Low Order-Value Optimization and applications *

R. Andreani † J. M. Martínez ‡ L. Martínez § F. Yano ¶

October 24, 2005/ January 4, 2007

Abstract

Given r real functions $F_1(x), \ldots, F_r(x)$ 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 $(y_1, \ldots, y_r)$ 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 at 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 our best succeeded applications include 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.

Key words: Order-Value Optimization, algorithms, convergence, robust estimation of parameters, hidden patterns.

1 Introduction

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

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

*Technical Report MCDO 051013, Department of Applied Mathematics, State University of Campinas, Brazil, October 2005. This work was 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.

†Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mail: andreani@ime.unicamp.br

‡Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mail: martinez@ime.unicamp.br

§Inst. Quim. State University of Campinas. E-mail: lmartinez@iqm.unicamp.br.

¶Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mail: yano@ime.unicamp.br

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

$$F_{i_1(x)}(x) \leq F_{i_2(x)}(x) \leq \ldots \leq F_{i_p(x)}(x) \leq \ldots \leq F_{i_r(x)}(x).$$

If the functions F_i are continuous, the function S_p is continuous as well, because it is a sum of continuous functions [3, 4]. 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 [3] 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 [4] 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 [26]. 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) [27] measurement with a confidence level (p/r) [7]. 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.

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

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

and

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

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

$$\text{Minimize } f_{min}(x) \text{ subject to } x \in \Omega. \tag{2}$$

Of course, the problem (2) is, at the same time, a particular case of (1), which corresponds to take $p = 1$ and $F_i(x) = f_i(x)$, $i = 1, \ldots, r$.

The characterization (2) of the LOVO problem will be used throughout this paper for theoretical purposes and for some relevant applications.

The High Order-Value function (that corresponds to CVaR) is $S^p(x) = f_{max}(x)$, where $f_{max}(x) = \max\{f_1(x), \ldots, f_m(x)\}$. 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 reduces to Linear Programming [39].

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 $y_1, \ldots, y_r \in I\!R$ 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 $i-$th 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 [8].

This paper is organized as follows. In Section 2 we define two types of optimality conditions for the LOVO problem. In Section 3 we define an algorithm for unconstrained LOVO problems, that converges to weakly critical points. In Section 4 we introduce a method that converges to strongly critical points. In both cases we prove local and global convergence. In Section 5 we introduce an algorithm for constrained LOVO problems and we prove its convergence. Hidden Pattern problems are reported in Section 6 and Protein Alignment are discussed in Section 7. Numerical examples are given in Section 8 and conclusions in Section 9.

Notation.

- The symbol $\|\cdot\|$ will denote the Euclidean norm of vectors and matrices, although many times it may be replaced by an arbitrary norm.
- $B(x_*, \epsilon) = \{x \in I\!R^n \mid \|x - x_*\| \leq \epsilon\}$.
- We denote $I\!N = \{0, 1, 2, \ldots\}$.
- We denote $I\!R_+ = \{t \in I\!R \mid t \geq 0\}$ 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 $i-$th component of the vector i. If there is no place to 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 (2).

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 (2) is, necessarily, a global minimizer of $f_i(x)$ for all $i \in I_{min}(x_*)$. As a consequence, 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 (2) 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 $I_{min}(x_*)$,

$$f_{min}(y) \leq f_i(y) < f_i(x_*) = f_{min}(x_*).$$

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 (2) 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

$$A = \{x \in \Omega \mid \|x - x_*\| \leq \epsilon\}.$$

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 (2).*

Proof. Let $\epsilon > 0$ be such that

$$f_i(x_*) > f_{min}(x_*) + \epsilon \text{ for all } i \notin I_{min}(x_*).$$

Since f_i is continuous for all $i \notin I_{min}(x_*)$, there exists $\delta_1 > 0$ such that

$$f_i(x) > f_{min}(x_*) \text{ for all } i \notin I_{min}(x_*) \tag{3}$$

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

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

$$f_i(x) \geq f_i(x_*) = f_{min}(x_*) \tag{4}$$

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

Define $\delta = \min\{\delta_1, \delta_2\}$. By (3) and (4), we have that, for all $x \in \Omega$ such that $\|x - x_*\| \leq \delta$, and for all $i = 1, \ldots, m$,

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

Therefore,

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

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

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

$$\text{Minimize } \Phi(x) \text{ subject to } x \in \Omega. \tag{5}$$

Necessary Optimality Conditions (NOC) are conditions that must be satisfied by local minimizers of (5). 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 [13]. 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 (2), 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*

$$\textit{Minimize } f_i(x) \textit{ subject to } x \in \Omega. \tag{6}$$

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 Necessary Optimality Condition (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 (6).

We say that $x_* \in \Omega$ is *weakly critical* if there exists $i \in I_{min}(x_*)$ such that x_* satisfies NOC, associated with (6).

3 Unconstrained LOVO algorithm with convergence to weakly critical points

Optimization algorithms for solving nonlinear programming problems (5) 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 (2) 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"

$$\nabla f_{min}(x) \leftarrow \nabla f_i(x). \tag{7}$$

(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 it 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)-(2), 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], Section 2.5.1). The reciprocal is not true. For example, 0 is a Clarke-stationary point of the problem

$$\text{Minimize } \min\{x, -x/2\},$$

but it is not a weakly critical point.

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" (7).

Algorithm U1. Let $\theta \in (0,1), \alpha \in (0,1), M > 1, \beta > 0, t_{one} > 0$ 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

$$\nabla f_{\nu(k)}(x_k)^T d_k \leq -\theta \|d_k\| \|\nabla f_{\nu(k)}(x_k)\| \quad \text{and} \quad \|d_k\| \geq \beta \|\nabla f_{\nu(k)}(x_k)\|. \tag{8}$$

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

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

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 $t_{one} = 1$ and $M = 2$. However, the choice based on (9)-(10) admits more sophisticated and efficient line-search procedures. See, for example, [14].

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.

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

Proof. Assume that x_k is not weakly critical and define $i = \nu(k)$. So, $\nabla f_i(x_k) \neq 0$. By (8) and the differentiability of f_i,

$$\lim_{t \to 0} \frac{f_i(x_k + t d_k) - f_i(x_k)}{t} = \nabla f_i(x_k)^T d_k < 0.$$

Then,

$$\lim_{t \to 0} \frac{f_i(x_k + t d_k) - f_i(x_k)}{t \nabla f_i(x_k)^T d_k} = 1.$$

Since $\alpha < 1$, for t small enough we have:

$$\frac{f_i(x_k + t d_k) - f_i(x_k)}{t \nabla f_i(x_k)^T d_k} \geq \alpha.$$

Since $\nabla f_i(x_k)^T d_k < 0$, we deduce:

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

But $f_{min}(x_k + t d_k) \leq f_i(x_k + t d_k)$ and $f_{min}(x_k) = f_i(x_k)$, so:

$$f_{min}(x_k + t d_k) \leq f_{min}(x_k) + \alpha t \nabla f_i(x_k)^T d_k \tag{11}$$

for t small enough.

Therefore, choosing t_k as the first number in the sequence $\{t_{one}, t_{one}/M, t_{one}/M^2, \ldots\}$ that satisfies (11), the conditions (9) and (10) are satisfied.

This proves that, whenever x_k is not weakly critical, a point x_{k+1} satisfying (9)-(10) may be found, so the algorithm is well defined. □

Let us remark that Theorem 3.1 says that, if Algorithm **U1** terminates at x_k, then x_k is weakly critical, but the reciprocal is not true. For example, define, with $n = 1, m = 2$, $f_1(x) = x$, $f_2(x) = x^2$. Clearly, 0 is weakly critical because $\nabla f_2(0) = 0$. However, if $x_k = 0$ and one chooses $\nu(k) = 1$ the algorithm will not stop and, in fact, it will find a better point such that $f_{min}(x) < f_{min}(0)$.

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,*

$$\lim_{k\in K} \|\nabla f_{\nu(k)}(x_k)\| = 0. \tag{12}$$

Proof. Let $x_* \in I\!R^n$ be a limit point of the sequence generated by Algorithm **U1**. Let $K = \{k_0, k_1, k_2, k_3, \ldots\}$ be an infinite sequence of integers such that:

1. There exists $i \in \{1, \ldots, m\}$ such that $i = \nu(k)$ for all $k \in K$.
2. $\lim_{k\in K} x_k = x_*$.

The sequence K and the index i necessarily exist since $\{1, \ldots, m\}$ is finite.

By the continuity of f_i,

$$\lim_{k\in K} f_i(x_k) = f_i(x_*). \tag{13}$$

Clearly, since $i = \nu(k)$, we have that

$$f_i(x_k) \leq f_\ell(x_k) \text{ for all } \ell \in \{1, \ldots, m\}.$$

for all $k \in K$.

Taking limits on both sides of this inequality, we see that $f_i(x_*) \leq f_\ell(x_*)$ for all $\ell \in \{1, \ldots, m\}$. Thus,

$$i \in I_{min}(x_*). \tag{14}$$

By the definition of Algorithm **U1**, since $k_{j+1} \geq k_j + 1$, we have:

$$f_i(x_{k_{j+1}})$$

$$= f_{min}(x_{k_{j+1}}) \leq f_{min}(x_{k_j+1}) \leq f_{min}(x_{k_j}) + \alpha t_{k_j} \nabla f_i(x_{k_j})^T d_{k_j} < f_{min}(x_{k_j}) = f_i(x_{k_j}) \tag{15}$$

for all $j \in I\!N$.

