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

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

July 15, 2004

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 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 Order-Value Optimization problem

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

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

The OVO function f is continuous. 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 $p \approx 0.9m$. 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 [10].

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 [7]. Sometimes, we want to eliminate (say) the 15% larger errors because the 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 [1] a steepest-descent type method was introduced for solving (1). Before reviewing this algorithm, let us define, for all $\varepsilon > 0$, $x \in \Omega$:

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

Algorithm 1.1

Let $x_0 \in \Omega$ 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

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

Consider the subproblem

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

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

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

If $M_k(d_k) = 0$ stop.

Step 2. (Steplength calculation)

Set $\alpha \leftarrow 1$.

If

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

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

$$\text{Minimize } w$$

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

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

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 $x, y \in \Omega$, $j = 1, \ldots, m$, 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 $\varepsilon-$ 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 [1].

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.*

2 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:

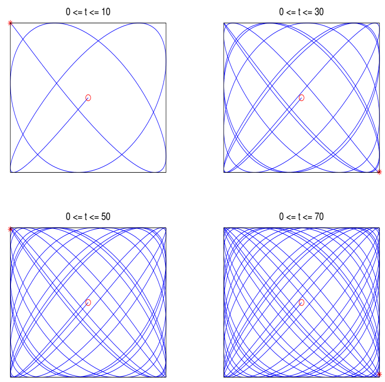
$$\ddot{x}_i + \theta_i^2 x_i = 0, \quad i = 1, \dots, n, \tag{5}$$

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.

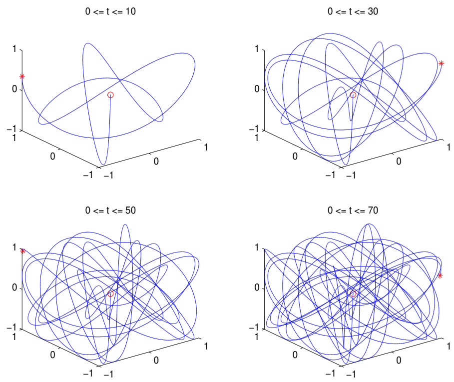
The solutions of (5) are:

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

The trajectory defined by each solution (6) is called *Lissajous curve*. See [8] pp 36. In Pictures 1 and 2 we show examples of these curves for $n = 2$ and $n = 3$ respectively. 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 Ω.



Picture 1



Picture 2

We need a previous definition.

Definition. We say $\theta_1, \dots, \theta_n \in I\!R$ 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 $r_1 = \cdots = r_n = 0$.

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

$$|h_j t - \xi_j| < \epsilon \quad (mod \quad 1) \ \forall \ j = 1, \ldots, n.$$

Proof. See [9] pp. 431-437. ∎

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

Proof. Let $\epsilon > 0$ be an arbitrary positive real number . Given $x^* = (\alpha_1^*, \ldots, \alpha_n^*) \in [-1,1]^n$, we want to obtain a $t \in I\!R$ such that $\| \alpha(t) - x^* \|_\infty < \epsilon$. Let $\lambda_1, \ldots, \lambda_n$ be real numbers such that $cos(\lambda_i) = \alpha_i^*, i = 1, \ldots, n$. Thus, we want to obtain a $t \in I\!R$ such that,

$$|cos(\theta_i t + \varphi_i) - cos(\lambda_i)| < \epsilon, \quad i = 1, \ldots, n. \tag{7}$$

Since the cosine function is continuous and 2π - periodic, the problem (7) is equivalent to: given $\tilde{\epsilon} > 0$, we look for a $t \in I\!R$ and $k_i \in \mathbb{Z}, i = 1, \ldots, n$, such that,

$$|(\theta_i t + \varphi_i) - (\lambda_i + 2k_i\pi)| < \tilde{\epsilon}, \quad i = 1, \ldots, n. \tag{8}$$

By putting $h_i = \dfrac{\theta_i}{2\pi}$ and $\xi_i = \dfrac{\lambda_i - \varphi_i}{2\pi}$, the Kronecker's Approximation Theorem assures that exists a $t \in I\!R$ and $k_i \in \mathbb{Z}, i = 1, \ldots, n$, such that (8) holds, and so (7). Therefore, $\| \alpha(t) - x^* \|_\infty < \epsilon$. ∎

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 $\{\Phi[\alpha(\beta(t))], t \in (-1,1)\}$ 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 $\{\Phi[\alpha(\beta(t))], t \in (-1, 1)\}$ 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. ∎

3 Global algorithm

The local Algorithm 1.1 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, therefore the number of possible limit points of Algorithm 1.1 that are not global solutions is enormous.

