Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy

R. Andreani * J. M. Martínez † M. Salvatierra ‡ F. Yano §

November 24, 2004 14:40

Abstract

The OVO (Order-Value Optimization) problem consists in the minimization of the Order-Value function $f(x)$, defined by

$$f(x) = f_{i_p(x)}(x),$$

where

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

The functions $f_1, \ldots, f_m$ are defined on $\Omega \subset I\!R^n$ and p is an integer between 1 and m.

When x is a vector of portfolio positions and $f_i(x)$ is the predicted loss under the scenario i, the Order-Value function is the discrete Value-at-Risk (VaR) function, which is largely used in risk evaluations.

The OVO problem is continuous but nonsmooth and, usually, has many local minimizers. A local method with guaranteed convergence

*Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 01-04597-4) and CNPq (Grant 301115/96-6). e-mail: andreani@ime.unicamp.br

†Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, This author was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 01-04597-4) and CNPq. Brazil. e-mail: martinez@ime.unicamp.br

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by PRONEX-Optimization 76.79.1008-00 and FAPESP (Grant 02-00171-5). e-mail: otita@ime.unicamp.br

§Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil.This author was supported by PRONEX-Optimization 76.79.1008-00 and FAPESP (Grant 02-14203-6). e-mail: yano@ime.unicamp.br

to points that satisfy an optimality condition was recently introduced by Andreani, Dunder and Martínez. The local method must be complemented with a global minimization strategy in order to be effective when m is large. A global optimization method is defined where local minimizations are improved by a tunneling strategy based on the harmonic oscillator initial value problem. It will be proved that the solution of this initial value problem is a smooth and dense trajectory if Ω is a box.

An application of OVO to the problem of finding hidden patterns in data sets that contain many errors is described. Challenging numerical experiments are presented.

Key words: Order-Value optimization, local methods, harmonic oscillator, tunneling, hidden patterns.

1 Introduction

Given m continuous functions $f_1, \ldots, f_m$, defined in a domain $\Omega \subset I\!R^n$ and an integer $p \in \{1, \ldots, m\}$, the $p-$Order-Value (OVO) function f is given by

$$f(x) = f_{i_p(x)}(x)$$

for all $x \in \Omega$, where $i_p(x)$ is an index function such that

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \ldots \leq f_{i_p(x)}(x) \leq \ldots \leq f_{i_m(x)}(x).$$

The OVO function is continuous [2]. However, even if the functions f_i are differentiable, the function f may not be smooth. The OVO problem consists in the minimization of the Order-Value function:

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

The definition of the OVO problem was motivated by two main applications.

1. Assume that Ω is a space of decisions and, for each $x \in \Omega$, $f_i(x)$ represents the cost of decision x under the scenario i. The Minimax decision corresponds to choose x in such a way that the maximum possible cost is minimized. This is a very pessimistic alternative and decision-makers may prefer to discard the worst possibilities in order to proceed in a more realistic way. For example, the decision maker may want to discard the 10 % more pessimistic scenarios. This corresponds

to minimize the $p-$Order-Value function with $p \approx 0.9 \times m$. When $f_i(x)$ represents the predicted loss for the set x of portfolio positions under the scenario i, the function $f(x)$ is the Value-at-Risk function, which is largely used in the finance industry. See [15].

2. Assume that we have a parameter-estimation problem where the space of parameters is Ω and $f_i(x)$ is the error corresponding to the observation i when the parameter x is adopted. The Minimax estimation problem corresponds to minimize the maximum error. As it is well-known this estimate is very sensitive to the presence of outliers [14]. Sometimes, we want to eliminate (say) the 15% larger errors because they can represent wrong observations. This leads to minimize the $p-$Order-Value function with $p \approx 0.85 \times m$. The OVO strategy is adequate to avoid the influence of systematic errors.

In [2] the continuity and differentiability properties of the Order-Value function was proved, nonsmoothness was discussed, local optimality conditions were introduced and, according to them, a local algorithm was defined. This algorithm is guaranteed to converge only to points that satisfy the optimality conditions, which are not necessarily global minimizers. A different approach was used in [1]. In this paper a nonlinear-programming reformulation of the OVO problem was defined. So, a general nonlinear programming solver can be used for solving it but, again, global solutions may be very difficult to find. Nonlinear programming methods that take advantage of the structure of the OVO reformulation were presented in [22, 23] but it is too soon for an evaluation of the effectiveness of these reformulations for solving practical problems.

The objective of the present work is to insert the local algorithm in a global heuristic and to apply the resulting method to the problem of finding hidden patterns. It is interesting to observe that, both in the application to decision problems and in the application to robust estimation of parameters, the OVO problem used corresponds to large values of p (generally close to m) whereas in the hidden-pattern problem the small values of p are the interesting ones. This is because we assume, in the latter problems, that all except a small number of data are corrupted.

This paper is organized as follows. The local algorithm is presented in Section 2. In Section 3 we introduce Lissajous motions, which are the basis of the global heuristic. In Section 4 we describe the global optimization algorithm. The hidden-pattern problem is discussed in Section 5. Numerical experiments are shown in Section 6 and in Section 7 we give some conclusions and discuss the lines for future research.

2 Local Algorithm

In this section we present the local algorithm which will be used in the calculations. Before, let us define for all $\varepsilon > 0$ and $x \in \Omega$:

$$I_\varepsilon(x) = \{j \in \{1, \ldots, m\} \mid f(x) - \varepsilon \leq f_j(x) \leq f(x) + \varepsilon\}.$$

Algorithm 2.1

Let $x_0 \in \Omega$ be an arbitrary initial point. Let $\theta \in (0,1)$, $\Delta > 0$, $\varepsilon > 0$, $0 < \sigma_{min} < \sigma_{max} < 1$, $\eta \in (0,1]$.

Given $x_k \in \Omega$ the steps of the k-th iteration are:

Step 1. (Solving the subproblem)

Define

$$M_k(d) = \max_{j \in I_\varepsilon(x_k)} \nabla f_j(x_k)^T d.$$

Consider the subproblem

$$\text{Minimize } M_k(d) \text{ subject to } x_k + d \in \Omega,\ \|d\|_\infty \leq \Delta. \tag{2}$$

Let $\bar{d}_k$ be a solution of (2). Let d_k be such that $x_k + d_k \in \Omega$, $\|d_k\| \leq \Delta$ and

$$M_k(d_k) \leq \eta M_k(\bar{d}_k). \tag{3}$$

If $M_k(d_k) = 0$ stop.

Step 2. (Steplength calculation)

Set $\alpha \leftarrow 1$.

If

$$f(x_k + \alpha d_k) \leq f(x_k) + \theta \alpha M_k(d_k) \tag{4}$$

set $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$ and finish the iteration. Otherwise, choose $\alpha_{new} \in [\sigma_{min}\alpha, \sigma_{max}\alpha]$, set $\alpha \leftarrow \alpha_{new}$ and repeat the test (4).

Observe that, when Ω is convex, (2) is equivalent to the convex optimization problem

$$\text{Minimize } w$$

$$\nabla f_j(x_k)^T d \leq w,\ \forall\, j \in I_\varepsilon(x_k),$$

$$x_k + d \in \Omega,\ \ \|d\|_\infty \leq \Delta.$$

Assume that $\Omega \subset I\!R^n$ is closed and convex, and $f_1, \ldots, f_m$ have continuous partial derivatives in an open set that contains Ω.

For all $x, y \in \Omega$, $j = 1, \dots, m$, we assume that

$$\|\nabla f_j(x)\|_\infty \leq c,$$

and

$$\|\nabla f_j(y) - \nabla f_j(x)\|_\infty \leq L\|y - x\|_\infty.$$

Definition. We say that x is ε-optimal (or critical) if

$$\mathcal{D} \equiv \{d \in I\!R^n \mid x + d \in \Omega \text{ and } \nabla f_j(x)^T d < 0, \ \forall\, j \in I_\varepsilon(x)\} = \emptyset.$$

The following theorem was proved in [2].