By (9), (13) and (15), we obtain:

$$\lim_{j\to\infty} t_{k_j} \nabla f_i(x_{k_j})^T d_{k_j} = 0.$$

Therefore, by (8),

$$\lim_{j\to\infty} t_{k_j} \|\nabla f_i(x_{k_j})\| \|d_{k_j}\| = 0. \tag{16}$$

If, for some subsequence $K_1 \subset K$, $\lim_{k\in K_1} \nabla f_i(x_k) = 0$, we deduce that $\nabla f_i(x_*) = 0$ and the thesis is proved. Therefore, we only need to analyze the possibility that $\|\nabla f_i(x_k)\|$ is bounded away from zero for $k \in K$. In this case, by (16),

$$\lim_{k\in K} t_k \|d_k\| = 0. \tag{17}$$

If, for some subsequence, $\|d_k\| \to 0$, the condition (8) also implies that $\nabla f_i(x_k) \to 0$ and $\nabla f_i(x_*) = 0$. Thus, we only need to consider the case in which $\lim_{k\in K} t_k = 0$. Without loss of generality, we may assume that $t_k < t_{one}$ for all $k \in K$. So, by (10), for all $k \in K$ there exists $\bar{t}_k > 0$ such that

$$f_i(x_k + \bar{t}_k d_k) \geq f_{min}(x_k + \bar{t}_k d_k) > f_{min}(x_k) + \alpha \bar{t}_k \nabla f_i(x_k)^T d_k = f_i(x_k) + \alpha \bar{t}_k \nabla f_i(x_k)^T d_k. \quad (18)$$

Moreover, by (10) and (17),

$$\lim_{k\in K} \bar{t}_k \|d_k\| = 0. \quad (19)$$

Define $s_k = \bar{t}_k d_k$ for all $k \in K$. Then, by (19),

$$\lim_{k\in K} \|s_k\| = 0. \quad (20)$$

By (18) and the Mean Value Theorem, for all $k \in K$ there exists $\xi_k \in [0, 1]$ such that

$$\nabla f_i(x_k + \xi_k s_k)^T s_k = f_i(x_k + s_k) - f_i(x_k) > \alpha \nabla f_i(x_k)^T s_k. \quad (21)$$

Moreover, by (8),

$$\frac{\nabla f_i(x_k)^T s_k}{\|s_k\|} \leq -\theta \|\nabla f_i(x_k)\| \quad (22)$$

for all $k \in K$.

Let $K_1 \subset K$, $s \in I\!R^n$ be such that $\lim_{k\in K_1} s_k/\|s_k\| = s$.

By (20), dividing both sides of the inequality (21) by $\|s_k\|$, and taking limits for $k \in K_1$, we obtain:

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

Since $\alpha < 1$ and $\nabla f_i(x_k)^T d_k < 0$ for all k, this implies that $\nabla f_i(x_*)^T s = 0$. Thus, taking limits in (22), we obtain that $\nabla f_i(x_*) = 0$. Therefore, by (14), x_* is weakly critical.

Finally, let us prove (12). If (12) is not true, there exists j and an infinite set of indices $k \in K$ such that $j = \nu(k)$ and $\|\nabla f_j(x_k)\|$ is bounded away from zero. This implies that $j \in I_{min}(x_*)$ and $\|\nabla f_j(x_*)\| \neq 0$, contradicting the first part of the proof. □

In the rest of this section we address the local convergence of Algorithm **U1**. The choice of x_{k+1} in this algorithm imposes that $f_{min}(x_{k+1}) \leq f_{min}(x_k) + \alpha t_k \nabla f_{\nu(k)}(x_k)^T d_k$. 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. This requirement is stated in the following Assumption **B1**.

Assumption B1

We assume that Algorithm **U1** is implemented in such a way that there exists $b > 0$ such that

$$\|x_{k+1} - x_k\| \leq b \|\nabla f_{\nu(k)}(x_k)\| \quad (23)$$

for all $k \in I\!N$.

Assumption **B1** is compatible with line searches based on (10). For gradient, Newton or quasi-Newton choices of d_k one generally has that $\|d_k\| = O(\|\nabla f_{\nu(k)}(x_k)\|)$. Obviously, backtracking preserves this property with $t_k d_k$ replacing d_k. So, a point x_{k+1} of the form $x_k + t_k d_k$ and satisfying (23) may be obtained.

Our strategy for proving local superlinear convergence has three parts. In Theorem 3.3 we show that, under an isolation assumption, if x_* is a limit point of the algorithm, the whole sequence converges to it. In Theorem 3.4 we prove that, if the algorithm is started near a strict local minimizer, the generated sequence converges. Neither Theorem 3.4 can be reduced to Theorem 3.3, nor Theorem 3.3 is a consequence of Theorem 3.4 (the assumption on x_* of Theorem 3.3 is weaker). However, both theorems show that convergence of the whole sequence to a point x_* may be expected in many cases. Under this assumption and assuming that the search directions are obtained as the inexact solutions of quasi-Newton linear systems with a Dennis-Moré compatibility condition we will show that superlinear convergence takes place.

We say that x_* is *very strongly isolated* if there exists $\epsilon > 0$ such that for all $x \in B(x_*, \epsilon) - \{x_*\}$ and for all $i \in I_{min}(x)$, we have that $\nabla f_i(x) \neq 0$. In other words, a reduced neighborhood of x_* does not contain weakly critical points.

Theorem 3.3. *Assume that x_* is very strongly isolated, the sequence $\{x_k\}$ is generated by Algorithm* **U1** *with Assumption* **B1** *and $\lim_{k\in K} x_k = x_*$ for some infinite sequence $K \subset \mathbb{N}$. Then, x_* is weakly critical and*

$$\lim_{k\to\infty} x_k = x_*.$$

Proof. The fact that x_* is weakly critical is a consequence of Theorem 3.2.

By (12) and (23), we have:

$$\lim_{k\in K} \|x_{k+1} - x_k\| = 0. \tag{24}$$

Since x_* is very strongly isolated, there exists $\epsilon > 0$ such that $\nabla f_i(x) \neq 0$ for all $i \in I_{min}(x)$ if $x \in B(x_*, \epsilon) - \{x_*\}$.

By (24) and the hypothesis of the theorem, there exists $k_1 \in K$ such that

$$\|x_{k+1} - x_k\| < \epsilon/2 \text{ and } \|x_k - x_*\| < \epsilon/2$$

for all $k \in K, k \geq k_1$.

Define

$$C = \{x \in \mathbb{R}^n \mid \epsilon/2 \leq \|x - x_*\| \leq \epsilon\}.$$

Clearly, C is compact and does not contain weakly critical points. Then, by Theorem 3.2, C cannot contain infinitely many iterates. Therefore, we have two possibilities:

1. There exists $k_2 \in \mathbb{N}$ such that $\|x_k - x_*\| \leq \epsilon/2$ for all $k \geq k_2$.
2. There exist infinitely many iterates $k \geq k_1$, such that $\|x_k - x_*\| \leq \epsilon/2$ and $\|x_{k+1} - x_k\| > \epsilon/2$.

In the first case, since x_* is the only possible limit point in the ball with radius $\epsilon/2$ we have that the sequence $\{x_k\}$ converges to x_*.

Let us analyze the second case. Let $K_1 \subset I\!N$ be such that $\|x_k - x_*\| \leq \epsilon/2$ and $\|x_{k+1} - x_k\| > \epsilon/2$ for all $k \in K_1$.

Since all the iterates belong to the ball with center $\epsilon/2$ and x_* is the only possible limit point in this ball, it turns out that

$$\lim_{k \in K_1} x_k = x_*.$$

Therefore, by (12),

$$\lim_{k \in K_1} \|\nabla f_{\nu(k)}(x_k)\| = 0.$$

By (23), this implies that

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0,$$

contradicting the assumption $\|x_{k+1} - x_k\| \geq \epsilon/2 \ \forall \ k \in K_1$. This means that the second case mentioned above is impossible. So, the proof is complete. □

Theorem 3.4. *Assume that x_* is a very strongly isolated strict local minimizer of f_{min}. Let $\{x_k\}$ be a sequence generated by Algorithm* **U1** *with Assumption* **B1**. *Then, there exists $\delta_1 > 0$ such that $\|x_0 - x_*\| \leq \delta_1$ implies that*

$$\lim_{k \to \infty} x_k = x_*.$$

Proof. Let $\epsilon > 0$ be such that x_* is a strict global minimizer of f_{min} in the ball $B(x_*, \epsilon)$ and that this ball does not contain weakly critical points other than x_*. Let us prove that there exists $\delta \in (0, \epsilon/2)$ such that

$$\|x_k - x_*\| \leq \delta \Rightarrow \|x_{k+1} - x_k\| \leq \epsilon/2. \tag{25}$$

Assume, by contradiction, that δ satisfying (25) does not exist. Given $x \in I\!R^n$ denote x_+ the possible *follower* of x by an iteration of Algorithm **U1**. Under the assumption that (25) is false, there exists a sequence $\{z_\ell\}$ such that $\lim_{\ell \to \infty} z_\ell = x_*$ and

$$\|z_\ell - (z_\ell)_+\| > \epsilon/2 \ \text{ for all } \ell = 0, 1, 2, \ldots$$

By (23) this implies that for all $\ell \in I\!N$, there exists j_ℓ such that

$$f_{j_\ell}(z_\ell) = f_{min}(z_\ell)$$

and $\|\nabla f_{j_\ell}(z_\ell)\|$ is bounded away from zero. Take j such that $j = j_\ell$ infinitely many times. Then, $j \in I_{min}(z_\ell)$ for all ℓ and $\|\nabla f_j(z_\ell)\|$ is bounded away from zero. This implies that $j \in I_{min}(x_*)$ and $\|\nabla f_j(x_*)\| \neq 0$. This cannot be true, since x_* is a local minimizer and, hence, it is strongly critical. Therefore, (25) is true.

Let c be the minimum of $f_{min}(x)$ on the set defined by $\delta \leq \|x - x_*\| \leq \epsilon$. Let $\delta_1 \in (0, \delta)$ be such that

$$\|x - x_*\| \leq \delta_1 \Rightarrow f_{min}(x) < c.$$

Let us prove by induction that, taking $\|x_0 - x_*\| \leq \delta_1$, one has that $\|x_k - x_*\| \leq \epsilon/2$ and $f(x_k) < c$ for all k. By the definition of δ_1 this is true for $k = 0$. For the inductive step, observe that, by (25), we have that $\|x_{k+1} - x_*\| \leq \epsilon$. But, by the definition of c and the fact that $f(x_{k+1}) < f(x_k)$, we have that $\|x_{k+1} - x_*\| \leq \epsilon/2$.