Our strategy for solving the OVO problem consists in 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 [14, 16, 17, 19], among others.

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

Algorithm 3.1

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

Step 1. *Random choice*

If $k > k_{max}$, stop. Else, 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 $f(x_{I_k}) \leq f_{min}$, $Prob \leftarrow 0$.
- If $f(x_{I_k}) \geq f_{max}$ and $\mathcal{A} \neq \emptyset$, $Prob \leftarrow 0.8$.
- If $f_{min} < f(x_{I_k}) < f_{max}$ and $\mathcal{A} \neq \emptyset$, $Prob \leftarrow 0.8 \left(\dfrac{f(x_{I_k}) - f_{min}}{f_{max} - f_{min}} \right)$.

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

Step 3. *Neighborhood 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. Else, go to Step 4.

Step 4. *10-Iterations Test*

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

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

and, 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. Else, 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 1.1 with the default algorithmic parameters (see [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:

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

If $f_{min} = f(x_{*,k})$,

$$x_{min} \leftarrow x_{*,k}.$$

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 computer time *time*. If *time* exceeds one second, go to Step 1.

The random choice at Step 1 of Algorithm 3.1 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

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

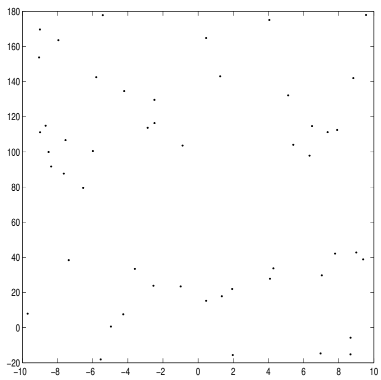
Since the initial point at Step 1 is chosen randomly and according to the uniform distribution, the probability of choosing $x_{I_k} \in \mathcal{B}$ at a particular iteration k is strictly positive. By the structure of Steps 2–4, the probability of discarding x_{I_k} is strictly smaller than 1. Therefore,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

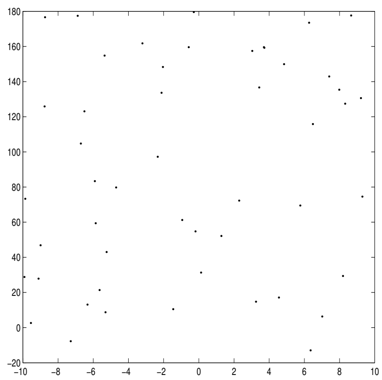
4 Hidden patterns

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 [15, 5], web-log analysis [11], Public Health administration [12], Spatio-temporal dynamics of wave modes [4], Art [22], Psychoanalytic literary criticism [18], Psychiatry [6], Social History [20], Demography [2, 3], city systems [13] and many others.

Consider the clouds of points given in Pictures 3a and 3b. At a first sight, the two clouds of points look 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.

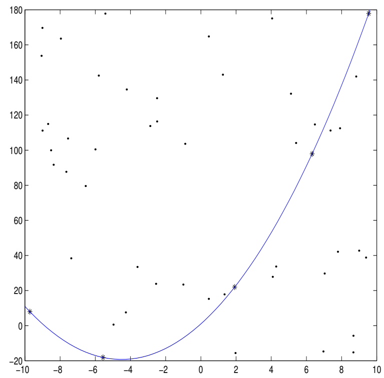


Picture 3a



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. In the following section we show how the OVO problem and Algorithm 3.1 may be used to find hidden patterns of this type.



Picture 4

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 a data, the reason being that most available information is severely corrupted.

5 Numerical experiments

5.1 Finding hidden polynomials

Assume that $\{(t_1, y_1), \ldots, (t_m, y_m)\} \subset I\!R^2$ is a set of data and that 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.