Theorem 2.1. *Assume that $x_k \in \Omega$ is the $k-th$ iterate of Algorithm 2.1. Then:*

1. *The algorithm stops at x_k if, and only if, x_k is a critical point. If the algorithm does not stop at x_k, then the $k-th$ iteration is well-defined and finishes at Step 2 with the computation of x_{k+1}.*
2. *Suppose that $x_* \in \Omega$ is a limit point of a sequence generated by Algorithm 2.1. Then x_* is critical.*

3 Lissajous motions

Assume that $\Omega \subset I\!R^n$ is a bounded box with nonempty interior. That is:

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

The Harmonic Oscillator Initial-Value problem is the system of n independent harmonic oscillators given by:

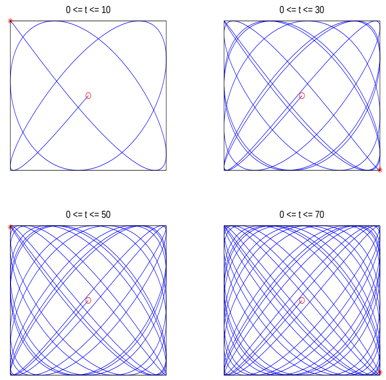
$$\frac{d^2}{dt^2} x_i(t) + \theta_i^2 x_i(t) = 0, \ \ i = 1, \dots, n \tag{5}$$

where $\theta_i^2 = \frac{[k_e]_i}{g}$, g is the mass of the body and the $[k_e]_i$'s are the elasticity constants.

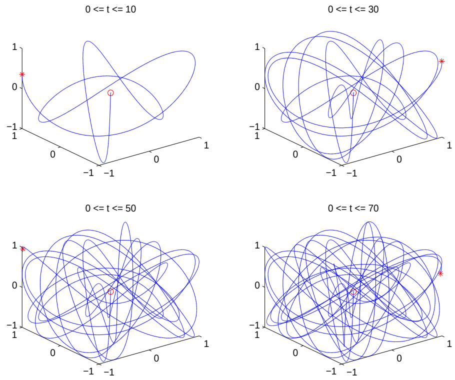
The solutions of (5) are:

$$\alpha(t) = (\cos(\theta_1 t + \varphi_1), \dots, \cos(\theta_n t + \varphi_n)), \tag{6}$$

where $\varphi_1, \dots, \varphi_n$ are constants. The trajectory defined by each solution (6) is called a *Lissajous curve*. See [12] p. 36. In Pictures 1 and 2 we show examples of these curves for $n = 2$ and $n = 3$, respectively.



Picture 1



Picture 2

Given $x_0 \in \Omega$ and choosing appropriately $\varphi_1, \dots, \varphi_n$ we can find a Lissajous curve such that $x(0) = x_0$. Clearly, Lissajous curves are smooth. In this section we give a simple proof that the image of a Lissajous curve is dense in $[-1, 1]^n$.

Definition. We say that $\theta_1, \dots, \theta_n \in \mathbb{R}$ are linearly independent over $\mathbb{Q}$ if

$$\sum_{i=1}^{n} r_i \theta_i = 0 \quad \text{and} \quad r_1, \dots, r_n \in \mathbb{Q}$$

only if $r_1 = \dots = r_n = 0$.

Theorem 3.1. (**Kronecker's Approximation Theorem**) *Let $h_1, h_2, \dots, h_n \in \mathbb{R}$ be linearly independent over $\mathbb{Q}$, $\xi_1, \dots, \xi_n \in \mathbb{R}$ and $\varepsilon > 0$. Then, there exists $t \in \mathbb{R}$ and $k_i \in \mathbb{Z}$ such that*

$$|h_i t - \xi_i - k_i| < \varepsilon, \ \forall \ \ i = 1, \dots, n.$$

Proof. See [13] pp. 431-437. ∎

Theorem 3.2. *Let $\alpha : \mathbb{R} \longrightarrow [-1, 1]^n$ be the Lissajous curve given by (6), where $\theta_1, \dots, \theta_n$ are linearly independent over $\mathbb{Q}$. Then, the image of $\alpha(t)$ is dense in $[-1, 1]^n$.*

Proof. Let $\varepsilon > 0$ be arbitrarily small and $x \in [-1, 1]^n$.

Let $\lambda \in I\!R^n$ be such that

$$\cos(\lambda_i) = x_i, \ \forall \ \ i = 1, \ldots, n. \tag{7}$$

By the uniform continuity of the cosine function, there exists $\delta > 0$ such that

$$|t_1 - t_2| \leq \delta \ \Rightarrow \ |\cos(t_1) - \cos(t_2)| \leq \varepsilon, \ \forall \ \ t_1, t_2 \in I\!R. \tag{8}$$

Define

$$h_i = \frac{\theta_i}{2\pi}, \quad \xi_i = \frac{\lambda_i - \varphi_i}{2\pi}, \quad i = 1, \ldots, n.$$

Since $\theta_1, \ldots, \theta_n$ are linearly independent, $h_1, \ldots, h_n$ are also linearly independent. By Theorem 3.1 there exists $t \in I\!R$ and $k_i \in \mathbb{Z}$ such that

$$\left| \frac{\theta_i}{2\pi} t - \frac{\lambda_i - \varphi_i}{2\pi} - k_i \right| < \frac{\delta}{2\pi}, \ \ i = 1, \ldots, n.$$

So,

$$|\theta_i t + \varphi_i - (\lambda_i + 2k_i\pi)| < \delta, \ \ i = 1, \ldots, n.$$

Then, by (8) and the periodicity of the cosine function,

$$|\cos(\theta_i t + \varphi_i) - \cos(\lambda_i)| \leq \varepsilon.$$

From (7), the desired result follows. ∎

Let Φ be the obvious linear dipheomorphism between $[-1, 1]^n$ and Ω. By Theorem 3.2, $\{\Phi(\alpha(t)), t \in I\!R\}$ is dense in Ω. Moreover, if $\beta : (-1, 1) \to I\!R$ is one-to-one and continuous, we have that the set $\{\Phi[\alpha(\beta(t))] \mid t \in (-1, 1)\}$ is also dense in Ω. Let us define $F : (-1, 1) \to I\!R$ by

$$F(t) = f[\Phi[\alpha(\beta(t))]]. \tag{9}$$

The problem of minimizing f on Ω is equivalent to the problem of minimizing F on $(-1, 1)$ in the sense given by the following theorem.

Theorem 3.3. *Let x_* be a global minimizer of f on Ω and $\varepsilon > 0$. Then, there exists $t \in (-1, 1)$ such that*

$$F(t) < f(x_*) + \varepsilon.$$

Proof. By the continuity of f, there exists $\delta > 0$ such that $f(x) < f(x_*) + \varepsilon$ whenever $\|x - x_*\| < \delta$. Since $\{\Phi[\alpha(\beta(t))] \mid t \in (-1, 1)\}$ is dense in Ω, there exists $t \in (-1, 1)$ such that $\|\Phi[\alpha(\beta(t))] - x_*\| < \delta$. Then, $f[\Phi[\alpha(\beta(t))] < f(x_*) + \varepsilon$ as we wanted to prove. ∎

4 Global algorithm

The local Algorithm 2.1 can be very effective in many cases for finding global minimizers of the OVO problem. However, when m is large, the number of local minimizers increases dramatically. Moreover, critical points are not necessarily local minimizers and, therefore, the number of possible limit points of Algorithm 2.1 that are not global solutions is enormous.

Our strategy for solving the OVO problem consists of using the local algorithm for finding a critical point x_* and, then, trying to "escape" from this critical point using a Lissajous curve that passes through it. The linearly independent parameters $\theta_1, \ldots, \theta_n$ that define the Lissajous curves are chosen to be the square roots of the n first prime numbers. So, $\{\theta_1, \theta_2, \theta_3, \ldots\} = \{\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots\}$. Therefore, the escaping strategy is reduced to a one-dimensional tunneling procedure. See [19, 25, 26, 28], among others.

After many trials and errors, we defined the global algorithm described below. Besides the local minimization and the tunneling phases we introduced a multistart procedure for generating different initial points, defining criteria for discarding poor initial points and establishing an upper limit of one second for each call to the tunneling phase.

Algorithm 4.1

Let $k_{max} > 0$ be an algorithmic parameter. Initialize $k \leftarrow 1$, $f_{min} \leftarrow \infty$, $\mathcal{C} \leftarrow \emptyset$, $\mathcal{A} \leftarrow \emptyset$, $\delta = 0.1 \times \min\limits_{1 \le i \le n} \{u_i - l_i\}$.

Step 1. *Random choice*

If $k > k_{max}$, stop. Otherwise, choose a random (uniformly distributed) initial point $x_{I_k} \in \Omega$.

Step 2. *Functional discarding test*

Define

$$f_{max} = \max\{f(x) \mid x \in \mathcal{A}\}.$$

The probability $Prob$ of discarding x_{I_k}, is defined in the following way:

- If $f(x_{I_k}) \le f_{min}$, $Prob \leftarrow 0$.
- If $f(x_{I_k}) \ge f_{max}$ and $\mathcal{A} \ne \emptyset$, $Prob \leftarrow 0.8$.
- If $f_{min} < f(x_{I_k}) < f_{max}$ and $\mathcal{A} \ne \emptyset$, $Prob \leftarrow 0.8 \left(\dfrac{f(x_{I_k}) - f_{min}}{f_{max} - f_{min}} \right)$.

