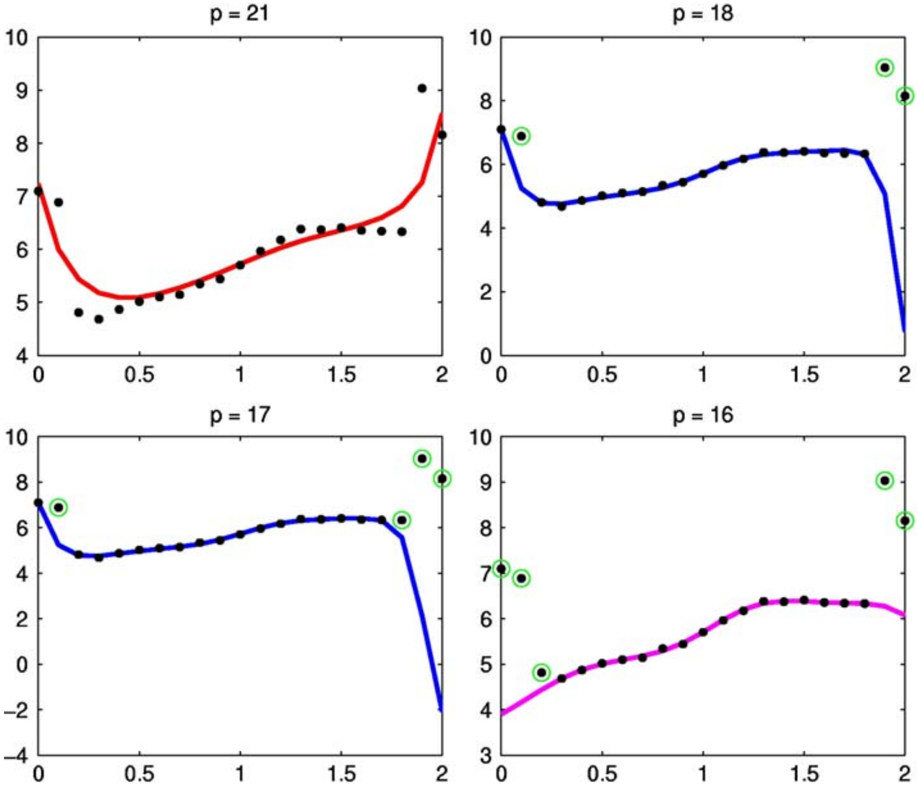

Fig. 3 Model fitting with constraints

Table 2 Constrained fitting example

p	x_1	x_r	z_1	z_2	z_3	$S_p(x)$	FE	GE	Inner	Outer	CPU Time (s)
21	7.2	8.6	0.2	1.7	0.4	5.267	121	81	74	6	0.167
20	7.4	8.1	0.3	2.6	1.1	1.709	77	53	45	6	0.166
19	45.7	8.2	4.1	38.9	31.4	0.127	341	318	312	2	0.251
18	7.1	0.7	1.0	9.3	7.3	0.039	96	66	59	3	0.104
17	7.1	−2.1	1.0	9.5	7.1	0.024	88	55	49	3	0.105
16	3.9	6.1	0.1	0.9	2.0	0.009	74	58	52	4	0.126
15	3.9	6.0	0.1	1.1	2.3	0.004	70	49	44	4	0.126
14	3.9	5.9	0.1	1.2	2.2	0.003	119	93	86	4	0.126

iterations performed by Algorithm **C-LOVO**. “Inner” means the number of (inner) iterations used by the subproblem solver at Step 2 of **C-LOVO**. The biggest percentual increase of $S_p(x)$ is detected from $p = 18$ to $p = 19$. In Fig. 3 we observe that, for $p = 16$ the fitted curve detects the outliers quite well and agrees with the correct model. However, for $p = 18$ the adjustment is still very good. (The only difference is due to the fact that for $p = 18$ the observations y_1, y_3 were not considered outliers.) In both cases, the correct data are very well reproduced. In this figure we also show, for completeness, the case $p = 17$ and the curve obtained for $p = 21$, which is obviously distorted by the presence of the outliers.