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 3.1 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 given in Tables 2 to 6 and the corresponding pictures. In all the cases we generated m random data. Ten per cent of them are "correct" in the sense that they fit exactly a previously chosen polynomial. In Algorithm 2.1 we used the parameters

$$\Delta = 1.0,\ \theta = 0.5,\ \sigma_{min} = 0.1,\ \sigma_{max} = 0.9,\ \eta = 1.0,\ \varepsilon = 0.001.$$

We defined $\Omega = [-10, 10]^n$. For solving the Linear Programming subproblem at Algorithm 2.1 we used the IMSL Library routine DDLPRS. 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 3.1 that gave rise to the initial point that produced the solution. The penultimate column indicates the number of calls to Algorithm 1.1 that were necessary to find the best point obtained.

All the numerical experiments were run on a Pentium 4, 2.4 Ghz, 1Gb RAM in double precision FORTRAN.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
3	10	2	-0.3403	1.6500	3.9696	1.40E-16	4	step 1
4	150	190	1.0000	-5.0000	1.9999	2.68E-13	233	step 1
5	250	395	0.9999	-4.9999	1.9999	2.94E-13	590	step 1
6	200	284	1.5070	0.8987	6.2336	0.2687	307	step 6
7	200	397	1.6577	1.3940	2.5871	1.3313	431	step 6
8	200	376	1.5396	0.8401	5.1352	2.3728	384	step 6

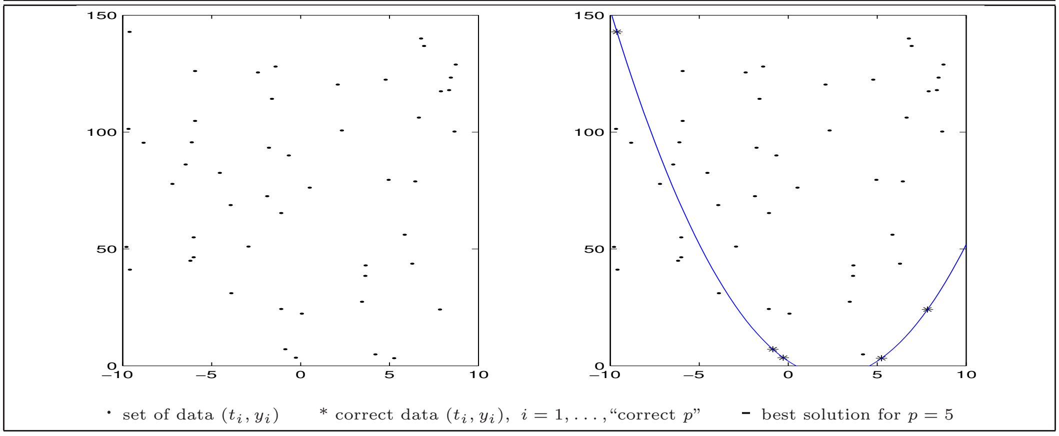


Table 1: $n = 2$, $m = 50$, "correct p"$= 5$, $x^* = (1, -5, 2)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
8	50	99	-1.0000	-5.9999	3.9999	7.65E-13	66	step 6
9	50	102	-1.0000	-5.9999	4.0000	2.28E-12	72	step 6
10	50	93	-1.0000	-6.0000	4.0000	4.94E-12	45	step 6
11	200	397	-0.9549	-6.9364	8.0750	0.9963	406	step 6
12	200	368	-1.0400	-5.8994	5.4501	1.4198	75	step 6
13	200	371	-0.9377	-6.1862	1.9483	3.1398	133	step 1