Therefore, the whole sequence is contained in $B(x_*, \epsilon/2)$. Since the only weakly critical point in this ball is x_*, Theorem 3.2 implies that the whole sequence converges to x_* as we wanted to prove. □

Assumption B2. In the implementation of Algorithm **U1** we have:

- $\alpha \in \left(0, \frac{1}{2}\right)$.
- The direction d_k is a solution of

$$B_k d = -\nabla f_{\nu(k)}(x_k) + r_k, \tag{26}$$

  where $B_k \in I\!R^{n\times n}$ is symmetric and positive definite and

$$\|r_k\| \leq \eta_k \|\nabla f_{\nu(k)}(x_k)\|. \tag{27}$$

- If

$$f_{min}(x_k + d_k) \leq f_{min}(x_k) + \alpha \nabla f_{\nu(k)}(x_k)^T d_k,$$

  we choose $t_k = 1$ and $x_{k+1} = x_k + d_k$.
- The set $\{\|B_k^{-1}\|, k \in I\!N\}$ is bounded.

Let us comment here some features of Algorithm **U1** under Assumption **B2**.

1. The coefficient α is restricted to $(0, 1/2)$ because this favors the acceptance of the steplength $t_k = 1$, as will be shown in the proofs.
2. The direction d_k comes from the inexact solution of a quasi-Newton equation. The matrices B_k will be positive-definite Hessian approximations.
3. When $x_k + d_k$ satisfies the sufficient descent condition we accept the steplength $t_k = 1$ and the point x_{k+1} is taken as $x_k + d_k$. Again, this enhances the probability of taking Newton-like steps.
4. If $\|B_k\| < \frac{1}{\beta}$, the condition $\|d_k\| \geq \beta \|\nabla f_{\nu(k)}(x_k)\|$ is satisfied. Moreover, if the condition number $\|B_k\|\|B_k^{-1}\|$ is less than or equal to $\frac{1}{\theta}$, the angle condition (8) is satisfied. Clearly, it is always possible to choose B_k satisfying both requirements.

Theorem 3.5 completes the convergence theory of Algorithm **U1**. We will show that, under Assumptions **B1** and **B2**, if the sequence $\{x_k\}$ converges to a local minimizer such that all the relevant Hessians are positive definite and the matrices B_k satisfy a Dennis-Moré condition, the convergence is superlinear and, eventually, $t_k = 1$.

Theorem 3.5. *Assume that:*

1. *The sequence* $\{x_k\}$ *is generated by Algorithm* **U1** *with Assumptions* **B1** *and* **B2***;*

2. x_* *is a local minimizer;*

3. f_i *admits continuous second derivatives in a neighborhood of* x_* *for all* $i \in I_{min}(x_*)$*;*

4. $\nabla^2 f_i(x_*) > 0$ *for all* $i \in I_{min}(x_*)$*;*

5. $\lim_{k\to\infty} x_k = x_*$*;*

6. *The Dennis-Moré condition*

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f_{\nu(k)}(x_k)]d_k\|}{\|d_k\|} = 0 \tag{28}$$

and the Inexact-Newton condition

$$\lim_{k\to\infty} \eta_k = 0 \tag{29}$$

are verified.

Then,

- *There exists* $k_0 \in I\!N$ *such that* $t_k = 1$ *for all* $k \geq k_0$.
- *The sequence* $\{x_k\}$ *converges superlinearly to* x_*.

Proof. By the continuity of the functions f_i, there exists $k_1 \in I\!N$ such that, for all $k \geq k_1$,

$$I_{min}(x_k) \subset I_{min}(x_*).$$

By Taylor's formula, for all $k \geq k_1$, we have that

$$f_{\nu(k)}(x_k+d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k) = (1-\alpha) d_k^T \nabla f_{\nu(k)}(x_k) + \frac{1}{2} d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2)$$

$$= (1-\alpha) d_k^T [\nabla f_{\nu(k)}(x_k) + \nabla^2 f_{\nu(k)}(x_k) d_k] + \left(\alpha - \frac{1}{2}\right) d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2).$$

But $B_k d_k + \nabla f_{\nu(k)}(x_k) = r_k$ and, by (8), (27) and (29), $\|r_k\| = o(\|\nabla f_{\nu(k)}(x_k)\|) = o(\|d_k\|)$. Therefore,

$$f_{\nu(k)}(x_k + d_k) - f_{\nu(k)}(x_k) - \alpha d_k^T \nabla f_{\nu(k)}(x_k)$$

$$= (1-\alpha)(d_k)^T r_k + (1-\alpha) d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k] d_k + \left(\alpha - \frac{1}{2}\right) d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2)$$

$$= (1-\alpha) d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k] d_k + \left(\alpha - \frac{1}{2}\right) d_k^T \nabla^2 f_{\nu(k)}(x_k) d_k + o(\|d_k\|^2).$$

But, by (28),

$$(1-\alpha) d_k^T [\nabla^2 f_{\nu(k)}(x_k) - B_k] d_k = o(\|d_k\|^2),$$

therefore,

$$f_{\nu(k)}(x_k+d_k)-f_{\nu(k)}(x_k)-\alpha d_k^T\nabla f_{\nu(k)}(x_k)=\left(\alpha-\frac{1}{2}\right)d_k^T\nabla^2 f_{\nu(k)}(x_k)d_k+o(\|d_k\|^2). \tag{30}$$

Let $\mu>0$ denote a lower bound for the eigenvalues of $\nabla^2 f_i(x_*), i\in I_{min}(x_*)$. Then, there exists $k_2>k_1$ such that $\mu/2$ is lower bound for the eigenvalues of $\nabla^2 f_{\nu(k)}(x_k)$ for all $k\geq k_2$. So, for all $k\geq k_2$, we have:

$$\frac{d_k^T\nabla^2 f_{\nu(k)}(x_k)d_k}{\|d_k\|^2}\geq \mu/2.$$

Since $\alpha<1/2$, by (30), we have:

$$\frac{f_{\nu(k)}(x_k+d_k)-f_{\nu(k)}(x_k)-\alpha d_k^T\nabla f_{\nu(k)}(x_k)}{\|d_k\|^2}\leq\left(\alpha-\frac{1}{2}\right)\frac{\mu}{2}+\frac{o(\|d_k\|^2)}{\|d_k\|^2} \tag{31}$$

for $k\geq k_2$. But, since $\{\|B_k^{-1}\|, k\in I\!N\}$ is bounded and $\nabla f_{\nu(k)}(x_k)\to 0$, by (26) and (27) we have that $\|d_k\|\to 0$. So, taking limits in (31) for $k\to\infty$, we get:

$$f_{\nu(k)}(x_k+d_k)-f_{\nu(k)}(x_k)-\alpha d_k^T\nabla f_{\nu(k)}(x_k)\leq 0 \tag{32}$$

for k large enough. So, by the definition of the algorithm, there exists $k_0\in I\!N$ such that $t_k=1$ for all $k\geq k_0$. Therefore, the first part of the thesis is proved.

By the first part of the thesis and Assumption **B2** we have that

$$x_{k+1}-x_k=d_k \quad \text{for all} \quad k\geq k_0.$$

Then, by Taylor's formula:

$$\begin{aligned}\nabla f_{\nu(k)}(x_{k+1})&=\nabla f_{\nu(k)}(x_k)+\nabla^2 f_{\nu(k)}(x_k)d_k+o(\|d_k\|)\\ &=B_k d_k+\nabla f_{\nu(k)}(x_k)+[\nabla^2 f_{\nu(k)}(x_k)-B_k]d_k+o(\|d_k\|)\\ &=r_k+[\nabla^2 f_{\nu(k)}(x_k)-B_k]d_k+o(\|d_k\|).\end{aligned}$$

As in the first part of the proof, we have that $\|r_k\|=o(\|d_k\|)$, therefore:

$$\nabla f_{\nu(k)}(x_{k+1})=[\nabla^2 f_{\nu(k)}(x_k)-B_k]d_k+o(\|d_k\|).$$

So, by (28),

$$\lim_{k\to\infty}\frac{\|\nabla f_{\nu(k)}(x_{k+1})\|}{\|x_{k+1}-x_k\|}=0.$$

By the continuity and nonsingularity of the Hessians at x_*, we deduce that

$$\lim_{k\to\infty}\frac{\|x_{k+1}-x_*\|}{\|x_{k+1}-x_k\|}=0.$$

Clearly, this implies that

$$\lim_{k\to\infty}\frac{\|x_{k+1}-x_*\|}{\|x_{k+1}-x_*\|+\|x_k-x_*\|}=0.$$

Therefore, after some manipulation, we obtain the superlinear convergence of $\{x_k\}$. □

4 Unconstrained LOVO algorithm with convergence to strongly critical points

In Section 3 we introduced Algorithm **U1** which, briefly speaking, converges to weakly critical points. Algorithm **U1** may converge to points that are not strongly critical and, of course, that are far from being minimizers of the unconstrained LOVO problem. 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.

Algorithm U2. Let $\theta \in (0, 1), \alpha \in (0, 1), M > 1, \beta > 0, t_{one} > 0, \varepsilon > 0, \delta > 0$ 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

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

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

$$\nabla f_i(x_k)^T d_k^i \leq -\theta \|d_k^i\| \|\nabla f_i(x_k)\| \quad \text{and} \quad \|d_k^i\| \geq \beta \|\nabla f_i(x_k)\|. \tag{33}$$

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

$$f_i(x_k + t_k^i d_k^i) \leq f_i(x_k) + \alpha t_k^i \nabla f_i(x_k)^T d_k^i \tag{34}$$

and

$$\left[t_k^i \geq t_{one}\right] \quad \text{or} \quad \left[f_i(x_k + \bar{t}_k d_k^i) > f_i(x_k) + \alpha \bar{t}_k^i \nabla f_i(x_k)^T d_k^i \text{ for some } \bar{t}_k^i \leq M t_k^i\right]. \tag{35}$$

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

$$f_{min}(x_{k+1}) \leq \min_{i \in J_k} \{f_i(x_k + t_k^i d_k^i)\}. \tag{36}$$

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 $f_i(x_k) \approx f_{min}(x_k)$ (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

$$f_j(x_k + t_k^j d_k^j) \leq f_i(x^k + t_k^i d_k^i) \; \forall \; i \in J_k.$$

However, the choice (36) allows one to use extrapolation steps to enhance the chance of convergence to global minimizers.