8 Conclusions

The LOVO problem defined in this paper is, in general, nonsmooth and nonconvex. Here we give (weak and strong) optimality conditions and introduce unconstrained and constrained algorithms for its resolution. An important consequence of the theory, confirmed by experiments, is that, unlike in most nonsmooth (even convex) problems, the consequences of ignoring nonsmoothness are not severe. Briefly speaking, smooth optimization algorithms, when applied to this problem, converge to weakly critical points and specific algorithms converge to strong critical points. This allows us to take advantage of the availability of efficient smooth optimization software.

The determination of the “correct” parameter p is very important in applications. At present, we advocate using the value of p that maximizes the percentage increase of $S_p(x_p)$, provided that this value is not unacceptably small, where x_p is the solution of the LOVO problems defined by p. However, the p-decision is quite case-sensitive.

The unconstrained algorithms introduced in this paper converge to different classes of stationary points and produce monotonically descent sequences, in the sense that the objective function decreases at every iteration. In practice, we did not observe differences in the behavior of these algorithms. Moreover, as subproblem solvers of the Augmented Lagrangian method, both produce the same theoretical results for constrained problems. This corroborates our point of view that the main difficulty of LOVO problems lies in the presence of many local minimizers. The development of powerful heuristics for overcoming this problem, finding initial points or escaping from local minimizers, will probably be subject of interesting research in the coming years.

Applications to Hidden Pattern recognition and to Robust Model fitting seem to be promising. Both problems are very important in many areas of Science and Engineering.

Undoubtedly, in the presence of specific technological applications it will be necessary to develop case-oriented algorithms but the possibility of using general software with reasonable results (an unusual feature in Engineering Optimization) is very encouraging.

Future research on this subject should include:

- Using DC-Programming [28,29] in the case that all the functions F_i involved in the LOVO problem are convex. In this case, the LOVO function has a natural decomposition as a difference of two convex functions:

$$S_p(x) = \sum_{i=1}^{r} F_i(x) - \sum_{j=p+1}^{r} F_{i_j(x)}(x)$$

  and practical improvements may be expected from DC-Algorithms.
- Exploiting smooth reformulations like the one proposed in [2] for the OVO problem.
- Adaptation and development of global-optimization strategies for finding suitable initial points to avoid the attractiveness of local-nonglobal minimizers.
- Development of constrained LOVO algorithms with convergence to strongly critical points.
- Extensions of the LOVO approach to the case in which p is not fixed in advance. This should enhance the applicability to similarity problems.
- Nonlinear programming problems with LOVO– and OVO–constraints.
- Sequential Quadratic Programming, Interior-Point and Restoration algorithms for nonlinearly constrained LOVO problems.
- Noisy Order-Value Optimization.

Acknowledgements We are indebted to Mario Gneri, for useful conversations on robust estimation problems and to an anonymous referee for many insightful comments. Research supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grants 06/53768-0, 05/56773-1 and 02-14203-6) and PRONEX CNPq/ FAPERJ 26/171.164/2003-APQ1. This paper was presented at the NCP07 Conference held in Rouen from December 17 to December 21, 2007, in the context of the minisymposium *Structured Optimization Problems and Methods*, organized by J. M. Martínez.