Discard the initial point x_{I_k} with probability *Prob*. If x_{I_k} was discarded, go to Step 1. Otherwise, go to Step 3.

Step 3. *Neighborhood discarding test*

Define

$$d_{min} = \min\{\|x - x_{I_k}\|_\infty \mid x \in \mathcal{C}\}, \ (\mathcal{C} = \emptyset \Rightarrow d_{min} = \infty).$$

Update $Prob$, the new probability of discarding x_{I_k}, in the following way:

- If $d_{min} \leq \delta$, $Prob = 0.8$.
- If $d_{min} > \delta$, $Prob = 0$.

Discard the initial point x_{I_k} with probability *Prob*. If x_{I_k} was discarded, go to Step 1. Otherwise, go to Step 4.

Step 4. *10-Iterations discarding test*

Perform 10 iterations of the local method (Algorithm 2.1) obtaining the iterate $x_{10,k}$. Set

$$f_{10} = f(x_{10,k}).$$

If $\mathcal{C} \neq \emptyset$, define

$$f_{aux} = f(x_{I_k}) - 0.1(f(x_{I_k}) - f_{min}).$$

Update the probability $Prob$ in the following way:

- If $f_{10} \leq f_{min}$, $Prob \leftarrow 0$.
- If $f_{10} \geq f_{aux}$, $Prob \leftarrow 0.8$.
- If $f_{min} < f_{10} < f_{aux}$, $Prob \leftarrow 0.8 \left(\dfrac{f_{10} - f_{min}}{f_{aux} - f_{min}} \right)$.

Discard x_{I_k} with probability *Prob*. If x_{I_k} was discarded, go to Step 1. Otherwise, update

$$\mathcal{A} \leftarrow \mathcal{A} \cup \{x_{I_k}\},$$

set $k \leftarrow k + 1$ and go to Step 5.

Step 5. *Local minimization*

Taking x_{I_k} as initial point, execute Algorithm 2.1 obtaining a critical point $x_{*,k}$. Update the set of critical points $\mathcal{C}$:

$$\mathcal{C} \leftarrow \mathcal{C} \cup \{x_{*,k}\}.$$

Update the best functional value:

$$f_{min} \leftarrow \min\{f_{min}, f(x_{*,k})\},$$

$$x_{min} \leftarrow x_{*,k} \ \text{ if } \ f_{min} = f(x_{*,k}).$$

Set $time \leftarrow 0$ and go to Step 6.

Step 6. *Tunneling*

Using the Lissajous curve that passes through $x_{*,k}$ and the definition (9), try to obtain $t \in (-1, 1)$ such that $F(t) < f(x_{*,k})$. If such a point is obtained, update

$$x_{I_k} \leftarrow \Phi[\alpha(\beta(t))]$$

and go to Step 5. At each step of the tunneling process, update the computer time parameter *time*. If *time* exceeds one second, go to Step 1.

The random choice at Step 1 of Algorithm 4.1 trivially guarantees that a global minimizer is found with probability 1. This is stated, for completeness, in the following theorem.

Theorem 4.1 *Let x^* be a global minimizer of (1) and let $\varepsilon > 0$ be arbitrarily small. Assume that $k_{max} = \infty$. Then, with probability 1, there exists $k \in \{1, 2, \ldots\}$ such that the point x_{min} computed at Step 5 of the algorithm satisfies $f(x_{min}) \leq f(x_*) + \varepsilon$.*

Proof. Since f is continuous there exists a ball $\mathcal{B}$ centered in x^* such that

$$f(x) \leq f(x_*) + \varepsilon \ \ \forall \ \ x \in \mathcal{B}.$$

Since the initial point at Step 1 is chosen randomly and according to the uniform distribution, the probability of choosing $x_{I_k} \in \mathcal{B}$ at a particular iteration k is strictly positive. By the structure of Steps 2–4, the probability of discarding x_{I_k} is strictly smaller than 1.

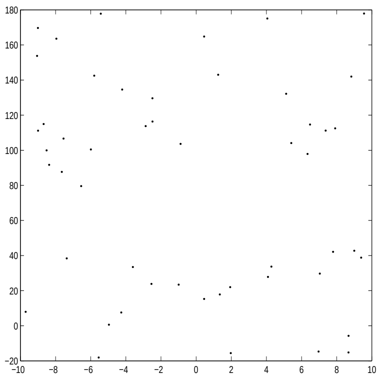
Therefore, at a fixed iteration k, a point belonging to the ball $\mathcal{B}$ is chosen as the initial point for Algorithm 2.1 with positive probability. Since Algorithm 2.1 does not increase the objective function value, it follows that, given k, a point $x_{min} \in \mathcal{B}$ is computed at Step 5 with positive probability. The desired result follows since $k_{max} = \infty$. ■

5 Hidden patterns

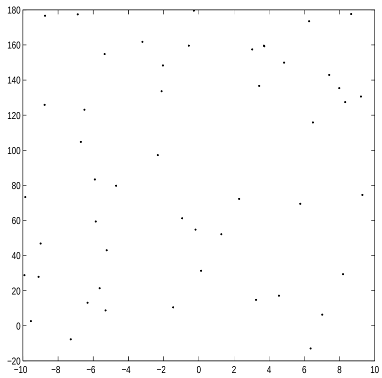
The search of hidden patterns is one of the most challenging issues in modern data mining. Many papers address the problem of hidden-patterns discovery in different areas, as Ecology [9, 24], Web-log Analysis [16], Public Health

Administration [17], Spatio-temporal Dynamics of Wave Modes [8], Art [35], Psychoanalytic Literary Criticism [27], Psychiatry [11], Social History [29], Demography [6, 7], City Systems [18] and many others.

Consider the clouds of points given in Pictures 3a and 3b. At a first sight, the two clouds of points look qualitatively similar. However, in Picture 3a there are exactly five points that lie on the same parabola whereas in Picture 3b such set of points does not exist.

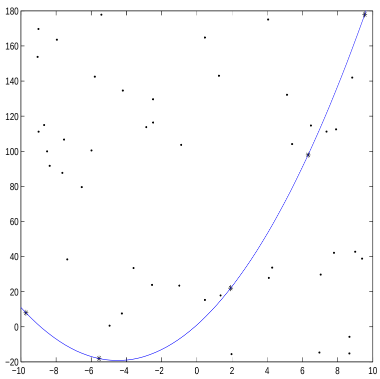


Picture 3a



Picture 3b

In Picture 4 we show again the points of Picture 3a together with the parabola that fits those special points. In situations like this, we say that Picture 3a hides the pattern of a parabola (or, simply, hides a parabola), whereas Picture 3b does not.



Picture 4

In the following section we show how the OVO problem and Algorithm 4.1 may be used to find hidden patterns of this type. Several examples presented in the following section are simple and small enough to be solved by exhaustive enumerative methods. However, if the number of data or the number of unknown parameters is moderately increased the use of combinatorial procedures is prohibitive. This is the reason why we did not consider enumerative algorithms in our analysis.

It must be mentioned that in many data-mining papers and applications, patterns are hidden in the sense that they are difficult to find, at least using standard fitting procedures. In our case patterns are hidden because they are revealed only taking into account a small amount of data, the reason being that most available information is severely corrupted.

6 Numerical experiments

Some practical features concerning the implementation of the algorithm are given below:

- As mentioned before, problem (2) is a convex optimization problem. Moreover, in our applications the constraints Ω will be linear, therefore (2) is a Linear Programming problem. For solving it we use the IMSL Library routine DDLPRS.
- The subproblems were solved exactly. This means that we used $\eta = 0$.
- The algorithmic parameters used were:

$$\theta = 0.5,\ \Delta = 1,\ \varepsilon = 10^{-3},\ \sigma_{\min} = 0.1,\ \sigma_{\max} = 0.9.$$

- In the backtracking process (4) we took $\alpha_{new} = 0.5\alpha$.
- All the numerical experiments were run on a Pentium 4, 2.4 Ghz, 1Gb RAM in double precision FORTRAN.