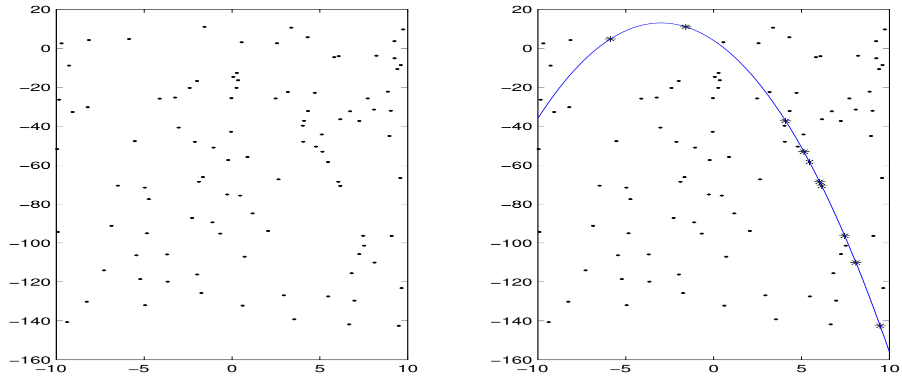


Table 2: $n = 2$, $m = 100$, "correct p"$= 10$, $x^* = (-1, -6, 4)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
13	50	91	-1.0000	0.9999	-6.0000	9.92E-14	33	step 6
14	50	95	-0.9999	0.9999	-6.0000	5.18E-14	36	step 6
15	100	177	-0.9999	0.9999	-6.0000	2.45E-14	221	step 6
16	200	350	-1.0004	0.9952	-5.9605	0.0272	14	step 6
17	200	358	-0.9962	1.0023	-6.1833	0.0339	179	step 6
18	200	332	-0.9976	0.9672	-6.1880	0.0865	288	step 6

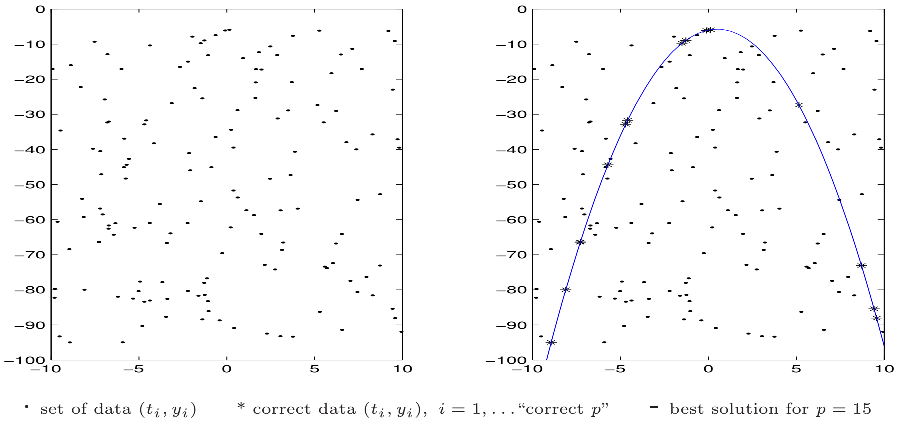


Table 3: $n = 2$, $m = 150$, "correct p"$= 15$, $x^* = (-1, 1, -6)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
3	10	4	1.2099	3.3367	-5.8411	1.3556	8.41E-20	4	step 6
4	50	74	1.2372	3.5453	-5.9048	-0.5820	1.14E-16	84	step 1
5	50	85	0.9999	4.9999	3.0000	-6.9999	1.26E-14	101	step 6
6	200	466	1.0478	5.0989	-0.9464	-10.000	140.50	309	step 6
7	200	397	0.9705	4.7553	5.7735	-10.000	679.53	119	step 6
8	200	450	0.4770	-0.5168	-7.8327	-10.000	3655.75	627	step 6