In the worst case situation, J_k may contain an unacceptably large number of indices (for example, if $f_i(x^k) = f_{min}(x^k)$ for all i. In practice, it is recommendable to limit the number of search directions to (say) 10, and switch to Algorithm **U1** if this number is exceeded. We tried this modification in our numerical examples without obtaining meaningful differences with the non-modified algorithm.

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,*

$$f_{min}(x_{k+1}) < f_{min}(x_k) \tag{37}$$

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 (37).

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

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

is verified. Therefore, choosing t_k^i as the first number in the sequence $\{t_{one}, t_{one}/M, t_{one}/M^2, \ldots\}$ that satisfies (34), the conditions (34) and (35) 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, (37) follows from (36). □

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

$$\|\nabla f_i(x_k)\| > \delta \text{ for all } i \in I_{min}(x_k), k \in K_1. \tag{38}$$

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 (38) 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 (38) 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*

$$\|\nabla f_i(x_k)\| \leq \epsilon \ \text{ for all } \ i \in I_{min}(x_k).$$

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

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

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},

$$f_i(x_k) \leq f_{min}(x_k) + \varepsilon. \tag{39}$$

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 $\nabla f_i(x_k) \neq 0$ for all $k \in K$. Therefore, by (39), $i \in J_k$ for all $k \in K$. Moreover,

$$\lim_{k\in K} f_i(x_k) - f_{min}(x_k) = f_i(x_*) - f_{min}(x_*) = 0. \tag{40}$$

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

$$f_{min}(x_{k_{j+1}}) < f_{min}(x_{k_j+1}) \leq f_i(x_{k_j} + t^i_{k_j} d^i_{k_j}) \leq f_i(x_{k_j}) + \alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j}$$

$$= f_{min}(x_{k_j}) + [f_i(x_{k_j}) - f_{min}(x_{k_j})] + \alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j}. \tag{41}$$

By (33), $\alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j} < 0$. Assume, for a moment, that there exists $c > 0$, $j_0 \in I\!N$, such that

$$\alpha t^i_{k_j} \nabla f_i(x_{k_j})^T d^i_{k_j} < -c \tag{42}$$

for all $j \geq j_0$. But, by (40), there exists $j_1 \geq j_0$ such that

$$f_i(x_{k_j}) - f_{min}(x_{k_j}) < c/2 \tag{43}$$

for all $j \geq j_1$. So, by (41), (42) and (43), we have that

$$f_{min}(x_{k_{j+1}}) \leq f_{min}(x_{k_j}) - c/2$$

for all $j \geq j_1$. 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 (42) is impossible. This implies that there exists K_1, an infinite subsequence of K, such that

$$\lim_{k\in K_1} \alpha t_k^i \nabla f_i(x_k)^T d_k^i = 0.$$

Therefore, by (33),

$$\lim_{k\in K_1} t_k^i \|\nabla f_i(x_k)\| \|d_k^i\| = 0.$$

The rest of the proof is similar to the proof of Theorem 3.2. 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,

$$\lim_{k\in K} t_k^i \|d_k^i\| = 0. \tag{44}$$

If, for some subsequence $K_3 \subset K_1$, $\lim_{k\in K_3} \|d_k^i\| = 0$, then, by (33), $\lim_{k\in K_3} \|\nabla f_i(x_k)\| = 0$ 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 (44),

$$\lim_{k\in K_1} t_k^i = 0.$$

Therefore, without loss of generality, we may assume that $t_k^i < t_{one}$ for all $k \in K_1$. Then, by (35), there exist $\bar{t}_k^i \leq M t_k^i$, $s_k = \bar{t}_k^i d_k^i$ such that

$$f_i(x_k + s_k) > f_i(x_k) + \alpha \nabla f_i(x_k)^T s_k \quad \text{for all} \quad k \in K_1 \tag{45}$$

and, by (44),

$$\lim_{k\in K_1} \|s_k\| = 0. \tag{46}$$

So, by (45) and the Mean Value Theorem, there exists $\xi_k \in [0, 1]$ such that

$$\nabla f_i(x_k + \xi_k s_k)^T s_k = f_i(x_k + s_k) - f_i(x_k) > \alpha \nabla f_i(x_k)^T s_k \tag{47}$$

for all $k \in K_1$. Moreover, by (33),

$$\frac{\nabla f_i(x_k)^T s_k}{\|s_k\|} \leq -\theta \|\nabla f_i(x_k)\| \tag{48}$$

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 (46), dividing both sides of (47) by $\|s_k\|$ and taking limits for $k \in K_4$, we obtain:

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

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 (48) 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 get that $i \in I_{min}(x_*)$ and $\|\nabla f_i(x_*)\| \geq \epsilon$. This contradicts the first part of the thesis. □

For proving local convergence convergence we follow similar steps to those of Algorithm **U1**. Assumption **B3** establishes that the distance between two consecutive iterates is less than or equal to the maximum gradient norm in J_k. This is always possible if the directions d_k^i are taken according to gradient-like, Newton or quasi-Newton paradigms.

Assumption B3

We assume that Algorithm **U2** is implemented in such a way that there exists $b > 0$ such that

$$\|x_{k+1} - x_k\| \leq b \max\{\|\nabla f_i(x_k)\|, i \in J_k\} \tag{49}$$

for all $k \in I\!N$.

We say that x_* is *strongly isolated* if there exists $\epsilon > 0$ such that for all $x \in B(x_*, \epsilon) - \{x_*\}$ there exists $i \in I_{min}(x)$ such that $\nabla f_i(x) \neq 0$. In other words, a reduced neighborhood of x_* does not contain strongly critical points.

Let $a > 0$. We say that x_* is *a-vertically isolated* if $f_i(x_*) > f_{min}(x_*) + a$ for all $i \notin I_{min}(x_*)$.

Theorem 4.3 is similar to Theorem 3.3 of Section 3. We prove that, under strongly isolation and vertical isolation assumptions, a limit point of the sequence generated by Algorithm **U2** is necessarily the limit of the whole sequence. Moreover, in Theorem 4.4 we show that convergence to a strict local minimizer occurs if the initial point is close enough to such a solution.

Theorem 4.3. *Assume that x_* is strongly isolated and a-vertically isolated with $a > \varepsilon$. Suppose that the sequence $\{x_k\}$ is generated by Algorithm* **U2** *with Assumption* **B3** *and $\lim_{k \in K} x_k = x_*$ for some infinite sequence $K \subset I\!N$. Then, x_* is strongly critical and*

$$\lim_{k \to \infty} x_k = x_*.$$

Proof. The fact that x_* is strongly critical is a consequence of Theorem 4.2.

By the assumption of vertical isolation, for $k \in K$ large enough we have that $J_k \subset I_{min}(x_*)$. Since $\nabla f_i(x_*) = 0$ for all $i \in I_{min}(x_*)$, by (49) we have that

$$\lim_{k \in K} \|x_{k+1} - x_k\| = 0. \tag{50}$$

By (50) and the hypothesis of the theorem, there exists $k_1 \in K$ such that

$$\|x_{k+1} - x_k\| < \epsilon/2 \text{ and } \|x_k - x_*\| < \epsilon/2$$

for all $k \in K, k \geq k_1$.

As in Theorem 3.3, define:

$$C = \{x \in I\!R^n \mid \epsilon/2 \leq \|x - x_*\| \leq \epsilon\}.$$

Clearly, C is compact and does not contain strongly critical points. Then, by Theorem 4.2, C cannot contain infinitely many iterates. Therefore, we have two possibilities:

1. There exists $k_2 \in I\!N$ such that $\|x_k - x_*\| \leq \epsilon/2$ for all $k \geq k_2$.
2. There exist infinitely many iterates $k \geq k_1$, such that $\|x_k - x_*\| \leq \epsilon/2$ and $\|x_{k+1} - x_k\| > \epsilon/2$.

In the first case, since x_* is the only possible limit point in the ball with radius $\epsilon/2$ we have that the sequence $\{x_k\}$ converges to x_*.

Let us analyze the second case. Let $K_1 \subset I\!N$ be such that $\|x_k - x_*\| \leq \epsilon/2$ and $\|x_{k+1} - x_k\| > \epsilon/2$ for all $k \in K_1$.

Since all the iterates belong to the ball with center $\epsilon/2$ and x_* is the only possible limit point in this ball, it turns out that

$$\lim_{k \in K_1} x_k = x_*.$$

By the hypothesis of vertical isolation, we have that $J_k \subset I_{min}(x_*)$ for $k \in K_1$ large enough. Moreover, $\nabla f_i(x_*) = 0$ for all $i \in I_{min}(x_*)$. Then, by (49),

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0,$$

contradicting the assumption $\|x_{k+1} - x_k\| \geq \epsilon/2 \ \forall \ k \in K_1$. This means that the second case mentioned above is impossible. So, the proof is complete. □

Remarks.

The result of the theorem does not hold if one uses $\varepsilon = 0$ in Algorithm **U2**. In fact, consider the problem (2) with $m = 2, n = 1$, $f_1(x) = x, f_2(x) = x^2$. For all $x \in (0, 1)$ we have that $f_{min}(x) = f_2(x) < f_1(x)$. Therefore, if $x_k \in (0, 1)$ and one uses Algorithm **U2** with $\varepsilon = 0$, the algorithm reduces to Algorithm **U1** and, with many admissible choices of the search directions, convergence to the weak (but not strong) critical point $x_* = 0$ occurs.

The assumption of vertical isolation cannot be eliminated. Consider the problem with $m = 2, n = 1$, $f_1(x) = (x + 1)^2, f_2(x) = (x - 1)^2$. The sequence produced by the algorithm may have two critical points $y_* = -1$ and $z_* = 1$. Take $\varepsilon > 4$. Assume that we start with x_0 close (but different) to y_*. The direction d_0^2 may be such that $x_0 + d_0^2$ is close (but different) to z_* and the direction d_0^1 may be that $x_0 + d_0^1$ is close (but different) to y_*. However, it may be possible that $f_2(x_0 + d_0^2) < f_1(x_0 + d_0^1)$ and that both directions satisfy the descent requirements of the algorithm. Therefore, x_1 will be close (but different) to z_*. This process may be repeated

indefinitely so that the sequence will have two different accumulation points.