6.1 Finding hidden polynomials

Assume that $\{(t_1, y_1), \ldots, (t_m, y_m)\} \subset I\!R^2$ is a set of data and we know that "most of them are wrong". Nevertheless, a few of these points contain valuable uncorrupted information, represented by a low-degree polynomial $x_1 t^{n-1} + \cdots + x_{n-1} t + x_n$. Least-squares fitting of the form $y_i \approx x_1 t_i^{n-1} + \ldots + x_{n-1} t_i + x_n$ leads to disastrous results due to the overwhelming influence of outliers.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
3	10	2	-0.34	1.65	3.97	1.40E-16	4	step 1
4	150	190	1.00	-5.00	2.00	2.68E-13	233	step 1
5	250	395	1.00	-5.00	2.00	2.94E-13	590	step 1
6	200	284	1.51	0.90	6.23	0.2687	307	step 6
7	200	397	1.66	1.39	2.59	1.3313	431	step 6
8	200	376	1.54	0.84	5.14	2.3728	384	step 6

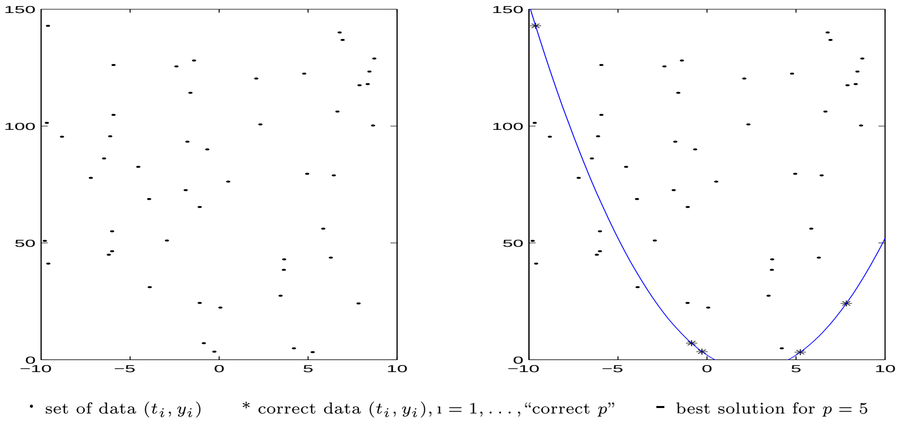


Table 1: $m = 50$, "correct p"$= 5$, $x^* = (1, -5, 2)$

The OVO approach for finding the hidden polynomial consists in defining, for each $i = 1, \ldots, m$, the error function

$$f_i(x) = (\sum_{j=1}^{n} x_j t_i^{n-j} - y_i)^2.$$

Given $p \in \{1, \ldots, m\}$, this set of functions defines an OVO problem (1) for which Algorithm 4.1 may be employed. The idea is to solve this problem for different values of p. If p is close to m we expect a large value of the OVO function at the solution found, showing that there are wrong data among the points that correspond to $f_{i_1}, \ldots, f_{i_p}$. When p is decreased, the OVO function at the solution tends to decrease as well. We expect that, when we take "the correct p", the OVO function would decrease abruptly, taking a value close to zero.

The results are summarized in Tables 1 to 6 and the corresponding pictures.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
8	50	99	-1.00	-6.00	4.00	7.65E-13	66	step 6
9	50	102	-1.00	-6.00	4.00	2.28E-12	72	step 6
10	50	93	-1.00	-6.00	4.00	4.94E-12	45	step 6
11	200	397	-0.95	-6.94	8.08	1.00	406	step 6
12	200	368	-1.04	-5.90	5.45	1.42	75	step 6
13	200	371	-0.94	-6.19	1.95	3.14	133	step 1

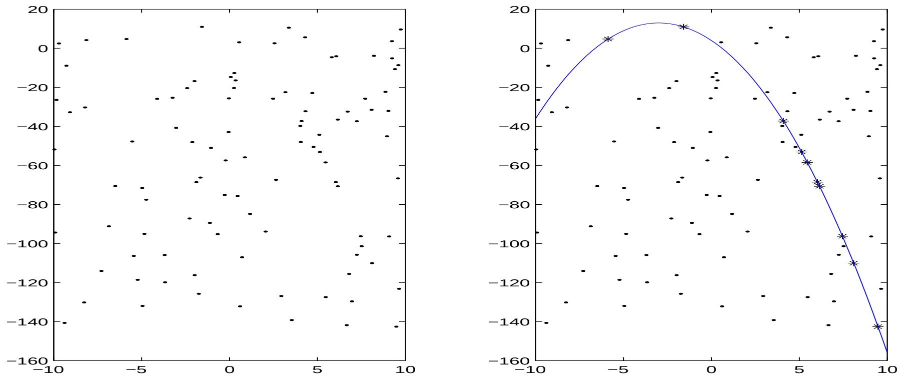


Table 2: $m = 100$, "correct p"$= 10$, $\quad x^* = (-1, -6, 4)$

In all cases we generated m random data. Ten per cent of them are "correct" in the sense that they fit exactly a previously chosen polynomial. We defined $\Omega = [-10, 10]^n$. The parameter k_{max} is reported under the second column of each table. Under the third column we report the total number of "better points" obtained by the tunneling Lissajous procedure. The last column indicates the step of Algorithm 4.1 that gave rise to the initial point that produced the solution. The penultimate column indicates the number of calls to Algorithm 2.1 that were necessary to find the best point obtained.

6.2 Finding hidden circles

The experiments are entirely analogous to the ones reported for polynomials. In this case we need to estimate three parameters $x_1, x_2, x_3 \in \Omega$ where

$$\Omega = \{x \in I\!R^3 \mid -10 \le x_1, x_2 \le 10, \ 0 \le x_3 \le 10\}.$$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
13	50	91	-1.00	1.00	-6.00	9.92E-14	33	step 6
14	50	95	-1.00	1.00	-6.00	5.18E-14	36	step 6
15	100	177	-1.00	1.00	-6.00	2.45E-14	221	step 6
16	200	350	-1.00	1.00	-5.96	0.0272	14	step 6
17	200	358	-1.00	1.00	-6.18	0.0339	179	step 6
18	200	332	-1.00	0.97	-6.19	0.0865	288	step 6

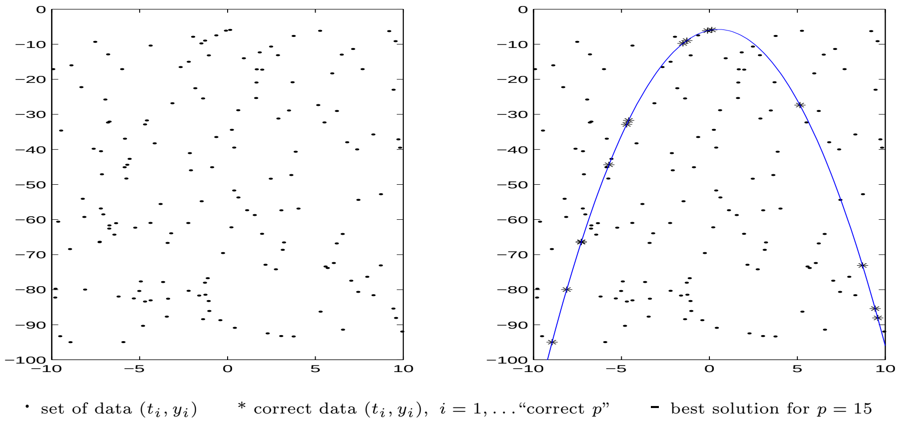


Table 3: $m = 150$, "correct p"$= 15$, $\quad x^* = (-1, 1, -6)$

The center of the unknown circle is (x_1, x_2) and its radius is x_3. The functions f_i are:

$$f_i(x) = [(t_i - x_1)^2 + (y_i - x_2)^2 - x_3^2]^2.$$

The results, following the same conventions as before, are summarized in Tables 7, 8 and 9 and the corresponding pictures.