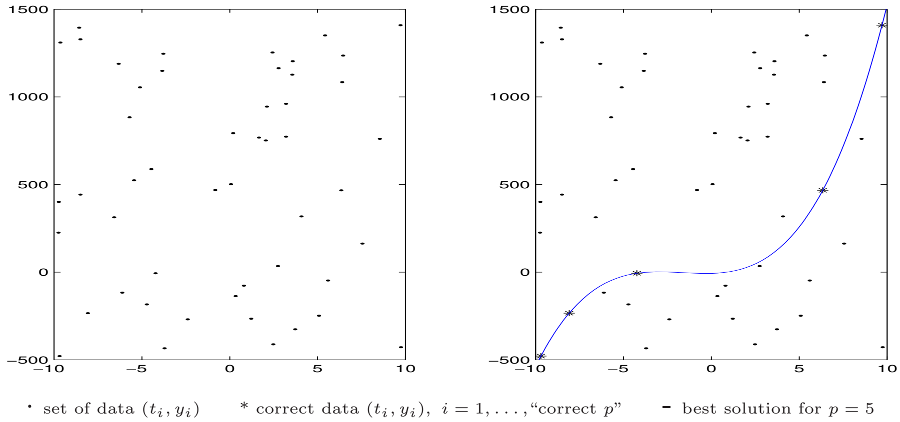


Table 4: $n = 3$, $m = 50$, "correct p"$= 5$, $x^* = (1, 5, 3, -7)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
8	100	181	-0.9999	3.0000	3.0000	8.9999	3.46E-17	144	step 6
9	50	100	-1.0000	3.0000	3.0000	8.9999	9.15E-15	93	step 6
10	50	80	-0.9999	3.0000	2.9999	9.0000	4.51E-14	23	step 6
11	200	407	-0.9996	3.0034	2.9732	8.8632	0.0272	369	step 6
12	200	414	-0.9750	3.2066	0.8018	1.7306	118.89	115	step 6
13	200	429	-0.9761	3.4551	2.1737	-5.1400	207.12	527	step 6

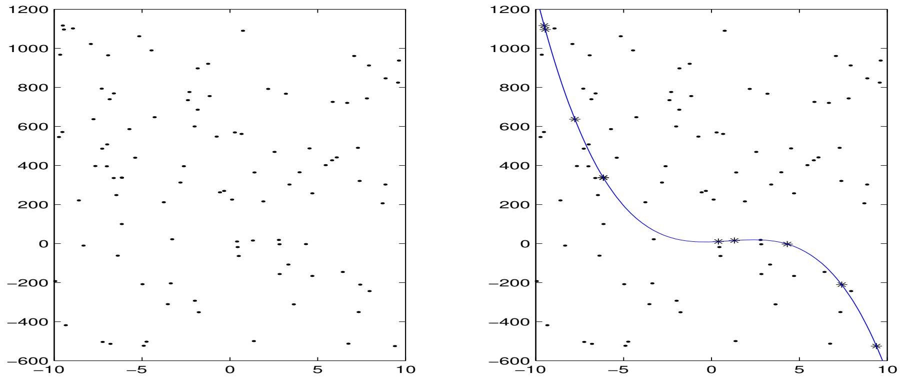


Table 5: $n = 3$, $m = 100$, "correct p"$= 10$, $x^* = (-1, 3, 3, 9)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
13	50	80	-0.9999	2.9999	7.0000	0.0000	3.24E-16	914	step 6
14	50	82	-1.0000	3.0000	6.9999	0.0000	8.30E-12	69	step 6
15	100	210	-1.0000	2.9999	7.0000	0.0000	1.45E-12	213	step 6
16	200	464	0.9952	5.0109	-6.7241	2.7233	0.5870	123	step 6
17	200	447	-1.0131	2.9643	8.5866	0.6558	37.10	61	step 6
18	200	409	-0.9776	3.1013	4.8894	-2.2077	55.24	225	step 6

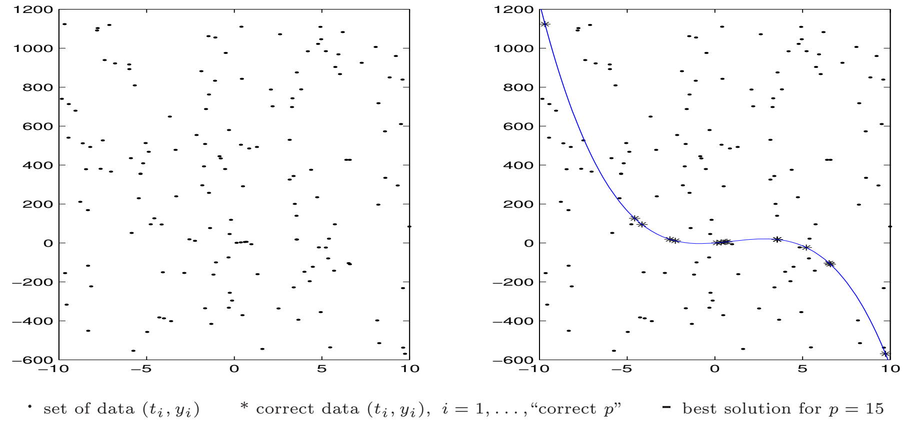


Table 6: $n = 3$, $m = 150$, "correct p"$= 15$ $x^* = (-1, 3, 7, 0)$

5.2 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. 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 are given in Tables 7, 8 and 9 and the corresponding pictures.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
3	50	3	-4.9882	-4.8114	5.6762	3.48E-19	4	step 1
4	2800	3002	5.0000	-3.0000	7.0000	8.77E-18	5670	step 6
5	200	263	5.0000	-3.0000	7.0000	2.93E-16	378	step 1
6	200	302	5.6617	3.0403	4.3245	0.0057	363	step 6
7	200	288	5.7104	3.0866	4.2891	0.0815	341	step 6
8	200	315	0.0122	1.5030	5.6278	0.1806	366	step 6