References

1. Andreani, R., Dunder, C., Martínez, J.M.: Order-Value Optimization: formulation and solution by means of a primal Cauchy method. Math. Methods Operat. Res. **58**, 387–399 (2003)
2. Andreani, R., Dunder, C., Martínez, J.M.: Nonlinear-Programming Reformulation of the Order-Value Optimization Problem. Math. Methods Operat. Res. **61**, 365–384 (2005)
3. Andreani, R., Martínez, J.M., Martínez, L., Yano, F.: Low Order-Value Optimization and Applications. Technical report MCDO 051013, Department of Applied Mathematics, State University of Campinas, Brazil, 2005
4. Andreani, R., Martínez, J.M., Schuverdt, M.L.: On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification. J. Optim. Theory Appl. **125**, 473–485 (2005)
5. Andreani, R., Martínez, J.M., Salvatierra, M., Yano, F.: Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy. In: Liberti, L., Maculan, N. (eds.) Global Optimization: Theory and Practice, pp. 379–404. Kluwer (2006)
6. Andreani, R., Martínez, J.M., Salvatierra, M., Yano, F.: Quasi-Newton methods for order-value optimization and value-at-risk calculations. Pac. J. Optim. **2**, 11–33 (2006)
7. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On Augmented Lagrangian methods with general lower-level constraints. SIAM J. Optim. **18**, 1286–1309 (2007)
8. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification. Math. Program. **112**, 5–32 (2008)

9. Andreani, R., Martínez, J.M., Martínez, L., Yano, F.: Continuous optimization methods for structural alignment. Math. Program. **112**, 93–124 (2008)
10. Audet, Ch., Dennis, J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**, 188–217 (2006)
11. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. Nucleic Acids Res. **28**, 235–242 (2000)
12. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
13. Birgin, E.G., Martínez, J.M.: Large-scale active-set box-constrained optimization method with spectral projected gradients. Comput. Optim. Appl. **23**, 101–125 (2002)
14. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints. Available in Optimization On-line, E-Print ID: 2006-12-1544, http://www.optimization-online.org/DB_HTML/2006/12/1544.html
15. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. **15**, 751–779 (2005)
16. Clarke, F.H.: Optimization and Nonsmooth Analysis. Classics in Applied Mathematics, SIAM, Philadelphia (1990)
17. Holm, L., Sander, C.: Mapping the Protein Universe. Science **273**, 595–602 (1996)
18. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
19. Jorion, P.: Value at Risk: The New Benchmark for Managing Financial Risk, 2nd edn. Mc Graw-Hill, New York (2001)
20. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallogr. A **34**, 827–828 (1978)
21. Kearsley, S.K.: On the orthogonal transformation used for structural comparisons. Acta Crystallogr. A **45**, 208–210 (1996)
22. Kolodny, R., Linial, N.: Approximate protein structural alignment in polynomial time. Proc. Natl. Acad. Sci. USA **101**, 12201–12206 (2004)
23. Kolodny, R., Koehl, P., Levitt, M.: Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. J. Mol. Biol. **346**, 1173–1188 (2005)
24. Mangasarian, O.L., Fromovitz, S.: The Fritz-John necessary optimality conditions in presence of equality and inequality constraints. J. Math. Anal. Appl. **17**, 37–47 (1967)
25. Martínez, J.M.: Order-Value Optimization and New Applications. Plenary Talk and the International Congress of Industrial and Applied Mathematics ICIAM 2007, Zurich (2007).
26. Martínez, L., Andreani, R., Martínez, J.M.: Convergent algorithms for Protein Structural Alignment. BMC Bioinform. **8**, 306 (2007)
27. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Software **7**, 17–41 (1981)
28. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to DC programming: theory, algorithm and applications. Acta Math. Vietnam. **22**, 289–355 (1997)
29. Pham Dinh, T., Le Thi, H.A.: DC Optimization Algorithms for solving the trust-region problem. SIAM J. Optim. **8**, 476–505 (1998)
30. Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to SQP methods. SIAM J. Optim. **10**, 963–981 (2000)
31. Rockafellar, R.T.: Lagrange multipliers and optimality. SIAM Rev. **35**, 183–238 (1993)
32. Rockafellar, R.T., Uryasev, S.: Conditional value-at-risk for general loss distributions. J. Bank. Finance **26**, 1443–1471 (2002)
33. Subbiah, S., Laurents, D.V., Levitt, M.: Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. Curr. Biol. **3**, 141–148 (1993)