6.3 Finding hidden "bananas"

The hidden pattern is a curve in the ty-plane, of the form

$$(y - x_2 - (t - x_1)^2)^2 + (1 - (t - x_1))^2 = x_3,$$

where x_1, x_2 and x_3 are the parameters that we need to estimate. We defined

$$\Omega = \{x \in I\!R^3 \mid -10 \le x_1, x_2 \le 10,\ 0 \le x_3 \le 20\}.$$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
3	10	4	1.21	3.34	-5.84	1.36	8.41E-20	4	step 6
4	50	74	1.24	3.55	-5.90	-0.58	1.14E-16	84	step 1
5	50	85	1.00	5.00	3.00	-7.00	1.26E-14	101	step 6
6	200	466	1.05	5.10	-0.95	-10.0	140.50	309	step 6
7	200	397	0.97	4.76	5.77	-10.0	679.53	119	step 6
8	200	450	0.48	-0.52	-7.83	-10.0	3655.75	627	step 6

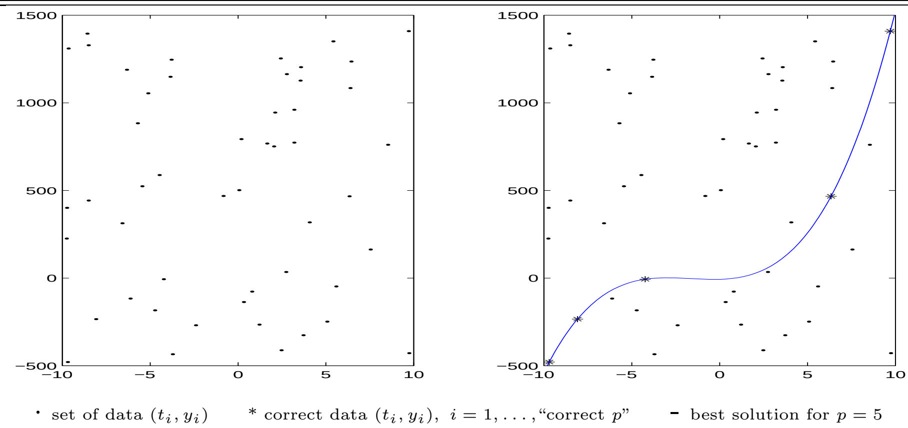


Table 4: $m = 50$, "correct p"$= 5$, $x^* = (1, 5, 3, -7)$

And, for each $i = 1, \ldots, m$ the error function is:

$$f_i(x) = [(y_i - x_2 - (t_i - x_1)^2)^2 + (1 - (t_i - x_1))^2 - x_3]^2.$$

The results, following the same conventions as before, are summarized in Tables 10, 11 and 12 and the corresponding pictures.

6.4 Finding hidden ellipses

In this section the hidden patterns are ellipses:

$$\frac{(t_i - x_1)^2}{x_3^2} + \frac{(y_i - x_2)^2}{x_4^2} = 1.$$

In this case we need to estimate four parameters x_1, x_2, x_3 and $x_4 \in \Omega$ where

$$\Omega = \{x \in I\!R^4 \mid -10 \le x_1, x_2 \le 10,\ 0 \le x_3, x_4 \le 10\}.$$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
8	100	181	-1.00	3.00	3.00	9.00	3.46E-17	144	step 6
9	50	100	-1.00	3.00	3.00	9.00	9.15E-15	93	step 6
10	50	80	-1.00	3.00	3.00	9.00	4.51E-14	23	step 6
11	200	407	-1.00	3.00	2.97	8.86	0.0272	369	step 6
12	200	414	-0.98	3.21	0.80	1.73	118.89	115	step 6
13	200	429	-0.98	3.46	2.17	-5.14	207.12	527	step 6

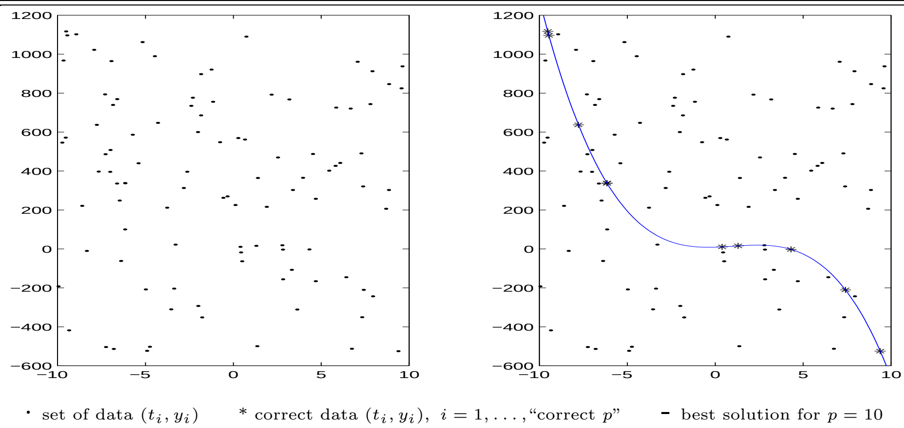


Table 5: $m = 100$, "correct p" $= 10$, $x^* = (-1, 3, 3, 9)$

Therefore, the error functions are:

$$f_i(x) = \left[\frac{(t_i - x_1)^2}{x_3^2} + \frac{(y_i - x_2)^2}{x_4^2} - 1 \right]^2 .$$

The results, following the same conventions as before, are summarized in Tables 13, 14 and 15 and the corresponding pictures.

We used the third problem of Table 15 for comparing the efficiency of Algorithm 4.1 with respect to a straightforward multistart strategy. The result reported in Table 15 was obtained using 25 minutes of CPU time. However, the pure random multistart method, which uses the local algorithm without discarding and tunneling, did not obtain a solution of similar quality after 12 hours of computation.

6.5 Finding a hidden polynomial-trigonometric function

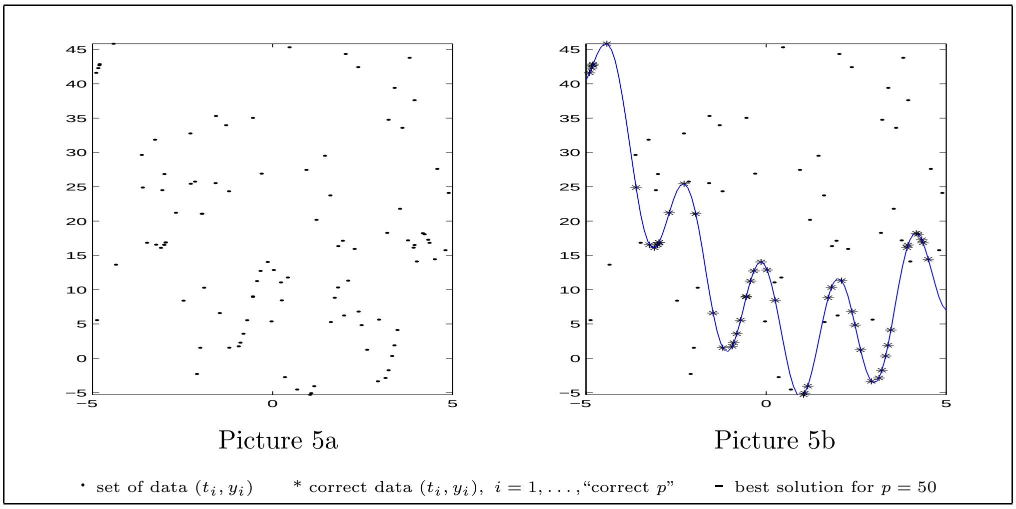
Consider the points in Picture 5a. We wish to fit a polynomial-trigonometric function of the form

$$y(x,t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3 + x_5 \cos(x_6 t + x_7) + x_8 \sin(x_9 t + x_{10}).$$

Consequently, the error functions are:

$$f_i(x) = [y_i - y(x,t_i)]^2.$$

Taking $p = 50$ we obtained using Algorithm 4.1 and after 2 hours of CPU time the function given in Picture 5b. Observe that since we need at least 10 points to fit this function and $C_{10}^{100} \cong 1.7 \times 10^{13}$, the use of enumerative schemes is completely impossible.



Picture 5: $m = 100$, "correct p"$= 50$

7 Conclusions

Order-Value Optimization is a serious global minimization problem with important applications, many of which remain to be discovered. In this paper we emphasized the application to finding hidden patterns in the presence of massive corrupted data. The local algorithm introduced in [2] is not efficient enough to cope the problem of multiple critical points, therefore the definition of a globalization strategy was necessary. In order to preserve the local

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
13	50	80	-1.00	3.00	7.00	0.00	3.24E-16	914	step 6
14	50	82	-1.00	3.00	7.00	0.00	8.30E-12	69	step 6
15	100	210	-1.00	3.00	7.00	0.00	1.45E-12	213	step 6
16	200	413	-1.01	3.06	7.79	-1.75	12.04	34	step 6
17	200	447	-1.01	2.96	8.57	0.66	37.10	61	step 6
18	200	409	-0.98	3.10	4.89	-2.21	55.24	225	step 6

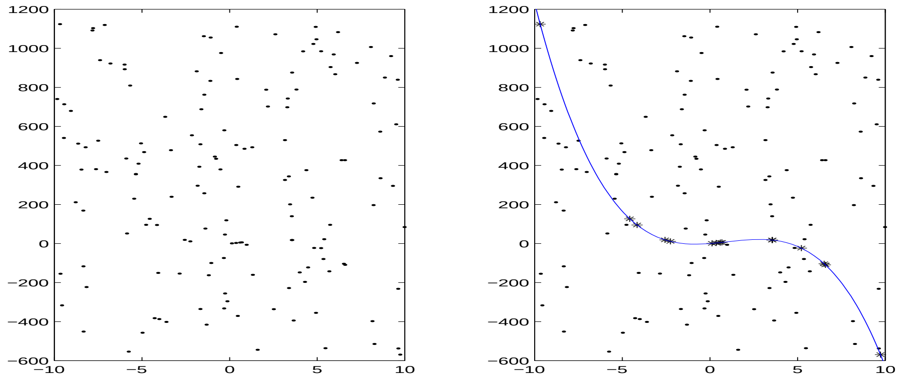


Table 6: $m = 150$, "correct p"$= 15$, $x^* = (-1, 3, 7, 0)$

efficiency of Algorithm 2.1, our global strategy incorporates multiple starts with discarding strategies and a one-dimensional tunneling procedure for escaping from critical points, based on Lissajous harmonic oscillator dense curves.