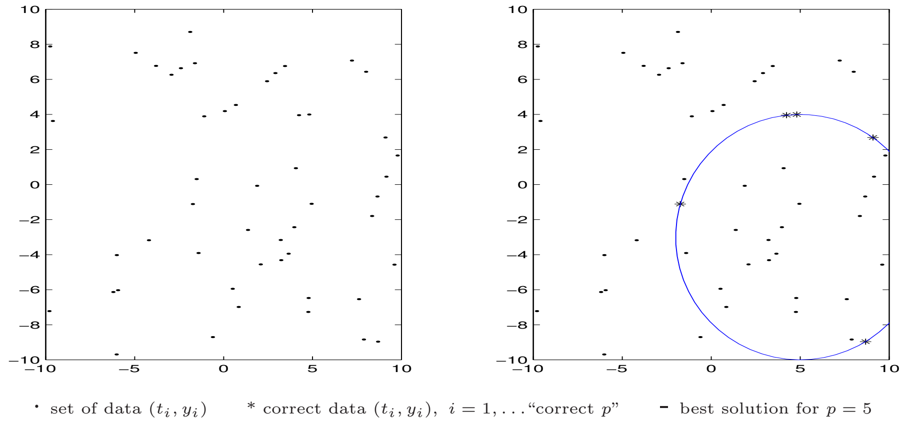


Table 7: $m = 50$, "correct p"$= 5$, $x^* = (5, -3, 7)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
8	50	89	1.0000	-1.9999	5.0000	1.21E-15	78	step 6
9	300	543	1.0000	-1.9999	4.9999	1.35E-17	781	step 6
10	50	88	0.9999	-1.9999	4.9999	5.11E-17	109	step 6
11	200	382	0.9192	-1.9863	4.9989	0.6348	355	step 6
12	200	407	0.9999	-2.0000	4.8778	1.4563	527	step 6
13	200	371	0.9999	-2.0000	4.8707	1.6275	185	step 6

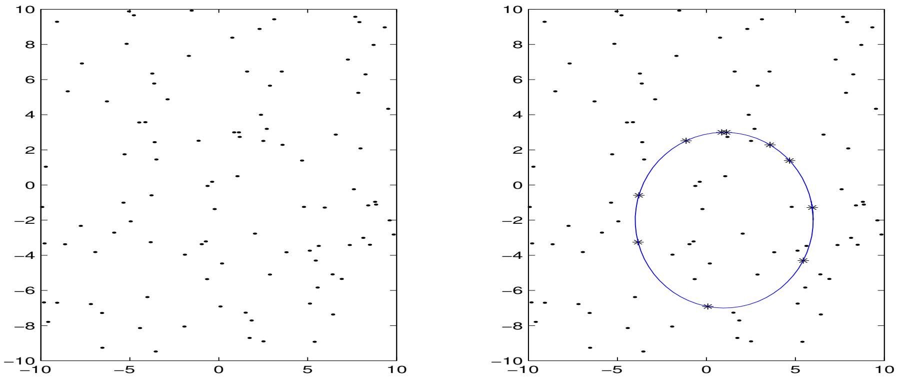


Table 8: $m = 100$, "correct p"$= 10$, $x^* = (1, -2, 5)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
13	300	567	-3.0000	0.9999	6.9999	1.52E-15	815	step 6
14	100	174	-3.0000	0.9999	7.0000	1.99E-13	256	step 6
15	100	173	-3.0000	0.9999	7.0000	5.98E-16	241	step 6
16	200	360	-3.0018	1.0443	7.0007	0.3760	287	step 6
17	200	349	-3.0717	1.6825	7.6909	3.3274	228	step 6
18	200	363	-3.0531	1.6873	7.6896	3.6929	105	step 6