Theorem 4.4. *Assume that x_* is a strongly isolated strict local minimizer which, in addition, is a-vertically isolated with $a > \varepsilon$. Let $\{x_k\}$ be a sequence generated by Algorithm* **U2** *with Assumption* **B3**. *Then, there exists $\epsilon > 0$ such that $\|x_0 - x_*\| \leq \epsilon$ implies that*

$$\lim_{k\to\infty} x_k = x_*.$$

Proof. Let $\epsilon > 0$ be such that x_* is a strict global minimizer of f_{min} in the ball $B(x_*, \epsilon)$ and that this ball does not contain strongly critical points other than x_*. Let us prove that there exists $\delta \in (0, \epsilon/2)$ such that

$$\|x_k - x_*\| \leq \delta \Rightarrow \|x_{k+1} - x_k\| \leq \epsilon/2. \tag{51}$$

In fact, since x_* is strongly critical, $\nabla f_i(x_*) = 0$ for all $i \in I_{min}(x_*)$. But the assumption of of vertical isolation with $a > \varepsilon$ implies that, in a neighborhood of x_*, $J_k \subset I_{min}(x_*)$. Then, by the continuity of the gradients and the assumption (49), we obtain (51).

The rest of the proof is as in Theorem 3.4. Let c be the minimum of $f_{min}(x)$ on the set defined by $\delta \leq \|x - x_*\| \leq \epsilon$. Let $\delta_1 \in (0, \delta)$ be such that

$$\|x - x_*\| \leq \delta_1 \Rightarrow f_{min}(x) < c.$$

Let us prove by induction that, taking $\|x_0 - x_*\| \leq \delta_1$, one has that $\|x_k - x_*\| \leq \epsilon/2$ and $f(x_k) < c$ for all k. By the definition of δ_1 this is true for $k = 0$. For the inductive step, observe that, by (51), we have that $\|x_{k+1} - x_*\| \leq \epsilon$. But, by the definition of c and the fact that $f(x_{k+1}) < f(x_k)$, we have that $\|x_{k+1} - x_*\| \leq \epsilon/2$.

Therefore, the whole sequence is contained in $B(x_*, \epsilon/2)$. Since the only strongly critical point in this ball is x_*, Theorem 4.2 implies that the whole sequence converges to x_* as we wanted to prove. □

The assumption of vertical isolation is essential for proving Theorems 4.3 and 4.4. In fact, consider the problem defined by $f_1(x) = x^2$, $f_2(x) = x + \varepsilon/2$, where vertical isolation does not hold. The point $x_* = 0$ is the unique strong local minimizer of this problem. However, for x_0 close to x_*, $J_0 = \{1, 2\}$. Taking $d_0^2 = -1$ we will have that $f_{min}(x^1) < 0$ so that convergence to 0 will be impossible. So, the thesis of Theorem 4.4 does not hold in this case.

Assumption **B4** establishes the specific implementation of Algorithm **U2** that produces local superlinear convergence. As in Algorithm **U1** we assume that the directions d_k^i are computed using the inexact solution of a linear Newton-like equation. To enhance the probability of taking pure Newton-like iterates we make the choice (55) below. This will be sufficient for proving, in Theorem 4.5 that superlinear convergence holds under similar conditions to those of Theorem 3.5.

Assumption B4. In the implementation of Algorithm **U2** we have:

- $\alpha \in \left(0, \frac{1}{2}\right)$.

- For all $i \in J_k$, the direction d_k^i is a solution of

$$B_k^i d = -\nabla f_i(x_k) + r_k, \tag{52}$$

where $B_k^i \in I\!R^{n\times n}$ is symmetric and positive definite and

$$\|r_k\| \leq \eta_k \|\nabla f_i(x_k)\|. \tag{53}$$

- If

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

then we choose $t_k^i = 1$.

- If there exists $j \in J_k$ such that $t_k^j = 1$, we choose

$$x_{k+1} = x_k + t_k^i d_k^i, \tag{54}$$

where

$$f_i(x_k + t_k^i d_k^i) = \min\{f_j(x_k + t_k^j d_k^j), j \in J_k\}. \tag{55}$$

- There exists $C > 0$ such that, for all $k \in I\!N$, $i \in J_k$, $\|(B_k^i)^{-1}\| \leq C$ and Assumption **B3** holds.

If η_k is small enough and $\|B_k^i\| < \frac{1}{\beta}$, the condition $d_k^i \geq \beta \|\nabla f_i(x_k)\|$ is satisfied. Moreover, if the condition number $\|B_k^i\|\|(B_k^i)^{-1}\|$ is less than or equal to $\frac{1}{\theta}$, the angle condition (33) is satisfied. Clearly, it is always possible to choose B_k^i satisfying both requirements.

Theorem 4.5. *Assume that:*

1. *The sequence $\{x_k\}$ is generated by Algorithm* **U2** *with Assumption* **B4**;
2. *x_* is a local minimizer;*
3. *For all $i \in I_{min}(x_*)$, the function f_i admits continuous second derivatives in a neighborhood of x_*;*
4. *$\nabla^2 f_i(x_*) > 0$ for all $i \in I_{min}(x_*)$;*
5. *$\lim_{k\to\infty} x_k = x_*$;*
6. *For all $i \in I_{min}(x_*)$, the Dennis-Moré condition*

$$\lim_{k\to\infty} \frac{\|[B_k^i - \nabla^2 f_i(x_k)]d_k^i\|}{\|d_k^i\|} = 0 \tag{56}$$

and the inexact-Newton condition

$$\lim_{k\to\infty} \eta_k = 0 \tag{57}$$

hold.

Then,

- *There exists $k_0 \in I\!N$ such that, for all $k \geq k_0$ and $i \in I_{min}(x_*)$, we have that $i \in J_k$ and $t_k^i = 1$.*
- *If $i \in J_k$ is such that*

$$f_i(x_k + t_k^i d_k^i) = \min\{f_j(x_k + t_k^j d_k^j), j \in J_k\} \tag{58}$$

for infinitely many indices k, then $i \in I_{min}(x_*)$.

- *There exists $k_1 \in I\!N$ such that for all $k \geq k_1$ there exists $\iota(k) \in J_k \cap I_{min}(x_*)$ such that $t_k^{\iota(k)} = 1$ and*

$$x_{k+1} = x_k + d_k^{\iota(k)}. \tag{59}$$

- *The sequence $\{x_k\}$ converges superlinearly to x_*.*

Proof. Let $i \in I_{min}(x_*)$. Since x_* must be strongly critical, we have that $\nabla f_i(x_*) = 0$. However, since $\nabla^2 f_i(x_*)$ is positive definite, $\nabla f_i(x_k) \neq 0$ for k large enough. Since $f_i(x_*) = f_{min}(x_*)$, by the continuity of f_i and f_{min} we have that $f_i(x_k) \leq f_{min}(x_k) + \varepsilon$ for k large enough. So, there exists $k_0' \in I\!N$ such that $i \in J_k$ for all $k \geq k_0'$.

The proof of (62) below mimics the proof of (32) in Theorem 3.5.

By Taylor's formula, for all $k \geq k_0'$, we have that

$$f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k) = (1-\alpha)(d_k^i)^T \nabla f_i(x_k) + \frac{1}{2}(d_k^i)^T \nabla^2 f_i(x_k)(d_k^i) + o(\|d_k^i\|^2)$$

$$= (1-\alpha)(d_k^i)^T[\nabla f_i(x_k) + \nabla^2 f_i(x_k) d_k^i] + (\alpha - \frac{1}{2})(d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2).$$

By (33), (53) and (57) we have that $\|r_k\| = o(\|d_k^i\|)$. Therefore,

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

$$= (1-\alpha)(d_k^i)^T[\nabla^2 f_i(x_k) - B_k^i] d_k^i + (\alpha - \frac{1}{2})(d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2).$$

But, by (56),

$$(1-\alpha)(d_k^i)^T[\nabla^2 f_i(x_k) - B_k^i] d_k^i = o(\|d_k^i\|^2),$$

therefore,

$$f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k) = \left(\alpha - \frac{1}{2}\right)(d_k^i)^T \nabla^2 f_i(x_k) d_k^i + o(\|d_k^i\|^2). \tag{60}$$

Let $\mu > 0$ a lower bound for the eigenvalues of $\nabla^2 f_i(x_*)$. Then, there exists $k_2 > k_0'$ such that $\mu/2$ is lower bound for the eigenvalues of $\nabla^2 f_i(x_k)$ for all $k \geq k_2$. So, for all $k \geq k_2$, we have:

$$\frac{(d_k^i)^T \nabla^2 f_i(x_k) d_k^i}{\|d_k^i\|^2} \geq \mu/2.$$

Since $\alpha < 1/2$, by (60), we have:

$$\frac{f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k)}{\|d_k^i\|^2} \leq \left(\alpha - \frac{1}{2}\right)\frac{\mu}{2} + \frac{o(\|d_k^i\|^2)}{\|d_k^i\|^2}. \tag{61}$$

for $k \geq k_2$. But, since $\|(B_k^i)^{-1}\| \leq C$ for all k and $\nabla f_i(x_*) = 0$, we have that $\|d_k^i\| \to 0$. So, taking limits in (61) for $k \to \infty$, we get:

$$f_i(x_k + d_k^i) - f_i(x_k) - \alpha(d_k^i)^T \nabla f_i(x_k) \leq 0 \tag{62}$$

for k large enough. So, by Assumption **B4**, there exists $k_0 \geq k_2$ such that $t_k^i = 1$ for all $k \geq k_0$. Therefore, the first part of the thesis is proved.

Let us now prove the second part of the thesis. By the first part of the thesis, for k large enough we choose x_{k+1} using (54) and (55). Assume that (58) holds for infinitely many indices $k \in K$. Then, for all $k \in K$,

$$f_i(x_{k+1}) = f_i(x_k + t_k^i d_k^i) \leq f_{min}(x_k).$$

Taking limits for $k \in K$, we obtain that $f_i(x_*) = f_{min}(x_*)$. So, $i \in I_{min}(x_*)$.