Although we do not have rigorous theoretical arguments to support the point of view that this strategy is the best possible for the global OVO problem (and probably it is not) a rather extensive numerical experimentation (some of which is reported here) suggests that we are not far from discovering a satisfactory methodology for many practical problems. In particular, the use of Lissajous curves has been a pleasant experience. The idea of transforming n-dimensional minimization problems into one-dimensional ones by means of dense curves is not new but the theoretical question about which is "the best" curve to fill the n-dimensional box does not seem to be explicitly formulated. Even the criteria that should define "the best" filling

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
3	50	3	-4.99	-4.81	5.68	3.48E-19	4	step 1
4	2800	3002	5.00	-3.00	7.00	8.77E-18	5670	step 6
5	200	263	5.00	-3.00	7.00	2.93E-16	378	step 1
6	200	302	5.66	3.04	4.32	0.01	363	step 6
7	200	288	5.71	3.09	4.29	0.08	341	step 6
8	200	315	0.01	1.50	5.63	0.18	366	step 6

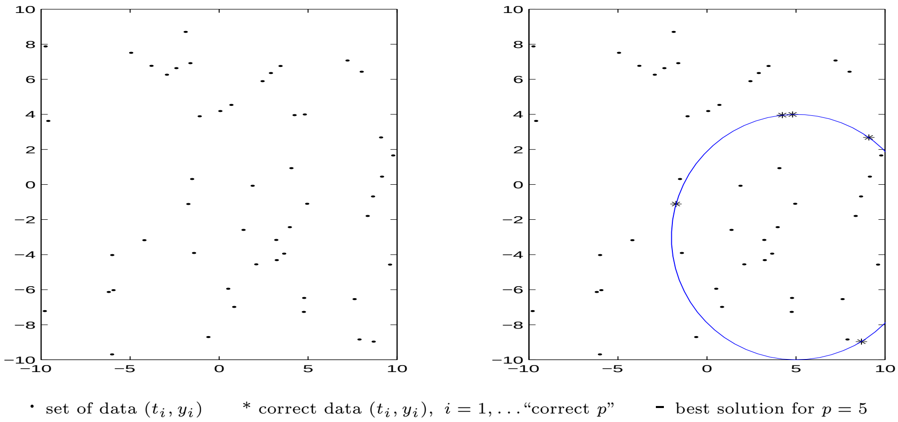


Table 7: $m = 50$, "correct p"$= 5$, $x^* = (5, -3, 7)$

curve are not completely clear. Although we feel that Lissajous curves satisfy many of these (non-formulated) criteria, we would like to point out the relevance of future theoretical research on this subject.

Valuable research on solving multi-dimensional multi-extremal optimization problem employing Peano-space-filling curves [4, 5, 10, 21, 30, 31, 32, 33, 34] may complement our Lissajous-based approach. A common drawback of space-filling algorithms is that closeness of points in the multidimensional space does not correspond to closeness in the corresponding one-dimensional interval. Strongin [33] introduced an attractive scheme which allows one to reflect, in the reduced one-dimensional problem, some information on the nearness of points in the multidimensional domain. His ideas should be adapted to our scheme in future research. We also need to improve the escaping procedure, which in the present implementation is rather naive.

Finally, we would like to mention that the application of the hidden-

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
8	50	89	1.00	-2.00	5.00	1.21E-15	78	step 6
9	300	543	1.00	-2.00	5.00	1.35E-17	781	step 6
10	50	88	1.00	-2.00	5.00	5.11E-17	109	step 6
11	200	382	0.92	-1.99	5.00	0.6348	355	step 6
12	200	407	1.00	-2.00	4.88	1.4563	527	step 6
13	200	371	1.00	-2.00	4.87	1.6275	185	step 6

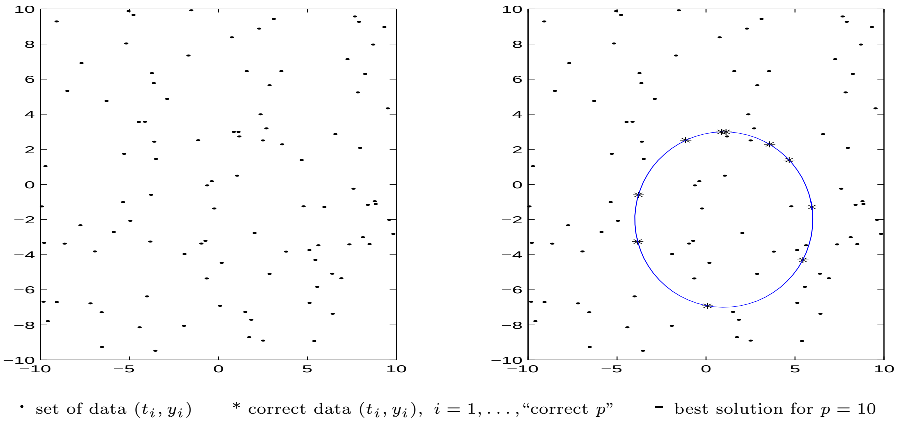


Table 8: $m = 100$, "correct p"$= 10$, $x^* = (1, -2, 5)$

pattern technology introduced in this paper to the Common Reflection Surface (CRS) problem [3, 20] has been suggested by Lúcio T. Santos and other members of the Computational Geophysics Group at the University of Campinas. Advances on this research will be reported in the near future.

A particularly interesting field of future research concerns the use of Order-Value Optimization for training neural networks in the presence of a large number of corrupted data.

Acknowledgement We are indebted to Lúcio T. Santos for his careful reading of the first draft of this paper and to Leo Liberti for suggesting many improvements.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
13	300	567	-3.00	1.00	7.00	1.52E-15	815	step 6
14	100	174	-3.00	1.00	7.00	1.99E-13	256	step 6
15	100	173	-3.00	1.00	7.00	5.98E-16	241	step 6
16	200	360	-3.00	1.04	7.00	0.3760	287	step 6
17	200	349	-3.07	1.68	7.69	3.3274	228	step 6
18	200	363	-3.05	1.69	7.69	3.6929	105	step 6

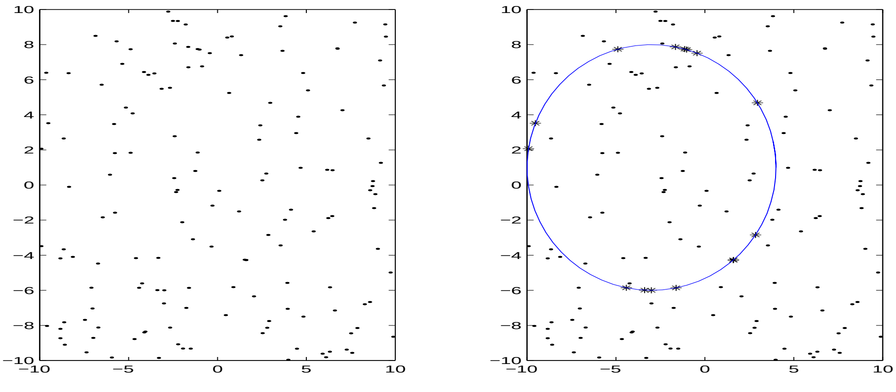


Table 9: $m = 150$, "correct p"$= 15$, $\quad x^* = (-3, 1, 7)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
3	50	9	4.70	5.73	5.82	3.52E-18	53	step 1
4	1000	1258	1.00	-4.00	7.00	1.38E-16	1828	step 6
5	1000	1899	1.00	-4.00	7.00	4.80E-17	2141	step 6
6	1000	2114	1.86	-5.49	12.9	0.0857	2835	step 6
7	1000	2284	1.89	-5.48	13.0	0.2796	2081	step 6
8	1000	2459	1.87	-5.52	13.2	0.4496	551	step 6