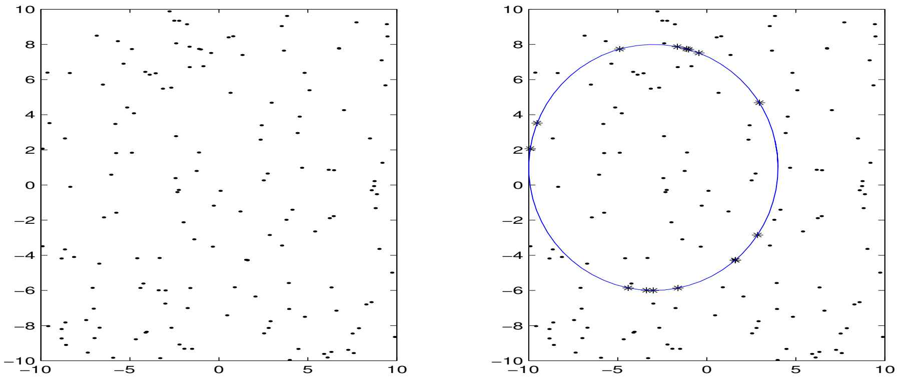


Table 9: $m = 150$, "correct p"$= 15$, $x^* = (-3, 1, 7)$

5.3 Finding multiple hidden circles I

We generated simultaneously four hidden circles centered in $(1, 1)$ with the following characteristics:

- Circle with radius 1: 5 points.
- Circle with radius 8: 12 points.
- Circle with radius 7: 13 points.
- Circle with radius 5: 20 points.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
5	5000	5896	1.0000	1.0000	4.9999	3.96E-14	42	step 6
			1.0000	0.9999	6.9999	8.76E-12	153	step 6
			0.9999	0.9999	1.0000	7.06E-14	927	step 6
10	5000	9332	0.9999	0.9999	4.9999	1.45E-12	70	step 6
			1.0000	1.0000	6.9999	4.97E-12	728	step 6
11	5000	9712	0.9999	1.0000	5.0000	3.05E-12	65	step 6
			1.0000	1.0000	6.9999	3.68E-12	137	step 6
			0.9999	0.9999	8.0000	4.67E-13	8220	step 6
12	5000	9490	0.9999	1.0000	5.0000	1.20E-11	106	step 6
			1.0000	0.9999	7.0000	1.12E-11	1737	step 6
			1.0000	0.9999	7.9999	2.41E-12	8523	step 6
13	5000	9893	0.9999	1.0000	5.0000	3.41E-12	114	step 6
			1.0000	1.0000	6.9999	2.40E-12	412	step 6
20	100	216	0.9999	0.9999	4.9999	6.66E-11	226	step 6

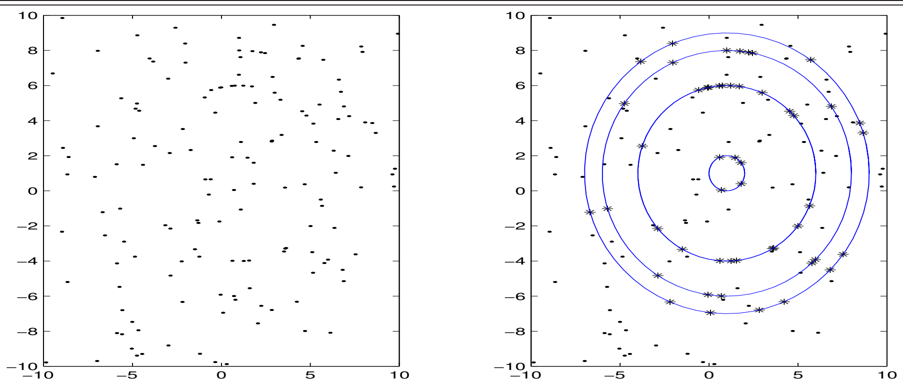


Table 10: $m = 150$

We used the algorithm for finding one hidden circle for different values of p. The problem has different solutions according to the hidden circle found. The results, including the identity of the hidden circle found, are in Table 10 and the corresponding Picture 10. We observe that the circle with radius 8 was not found for $p = 5$ and $p = 10$. As expected, the circles with radius 1, 8 and 7 were not found for $p > 5, p > 12$ and $p > 13$ respectively.