The third part of the thesis follows as a consequence of the first two. For k large enough $J_k \cap I_{min}(x_*) \neq \emptyset$. Therefore, by the first part of the thesis, for k large enough there exists i such that $t_k^i = 1$. Then, by (54) and the second part of the thesis, $x_{k+1} = x_k + t_k^{\iota(k)} d_k^{\iota(k)}$ and $\iota(k) \in I_{min}(x_*)$ for all k large enough. Then, by the first part of the thesis again, we obtain (59).

Now, we are able to prove the last part of the thesis.

As in the first part of the proof, by (33), (53) and (57) we have that $\|r_k\| = o(\|\nabla f_{\iota(k)}(x_k)\|) = o(\|d_k^{\iota(k)}\|)$. Then, by Taylor's formula:

$$\nabla f_{\iota(k)}(x_{k+1}) = \nabla f_{\iota(k)}(x_k) + \nabla^2 f_{\iota(k)}(x_k) d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|)$$

$$= B_k^{\iota(k)} d_k^{\iota(k)} + \nabla f_{\iota(k)}(x_k) + [\nabla^2 f_{\iota(k)}(x_k) - B_k^{\iota(k)}] d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|).$$

Then, by (52), (53) and (57),

$$\nabla f_{\iota(k)}(x_{k+1}) = [\nabla^2 f_{\iota(k)}(x_k) - B_k^{\iota(k)}] d_k^{\iota(k)} + o(\|d_k^{\iota(k)}\|).$$

So, by (56),

$$\lim_{k\to\infty} \frac{\|\nabla f_{\iota(k)}(x_{k+1})\|}{\|x_{k+1} - x_k\|} = 0.$$

But the continuity and nonsingularity of $\nabla^2 f_i(x)$ at x_*, this implies that

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_k\|} = 0.$$

It follows that

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_*\|}{\|x_{k+1} - x_*\| + \|x_k - x_*\|} = 0.$$

Therefore, superlinear convergence follows. □

5 Constrained LOVO problems

In this section we address the LOVO problem when the feasible set Ω is not whole space $\mathbb{R}^n$. We will assume that Ω is described by a set of equations and inequations and we will define a globally convergent Augmented Lagrangian algorithm for solving the constrained LOVO problem. For that purpose we need, first, to recall a suitable Augmented Lagrangian method for solving smooth constrained optimization problems.

5.1 Smooth Augmented Lagrangian method

We consider the problem

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, g(x) \leq 0, \tag{63}$$

where $f : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^{n_h}, g : \mathbb{R}^n \to \mathbb{R}^{n_g}$. We assume that f, h, g are continuously differentiable.

For all $x \in \mathbb{R}^n, \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}^{n_h}, \mu \in \mathbb{R}^{n_g}_+$ we define the Augmented Lagrangian [24, 34, 36, 37]:

$$L(x, \lambda, \mu, \rho) = f(x) + \frac{\rho}{2}\left[\left\|h(x) + \frac{\lambda}{\rho}\right\|^2 + \left\|\left(g(x) + \frac{\mu}{\rho}\right)_+\right\|^2\right]. \tag{64}$$

Algorithm **C** is an Augmented Lagrangian method for solving the smooth problem (63). Essentially, it is a particular case of the Augmented Lagrangian algorithm with arbitrary lower-level constraints described in [2] and implemented in the `Tango` web-page.

Algorithm C.

Let $x_0 \in \mathbb{R}^n$ be an arbitrary initial point.

The parameters for the execution of the algorithm are:

$$\tau \in [0, 1), \gamma > 1,$$

$$-\infty < \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty,$$

$$0 \leq \bar{\mu}_{\max} < \infty,$$

$$\rho_1 \in \mathbb{R}_{++},$$

$$[\bar{\lambda}_1]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}] \text{ for all } j = 1, \ldots, n_h,$$

$$[\bar{\mu}_1]_j \in [0, \bar{\mu}_{\max}] \text{ for all } j = 1, \ldots, n_g.$$

$$\varepsilon_1 > 0.$$

Step 1. *Initialization*

Set $k \leftarrow 1$. For $j = 1, \ldots, n_g$, compute

$$[\sigma_0]_j = \max\{g_j(x_0), 0\}.$$

Step 2. *Solving the subproblem*

Compute $x_k \in \Omega$ such that

$$\|\nabla L(x_k, \lambda_k, \mu_k, \rho_k)\|_\infty \leq \varepsilon_k.$$

Step 3. *Estimate multipliers*

For all $j = 1, \ldots, n_h$, 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 $j = 1, \ldots, n_g$, compute

$$[\mu_{k+1}]_j = \max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\},$$

$$[\sigma_k]_j = \max\left\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\right\},$$

and

$$[\bar{\mu}_{k+1}]_j \in [0, \bar{\mu}_{max}].$$

Step 4. *Update the penalty parameters*

If

$$\max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \leq \tau \max\{\|h(x_{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\}, \tag{65}$$

define

$$\rho_{k+1} \geq \rho_k. \tag{66}$$

Else, define

$$\rho_{k+1} \geq \gamma \rho_k. \tag{67}$$

Step 5. *Begin a new outer iteration*

Compute $\varepsilon_{k+1} > 0$. Set $k \leftarrow k + 1$. Go to Step 2.

The only differences between Algorithm **C** and the algorithm introduced in [2] (in the case that no lower-level constraints are present) is in the updating rules (66) and (67). In [2] the authors set $\rho_{k+1} = \rho_k$ when (65) holds and $\rho_{k+1} = \gamma \rho_k$ otherwise. This difference does not affect at all the proofs of the following convergence theorems.

Theorem 5.1. *Assume that $\{x_k\}$ is an infinite sequence generated by Algorithm C 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. See Theorem 4.1 of [2]. □

Theorem 5.2. *Assume that $\{x_k\}$ is an infinite sequence generated by Algorithm C with $\varepsilon_k \to 0$, x_* is a limit point and the constant positive linear dependence (CPLD) constraint qualification [9, 35] is fulfilled at x_*. Then, x_* is a KKT point of (63).*

Proof. See Theorem 4.2 of [2]. □

The final boundedness result for the penalty parameters associated to Algorithm C is given in Theorem 5.3. A crucial assumption will be that the precision used to solve subproblems must tend to zero faster than the feasibility measure. This type of requirement is usual in many Augmented Lagrangian and Multiplier methods [12, 13, 17, 18, 19, 20, 21, 23].

Assumption C1. *We assume that*

1. *The sequence $\{x_k\}$ is generated by the application of Algorithm C to the problem (63) and*

$$\lim_{k\to\infty} x_k = x_*.$$

2. *In (66) the rule $\rho_{k+1} = \rho_k$ is employed.*

3. *The point x_* is feasible ($h(x_*) = 0, g(x_*) \leq 0$).*

4. *The gradients*

$$\{\nabla h_j(x_*)\}_{j=1}^{n_h}, \{\nabla g_j(x_*)\}_{\{j \mid g_j(x_*)=0\}},$$

   are linearly independent.

5. *Strict complementarity takes place at x_*. This means that, if $\mu_* \in I\!R_+^{n_g}$ is the vector of Lagrange multipliers corresponding to the constraints $g(x) \leq 0$, then:*

$$g_j(x_*) = 0 \Rightarrow [\mu_*]_j > 0.$$

6. *The functions f, h, g admit continuous second derivatives in a neighborhood of x_*.*

7. *Define the tangent subspace T as the set of all $z \in I\!R^n$ such that*

$$\nabla h(x_*)^T z = 0,$$

$$\nabla g_j(x_*)^T z = 0$$

   for all j such that $g_j(x_*) = 0$.

   Then, for all $z \in T, z \neq 0$,

$$z^T \left(\nabla^2 f(x_*) + \sum_{i=1}^{n_h} [\lambda_*]_j \nabla^2 h_j(x_*) + \sum_{j=1}^{n_g} [\mu_*]_j \nabla^2 g_j(x_*) \right) z > 0.$$

Theorem 5.3. *Suppose that Assumption* **C1** *holds. In addition, assume that:*

1. *There exists a sequence* $\eta_k \to 0$ *such that*

$$\varepsilon_k \leq \eta_k \max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \; \forall \; k \in I\!N.$$

2. $[\lambda_*]_j \in (\bar{\lambda}_{\min}, \bar{\lambda}_{\max}) \; \forall \; j = 1, \ldots, n_h$ *and* $[\mu_*]_j \in (\bar{\mu}_{\min}, \bar{\mu}_{\max}) \; \forall \; j = 1, \ldots, n_g$.

3. $[\bar{\lambda}_{k+1}]_j$ *is the projection of* $[\lambda_{k+1}]_j$ *on* $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ *and* $[\bar{\mu}_{k+1}]_j$ *is the projection of* $[\mu_{k+1}]_j$ *on* $[0, [\bar{\mu}_{\max}]$ *for all* $j = 1, \ldots, n_h, j = 1, \ldots, n_g, k \in I\!N$.

Then, the sequence of penalty parameters $\{\rho_k\}$ *is bounded.*

Proof. See Theorem 5.5 of [2]. □

5.2 Augmented Lagrangian method for LOVO

Now we are in conditions to define natural extensions of Algorithm C to the LOVO problem. When the solution of unconstrained minimization subproblems is needed, one may use Algorithms **U1** or **U2**.

We consider the problem

$$\text{Minimize } f_{min}(x) \text{ subject to } h(x) = 0, g(x) \leq 0, \tag{68}$$

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

As in (64), 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 $x \in I\!R^n$,

$$I_{min}(x) = \{i \in \{1, \ldots, m\} \mid f_i(x, \lambda, \mu, \rho) = f_{min}(x, \lambda, \mu, \rho)\}.$$

Observe that

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

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

Algorithm C-LOVO.

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

The parameters for the execution of the algorithm are:

$$\tau \in [0, 1), \gamma > 1,$$

$$-\infty < \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty,$$
$$0 \leq \bar{\mu}_{\max} < \infty,$$
$$\rho_1 \in I\!R_{++},$$
$$[\bar{\lambda}_1]_j \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}] \; \forall j = 1, \ldots, n_h,$$
$$[\bar{\mu}_1]_j \in [0, \bar{\mu}_{\max}] \; \forall j = 1, \ldots, n_g.$$
$$\varepsilon_1 > 0.$$

Step 1. *Initialization*

Set $k \leftarrow 1$. For $j = 1, \ldots, n_g$, compute

$$[\sigma_0]_j = \max\{g_j(x_0), 0\}.$$

Step 2. *Solving the subproblem*

Compute $x_k \in I\!R^n$ such that

$$\|\nabla L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)\|_\infty \leq \varepsilon_k \tag{69}$$

for some $i \in I_{min}(x_k)$.