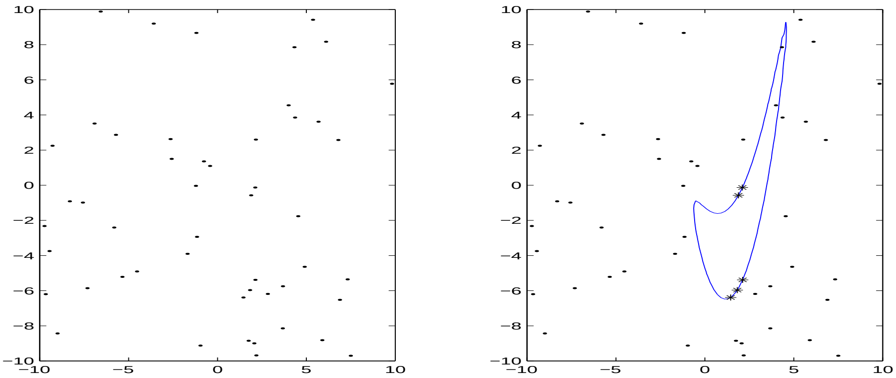


Table 10: $m = 50$, "correct p"$= 5$, $x^* = (1, -3, 7)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
8	500	1018	-1.00	-7.00	9.00	1.24E-05	403	step 6
9	500	1013	-1.00	-7.00	8.99	6.66E-05	40	step 6
10	500	1152	-1.00	-7.00	9.00	2.28E-08	760	step 6
11	500	1102	-1.00	-7.00	9.00	0.0001	695	step 6
12	500	1068	-1.00	-6.95	9.01	0.0646	458	step 6
13	500	1103	-1.00	-6.89	8.98	0.3416	1263	step 1

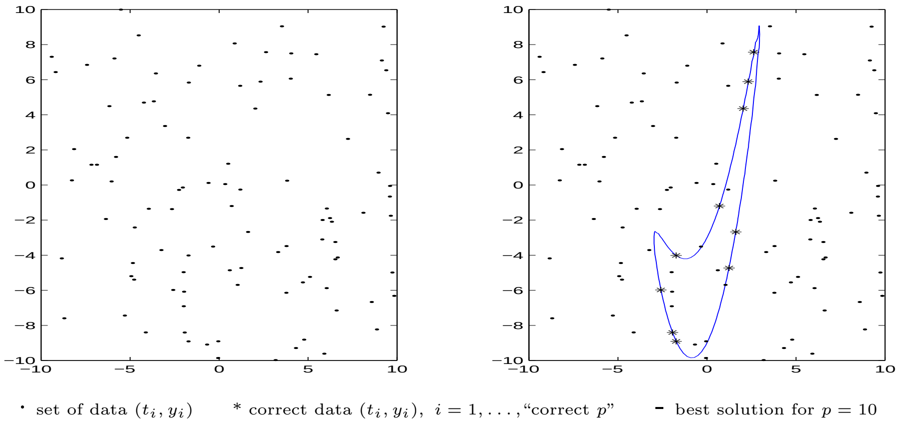


Table 11: $m = 100$, "correct p"$= 10$, $\quad x^* = (-1, -7, 9)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 2.1	Successful Step
13	100	216	0.00	-6.00	15.0	1.45E-15	107	step 6
14	100	207	-0.00	-6.00	15.0	1.31E-16	41	step 6
15	100	221	0.00	-6.00	15.0	2.49E-16	70	step 6
16	200	427	-0.00	-6.00	14.7	0.0675	169	step 6
17	200	481	-0.01	-6.01	15.5	0.6014	458	step 6
18	200	450	-0.02	-6.07	15.2	0.8290	515	step 6

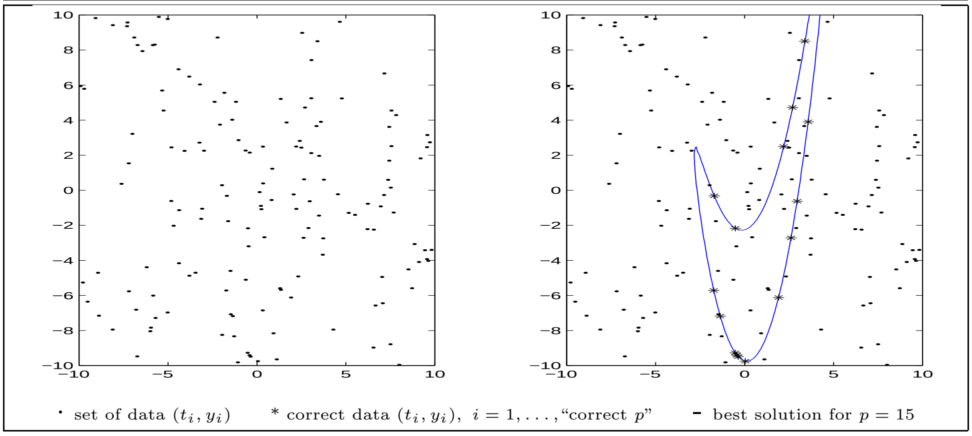


Table 12: $m = 150$, “correct p”$= 15$ $x^* = (0, -6, 15)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
8	1819	4181	0.00	4.00	9.00	5.00	1.73E-13	5063	step 6
9	500	1151	0.00	4.00	9.00	5.00	3.18E-13	1587	step 6
10	500	1091	0.00	4.00	9.00	5.00	2.39E-13	993	step 6
11	500	1218	-0.02	4.01	9.12	4.94	0.0006	1601	step 6
12	500	1226	-0.08	4.03	9.12	4.93	0.0008	1229	step 6
13	500	1183	0.31	4.22	8.14	6.06	0.0033	303	step 6

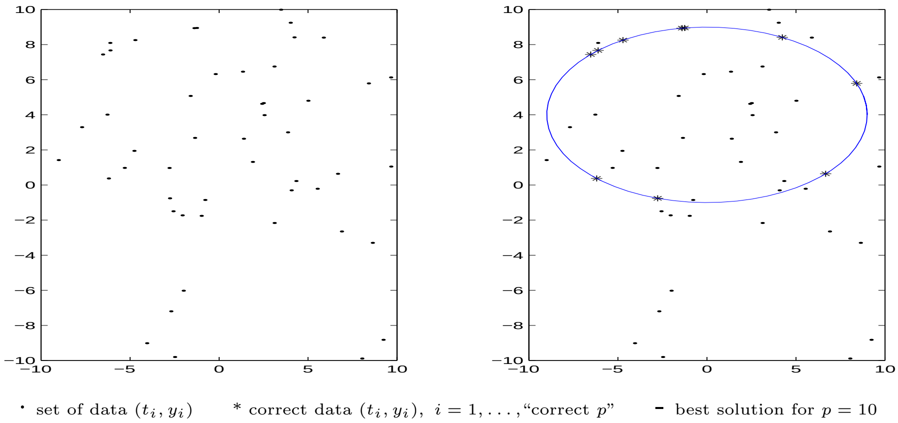


Table 13: $m = 50$, "correct p"$= 10$, $x^* = (0, 4, 9, 5)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
18	1000	3008	0.01	-2.00	9.00	6.99	7.07E-6	1085	step 6
19	1000	2954	0.00	-2.00	8.99	7.00	6.00E-6	3057	step 1
20	6000	17667	-0.00	-2.00	9.00	7.00	2.58E-6	23201	step 6
21	1000	2966	0.01	-1.98	8.98	6.99	7.76E-5	3914	step 6
22	1000	2950	0.03	-1.99	8.93	7.04	0.0002	229	step 6
23	1000	2859	0.00	-1.99	8.90	7.04	0.0004	1728	step 6

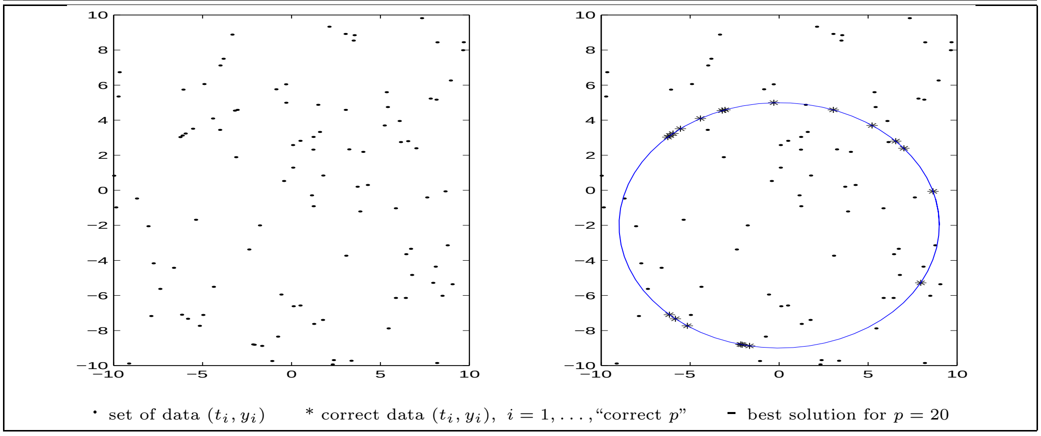


Table 14: $m = 100$, “correct p”$= 20$, $x^* = (0, -2, 9, 7)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 2.1	Successful Step
28	1000	3494	0.97	-0.00	7.99	9.00	7.43E-5	1660	step 6
29	1000	3428	1.00	-0.02	8.00	9.00	2.35E-5	1959	step 6
30	1000	3311	1.00	-0.00	8.00	8.99	3.85E-6	1492	step 6
31	1000	3388	1.02	0.01	7.96	8.99	0.0002	333	step 6
32	1000	3376	1.04	0.01	8.01	9.00	0.0003	4149	step 6
33	1000	3338	1.00	0.02	8.03	9.02	0.0004	1761	step 6

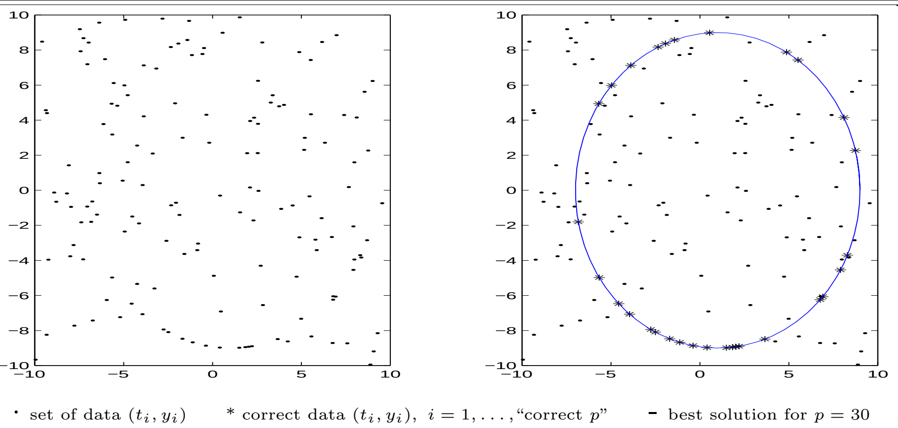


Table 15: $m = 150$, "correct p"$= 30$, $x^* = (1, 0, 8, 9)$

References

[1] R. Andreani, C. Dunder, and J. M. Martínez. Nonlinear-Programming reformulation of the Order-Value Optimization problem. Technical Report MCDO 22/05/01, Department of Applied Mathematics, University of Campinas, 2001. To appear in *Mathematical Methods of Operations Research* (2005).

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

[3] R. Biloti, L. T. Santos, and M. Tygel. Multi-parametric traveltime inversion. *Stud. geophysica et geodaetica*, 46:177–192, 2002.

[4] A. R. Butz. Space filling curves and mathematical programming. *Inform. Control*, 12(4):314–330, 1968.

[5] A. R. Butz. Alternative algorithm for Hilbert's space filling curve. *IEEE Trans. Computers*, C-20(4):424–426, 1971.

[6] E. Cam and J. Y. Monnat. Stratification based on reproductive state reveals contrasting patterns of age-related variation in demographic parameters in the kittiwake. *OIKOS*, 90:560–574, 2000.

[7] L. L. Cavalli-Sforza. Genes, peoples, and languages. *Proceedings of the National Academy of Sciences of the United States of America*, 94:7719–7724, 1997.

[8] A. Dinklage and C. Wilke. Spatio-temporal dynamics of hidden wave modes in a dc glow discharge plasma. *Physics Letters A*, 277:331–338, 2000.

[9] X. Espadaler and C. Gómez. The species body-size distribution in Iberian ants is parameter independent. *Vie et Milieu*, 52:103–107, 2002.

[10] V. P. Gergel, L. G. Strongin, and R. G. Strongin. The vicinity method in pattern recognition. *Engineering Cybernetics.* (Transl. from Izv. Acad. *Nauk* USSR, Techn, Kibernetika 4:14-22, 1987).

[11] R. Gronner and L. Manheim. Hidden patterns. *Arch. Gen. Psychiat.*, 17:248, 1967.

[12] J. Haag. *Oscillatory Motions.* Wadsworth, Belmont, California, 1962.

[13] E. Hewitt and K. A. Ross. *Abstract Harmonic Analysis I.* Springer-Verlag, Berlin, 1963.

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

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

[16] K. P. Joshi, A. Joshi, and Y. Yesha. On using a warehouse to analyze web logs. *Distributed and Parallel Databases*, 13:161–180, 2003.

[17] H. C. Koh and S. K. Leong. Data mining applications in the context of casemix. *Annals Academy of Medicine Singapore*, 30:41–49, 2001.

[18] J. Kropp. A neural network approach to the analysis of city systems. *Applied Geography*, 18:83–96, 1998.

[19] A. V. Levy and A. Montalvo. The tunneling algorithm for the global minimization of functions. *SIAM J. Sci. Stat. Comp.*, 6:15–29, 1985.

[20] J. Mann, R. Jager, T. Muller, G. Hocht, and P. Hubral. Common-reflection-surface stack - a real data example. *Journal of Applied Geophysics*, 42, 3-4:301–318, 1999.

[21] D. L. Markin and R. G. Strongin. A method for solving multi-extremal problems with non-convex constraints, that uses a priori information about estimates of the optimum. *U.S.S.R. Comput. Maths. Math. Phys.*, 27(1):33–39, 1987. Pergamon Press 1988 (*Zh. vychisl. Mat. mat. Fiz.* 27(1):52-62, 1987).

[22] J. M. Martínez. Inexact restoration: advances and perspectives. Invited talk at *Workshop on Control and Optimization.* Erice, Italy, July, 2001.

[23] J. M. Martínez and E. A. Pilotta. Inexact restoration methods for nonlinear programming: advances and perspectives. To appear in *Optimization and Control with applications*, edited by L. Q. Qi, K. L. Teo and X. Q. Yang. Kluwer Academic Publishers.

[24] M. May and L. Ragia. Spatial subgroup discovery applied to the analysis of vegetation data. *Lecture Notes in Artificial Intelligence*, 2569:49–61, 2002.

[25] D. V. Nichita, S. Gómez, and E. Luna-Ortiz. Multiphase equilibria calculation by direct minimization of Gibbs free energy with a global optimization method. *Comput. Chem. Eng.*, 26:1703–1724, 2002.

[26] D. V. Nichita, S. Gómez, and E. Luna-Ortiz. Multiphase equilibria calculation by direct minimization of Gibbs free energy using the tunneling global optimization method. *J. Can. Petrol. Technol.*, 43:13–16, 2004.

[27] N. H. Pronko and L. Manheim. Hidden patterns - Studies in psychoanalitic literary criticism. *Psychol. Rec.*, 17:283–&, 1967.

[28] D. Romero, C. Barron, and S. Gómez. The optimal geometry of Lennard-Jones clusters: 148-309. *Comput. Phys. Commun.*, 123:87–96, 1999.

[29] P. Street. The logic and limits of "plant loyalty": Black workers, white labor, and corporate racial paternalism in Chicago's stockyards, 1916-1940. *Journal of Social History*, 29:659–&, 1996.

[30] R. G. Strongin. On the convergence of an algorithm for finding a global extremum. *Engineering Cybernetics*, 11:549–555, 1973.

[31] R. G. Strongin. *Numerical Methods for Multi-extremal Problems.* Nauka, Moscow, 1978. (in Russian).

[32] R. G. Strongin. Numerical methods for multi-extremal nonlinear programming problems with nonconvex constraints. In V. F. Demyanov and D. Pallaschke, eds., *Lecture Notes in Economics and Mathematical Systems*, 225:278–282, 1984. Proceedings 1984. Springer-Verlag. IIASA, Laxenburg/Austria 1985.

[33] R. G. Strongin. Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves. *Journal of Global Optimization*, 2:357–378, 1992.

[34] R. G. Strongin and D. L. Markin. Minimization of multi-extremal functions with nonconvex constraints. *Cybernetics*, 22(4):486–493, 1986. Translated from Russian. Consultant Bureau. New York.

[35] J. W. Weaver. Hidden patterns in Joyce 'Portrait of the artist as a young man'. *South Atlantic Bulletin*, 41:63–63, 1976.