Step 3. *Estimate multipliers*

For all $j = 1, \ldots, n_h$, 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 $j = 1, \ldots, n_g$, compute

$$[\mu_{k+1}]_j = \max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\},$$

$$[\sigma_k]_j = \max\left\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\right\},$$

and

$$[\bar{\mu}_{k+1}]_j \in [0, \bar{\mu}_{max}].$$

Step 4. *Update the penalty parameters*

If

$$\max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \leq \tau \max\{\|h(x_{k-1})\|_\infty, \|\sigma_{k-1}\|_\infty\},$$

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 (69) 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 (69) can be found, provided that the generated sequence is bounded. On the other hand, boundedness of the sequences generated by Algorithms **U1** or **U2** may be guaranteed under suitable relations between objective function and constraints.

In Theorem 5.4 we prove that Algorithm **C-LOVO** finds stationary points of the constraint infeasibility.

Theorem 5.4. *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 $i \in \{1, \ldots, m\}$ such that (69) holds for f_i infinitely many times. Taking the corresponding subsequence of $\{x_k\}$, it turns out that this subsequence may be thought as generated by Algorithm **C**. Therefore, the thesis follows by Theorem 5.1. □

Theorem 5.5. *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*

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

Proof. As in Theorem 5.4, consider an infinite subsequence of $\{x_k\}$ such that (69) holds with the same index i for all the terms of this subsequence. Again, this subsequence may be thought as having been generated by Algorithm C. By Theorem 5.2 there exists x_* satisfying the thesis of this theorem. The fact that $i \in I_{min}(x_*)$ follows trivially from $L_i(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k) \leq L_j(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)$ for all j. □

Remark.

In (69) 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 5.5, 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 unuseful in terms of the solutions that can be obtained by **C-LOVO**.

The final boundedness result for the penalty parameters associated to Algorithm **C-LOVO** is given in Theorem 5.6. As in the previous theorems, the technique consists of reducing the LOVO problem to a smooth nonlinear programming problem. However, in this case, we will need an additional assumption: given a convergent sequence generated by Algorithm **C-LOVO**, we will assume that there exists a unique index i_{min} such that $f_{min}(x_k) = f_{i_{min}}(x_k)$ if k is large enough. In this way, we are able to ensure that the algorithm, ultimately, behaves as Algorithm **C** for the minimization of $f_{i_{min}}$.

Assumption C2. *We assume that*

1. *The sequence $\{x_k\}$ is generated by the application of Algorithm* **C-LOVO** *to the problem (63) and*
$$\lim_{k\to\infty} x_k = x_*.$$
2. *The point x_* is feasible ($h(x_*) = 0, g(x_*) \leq 0$).*
3. *There exists $i_{min} \in \{1, \ldots, m\}$ such that*
$$f_{i_{min}}(x_k) = f_{min}(x_k) < f_i(x^k)$$
for all k large enough and $i \neq i_{min}$.
4. *The gradients*
$$\{\nabla h_j(x_*)\}_{j=1}^{n_h}, \{\nabla g_j(x_*)\}_{\{j \mid g_j(x_*)=0\}}$$
are linearly independent.
5. *Strict complementarity takes place at x_*. This means that, if $\mu_* \in \mathbb{R}_+^{n_g}$ is the vector of Lagrange multipliers corresponding to the constraints $g(x) \leq 0$, then:*
$$g_j(x_*) = 0 \Rightarrow [\mu_*]_j > 0.$$
6. *The functions $f_{i_{min}}, h, g$ admit continuous second derivatives in a neighborhood of x_*.*
7. *Define the tangent subspace T as the set of all $z \in \mathbb{R}^n$ such that*
$$\nabla h(x_*)^T z = 0,$$
$$\nabla [g(x_*)]_j^T z = 0$$
for all j such that $g_j(x_*) = 0$.

Then, for all $z \in T, z \neq 0$,

$$z^T \left(\nabla^2 f_{i_{min}}(x_*) + \sum_{j=1}^{n_h} [\lambda_*]_j \nabla^2 h_j(x_*) + \sum_{j=1}^{n_g} [\mu_*]_j \nabla^2 g_j(x_*) \right) z > 0.$$

Theorem 5.6. *Suppose that Assumption* **C2** *holds. In addition, assume that:*

1. *There exists a sequence* $\eta_k \to 0$ *such that*

$$\varepsilon_k \leq \eta_k \max\{\|h(x_k)\|_\infty, \|\sigma_k\|_\infty\} \; \forall \; k \in I\!N.$$

2. $[\lambda_*]_j \in (\bar{\lambda}_{\min}, \bar{\lambda}_{\max}) \; \forall \; j = 1, \ldots, n_h$ *and* $[\mu_*]_j \in (\bar{\mu}_{\min}, \bar{\mu}_{\max}) \; \forall \; j = 1, \ldots, n_g$.

3. $[\bar{\lambda}_{k+1}]_j$ *is the projection of* $[\lambda_{k+1}]_j$ *on* $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ *for all* $j = 1, \ldots, n_h$, *and* $[\bar{\mu}_{k+1}]_j$ *is the projection of* $[\mu_{k+1}]_j$ *on* $[0, \bar{\mu}_{\max}]$ *for all* $j = 1, \ldots, n_g, k \in I\!N$.

Then, the sequence of penalty parameters $\{\rho_k\}$ *is bounded.*

Proof. For k large enough the sequence may be thought as being generated by Algorithm **C** with Assumption **C1**. So, the thesis follows from Theorem 5.3. □

6 Hidden patterns

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 $D : I\!R^{n_q} \to I\!R^{n_p}$ such that some subset of $\{D(Q_1), \ldots, D(Q_N)\}$ fits some subset of $\mathcal{P}$. In Section 7, D could represent only rigid-body displacements but here we allow more general transformations. 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 de 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} follows 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

$$\|D(Q_i) - P_{c(i)}(D)\| \leq \|D(Q_i) - P\| \ \ \forall\, P \in \mathcal{P}. \tag{70}$$

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 (70). 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 independently 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.

7 Protein alignment

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 to the displacement D and the bijection Φ is:

$$StS(D, \Phi) = \sum \frac{20}{1 + \|P_k - D(Q_{\Phi(k)})\|^2/5}, -10 \times gaps, \tag{71}$$

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 [30]. However, this method is not computationally affordable (see [30]) and, in practice, an heuristic procedure called *Structal Method* [40] is generally used. In [31], 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 [28, 29]. The main drawback of the Structal Method is that the Update-Φ step aims 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 [5]. With the aim of overcoming this problem we suggest a different algorithm (DP-Newton), 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-Newton 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-Newton is Algorithm **U1** applied to the maximization of the Structal Score, both with respect to Φ and D.

The application of DP-Newton to the alignment of proteins of the Protein Data Bank (PDB) [11] is fully described in [5]. Using 79800 individual protein comparisons it may be concluded that:

- DP-Newton is systematically able to obtain the best scores in the highest percentage of cases for all alignment qualities. For alignments with (scaled) best-scores greater than 6, for example, DP-Newton obtains the best scores in at least 90% of the cases. For alignments with best scores greater than 12, DP-Newton obtains the best scores in 98% of the problems. The Structal Method is competitive with DP-Newton for bad alignments (scores lower than 3) and for very good alignments (scores grater than 18), but for most cases the best scores are obtained in only 10 to 40% of the problems.

- The computer time used by DP-Newton is, on average, 2/3 the computer time employed by the Structal Method on the tests reported in [5].

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 [5]. 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-Newton 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 [6]. 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.

8 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 one ignores the multiplicity of gradients at a given point x^k and we use straightforward smooth minimization solvers, the bad consequences are rather mild. In fact, a far more serious inconvenient is the fact that convergence to global minimizers is not guaranteed, but this inconvenient is shared by most practical smooth nonlinear-programming methods.

Many smooth optimization algorithms, when applied to LOVO, may be considered particular cases of Algorithms **U1** and **C-LOVO**. With this 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 [1, 2, 14]. Considering a rather large number of unconstrained and constrained tests, we did not detect practical differences between the performance of algorithms **U1** and **U2**. In constrained problems this is as predicted by theory, because **C-LOVO** cannot guarantee convergence to strongly critical points.

All the experiments were run on an 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.

8.1 A Hidden-Pattern Example

We consider the application LOVO described in Section 6. The points of $\mathcal{P}$, represented in Figure 1.(a) 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 Figure 8.1(a), are 78 Cα atoms of the C-terminal region of a similar protein, however 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 [25].

We used a multistart approach, since this type of problems has 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.

The best solution was obtained in the third trial. The last execution of the unconstrained algorithm used 21 iterations. So, Algorithm **U1** ran three times, finding critical points in the first two cases. On average, the distance between the displaced points of $\mathcal{Q}$ and the points of $\mathcal{P}$ was 1.07 angstrons (the best solution found is correct from the point of view of protein function and is, very likely, the global solution). In Figure 1.(b) 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 1.(c) 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).

8.2 Fitting Models with Outliers

8.2.1 Unconstrained fitting

Assume that $\{(t_1, y_1), \ldots, (t_m, y_m)\} \subset I\!R^2$ 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 $y_i \approx T(x, t_i)$ leads to unsatisfactory results due to the overwhelming influence of outliers.

The LOVO approach for robust estimation of parameters consists in defining, for each $i = 1, \ldots, r$, the error function

$$F_i(x) = (T(x, t_i) - y_i)^2.$$

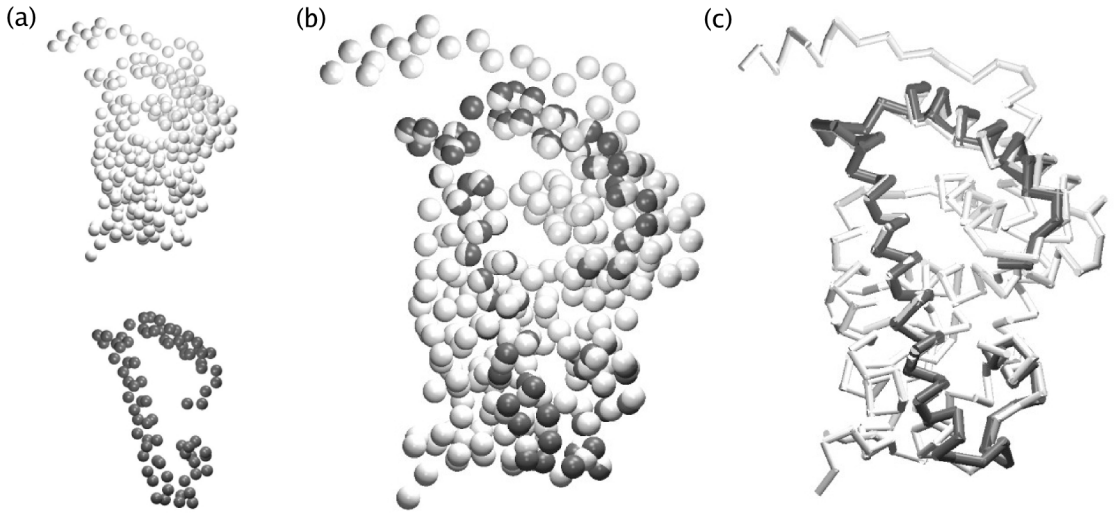

Figure 1: Finding patterns of protein folding with LOVO.

Given $p \in \{1, \ldots, r\}$, this set of functions defines a LOVO problem (1) for which algorithms **U1**, **U2** (unconstrained cases) and **C-LOVO** (constrained cases) may be employed. 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. The idea is to solve this problem for different values of p. If $p = r$ we expect a large value of the LOVO function at the solution, showing that there are wrong data among the points that correspond to $F_{i_1}, \ldots, F_{i_r}$. When p is decreased, the LOVO function at the solution tends to decrease as well. Obviously, this decrease is due to the fact that the quantity of terms in the sum is smaller but, we expect that, when we take "the correct p", the magnitude of this decrease would be greater.

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 [33], where $r = 65$). Here we introduced 13 additional data representing systematic errors. The results are shown in Figure 2. The points in the graphics represent the given data (t_i, y_i). The rounded points are the detected outliers. The full line is the fitted curve. For $p = 78$ the full line gives the ordinary least-squares fitting. For $p = 65$ all the outliers are detected and the fitted curve is the "correct" one. In both cases we used the initial point given in [33]. The sum of squares was observed to decrease abruptly from $p = 66$ to $p = 65$, as expected.

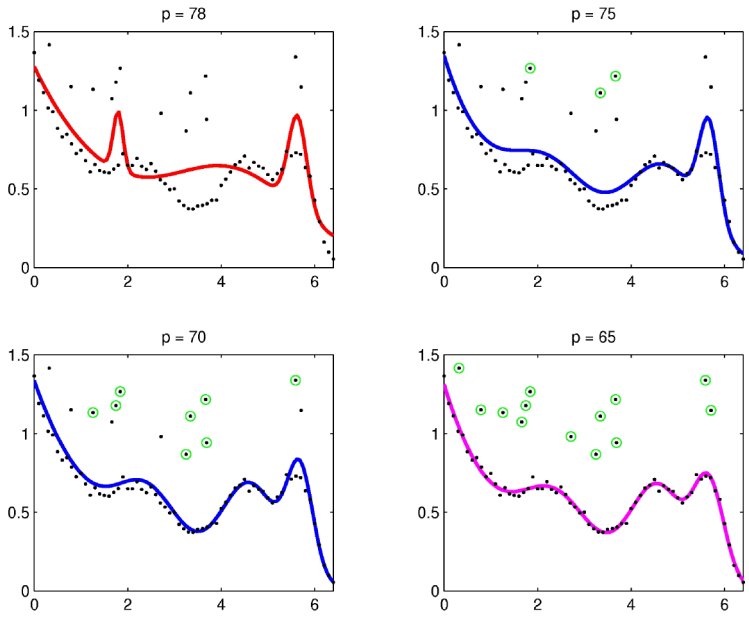

Figure 2: Unconstrained model fitting

8.2.2 Constrained fitting

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

$$\frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} = \Phi(t_i, x_i, z) \tag{72}$$

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 $y_1, \ldots, y_r$. We know 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 (72).

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 (72) 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 (Figure 3). The results for $p = 21$ and $p = 16$ are shown in Figure 3. As initial approximation we used x_i random between 0 and $2|y_i|$ and z_i random between -10 and 10. For $p = 21$ the solution is distorted by the necessity of fitting the outliers and the value of the LOVO function at the solution was 5.27. For $p = 16$ the fitted solution coincided with the correct data and the LOVO function value was less than 0.001.

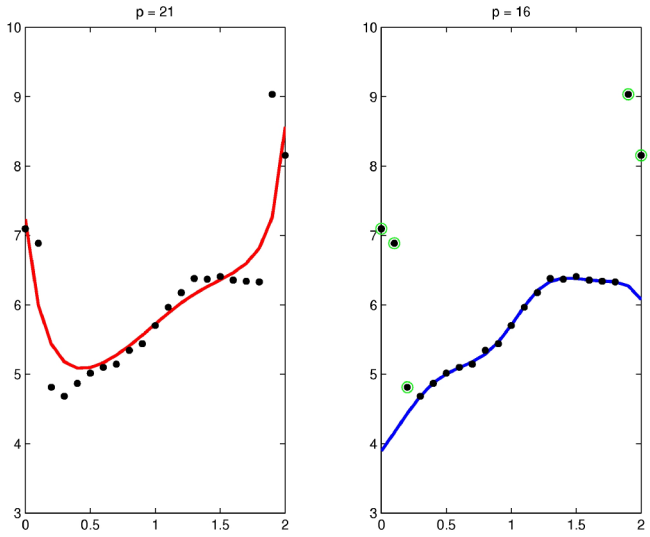

Figure 3: Model fitting with constraints

9 Final remarks

The LOVO problem defined in this paper is, in general, nonsmooth and nonconvex. Here we gave (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 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.

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:

- Exploiting smooth reformulations like the one proposed in [4] 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.

Acknowledgement. We are indebted to Mario Gneri, for insightful conversations on robust estimation problems.

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt. Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification. To appear in *Mathematical Programming* (DOI: 10.1007/s10107-006-0077-1).

[2] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. Technical Report MCDO-050303, Department of Applied Mathematics, UNICAMP, Brazil, 2005 (Available in *Optimization On Line*).

[3] 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) .

[4] 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).

[5] R. Andreani, J. M. Martínez and L. Martínez. Protein structural alignment: Development of convergent, continuous score optimization methods in theory and in practice, Technical Report, Department of Applied Mathematics, State University of Campinas. Available in `www.ime.unicamp.br/∼martinez/lovoalign`.

[6] R. Andreani, J. M. Martínez, L. Martínez and F. Yano. Continuous Optimization Methods for Structural Alignment. To appear in *Mathematical Programming*, 2007.

[7] 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).

[8] R. Andreani, J. M. Martínez, M. Salvatierra and F. Yano. Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy. In *Global Optimization: Theory and Practice*, pp. 379–404. Edited by L. Liberti and N. Maculan, Kluwer (2006).

[9] 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).

[10] Ch. Audet and J. E. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization* 17, pp. 188-217 (2006).

[11] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov and P.E. Bourne, The Protein Data Bank, *Nucleic Acids Res.* 28, pp. 235-242 (2000).

[12] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982; republished by Athena Scientific, 1996.

[13] D. P. Bertsekas. *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.

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

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

[16] F.H. Clarke. Optimization and nonsmooth Analysis Classic in Applied mathematics, SIAM, Philadelphia, 1990.

[17] A. R. Conn, N. I. M. Gould, A. Sartenaer and Ph. L. Toint. Convergence properties of an Augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints, *SIAM Journal on Optimization* 6, pp. 674–703 (1996).

[18] A. R. Conn, N. I. M. Gould and Ph. L. Toint. A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Numerical Analysis* 28, pp. 545–572 (1991).

[19] Z. Dostál. Inexact semi-monotonic Augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming. To appear in *SIAM Journal on Numerical Analysis*.

[20] Z. Dostál, A. Friedlander and S. A. Santos. Augmented Lagrangian with adaptive precision control for quadratic programming with simple bounds and equality constraints, *SIAM Journal on Optimization* 13, pp. 1120–1140 (2003).

[21] Z. Dostál, F. A. M. Gomes and S. A. Santos. Duality based domain decomposition with natural coarse space for variational inequalities, *Journal of Computational and Applied Mathematics* 126, pp. 397–415 (2000).

[22] R. Fletcher. *Practical Methods of Optimization*, Academic Press, London, 1987.

[23] W. W. Hager. Analysis and implementation of a dual algorithm for constrained optimization, *Journal of Optimization Theory and Applications* 79, pp. 37–71 (1993).

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

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

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

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

[28] W. Kabsch, W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallog. A* 34, pp. 827-828 (1978).

[29] S. K. Kearsley. On the orthogonal transformation used for structural comparisons. *Acta Crystallog. A* 45, pp. 208-210 (1996).

[30] R. Kolodny and N. Linial. . Approximate protein structural alignment in polynomial time, *P. Natl. Acad. Sci. USA* 101, pp. 12201–12206 (2004).

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

[32] O. L. Mangasarian and S. Fromovitz. The Fritz-John necessary optimality conditions in presence of equality and inequality constraints, *Journal of Mathematical Analysis and Applications* 17, pp. 37–47 (1967).

[33] J. J. Moré, B. S. Garbow and K. E. Hillstrom. Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7, pp. 17-41 (1981).

[34] 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).

[35] 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).

[36] R. T. Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming, *Journal of Optimization Theory and Applications* 12, pp. 555–562 (1973).

[37] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control* 12, pp. 268-285 (1974).

[38] R. T. Rockafellar. Lagrange multipliers and optimality, *SIAM Review* 35, pp. 183–238 (1993).

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

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