5.4 Finding multiple hidden circles II

Similarly to the previous case, we generated 5 circles with radius 4 as follows:

- Circle with center $(0, 3)$: 6 points.
- Circle with center $(-3, 3)$: 7 points.
- Circle with center $(-6, -3)$: 8 points.
- Circle with center $(6, 3)$: 9 points.
- Circle with center $(3, -3)$: 10 points.

The results are given in Table 11 and the corresponding Picture 11. Observe that, for $p = 6$ the 5 circles were found. For $p = 7$ all the circles are found except the first one. This is exactly what is expected, since the first circle contains only 6 correct points. Similarly, for $p = 8$ we find the last three circles, for $p = 9$ we find the fourth and the fifth circle and for $p = 10$ we only find the last circle.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
6	834	1166	2.9999	-3.0000	3.9999	8.49E-18	40	step 6
			6.0000	2.9999	4.0000	1.59E-15	78	step 6
			-5.9999	2.9999	3.9999	5.42E-17	1046	step 6
			-2.9999	-2.9999	4.0000	1.16E-15	1282	step 6
			0.0000	3.0000	4.0000	4.82E-15	1453	step 6
7	378	622	5.9999	3.0000	3.9999	1.55E-16	6	step 6
			-5.9999	3.0000	4.0000	3.28E-14	123	step 6
			3.0000	-2.9999	3.9999	3.72E-14	398	step 6
			-3.0000	-3.0000	3.9999	1.81E-14	503	step 6
8	376	624	6.0000	2.9999	4.0000	1.02E-14	43	step 6
			2.9999	-3.0000	3.9999	3.58E-17	170	step 6
			-6.0000	2.9999	4.0000	1.25E-16	419	step 6
9	358	642	5.9999	2.9999	4.0000	7.08E-12	50	step 6
			2.9999	-2.9999	3.9999	6.20E-17	71	step 6
10	200	392	3.0000	-2.9999	3.9999	6.16E-14	308	step 6

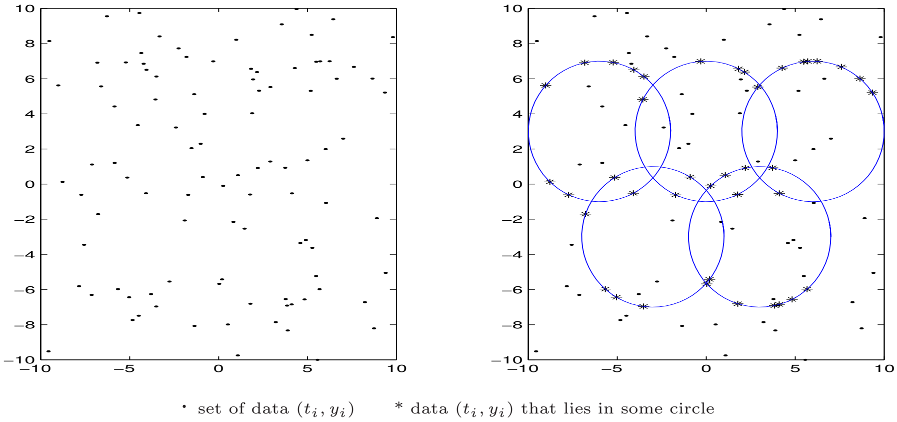


Table 11: $m = 100$

5.5 Finding hidden "bananas"

The hidden pattern is a curve in the plane t, y, 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.

Therefore, 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 are given in Tables 12, 13 and 14 and the corresponding pictures.

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
3	50	9	4.6998	5.7338	5.8187	3.52E-18	53	step 1
4	1000	1258	0.9999	-4.0000	6.9999	1.38E-16	1828	step 6
5	1000	1899	1.0000	-3.9999	7.0000	4.80E-17	2141	step 6
6	1000	2114	1.8561	-5.4885	12.9234	0.0857	2835	step 6
7	1000	2284	1.8867	-5.4765	12.9739	0.2796	2081	step 6
8	1000	2459	1.8709	-5.5161	13.1743	0.4496	551	step 6

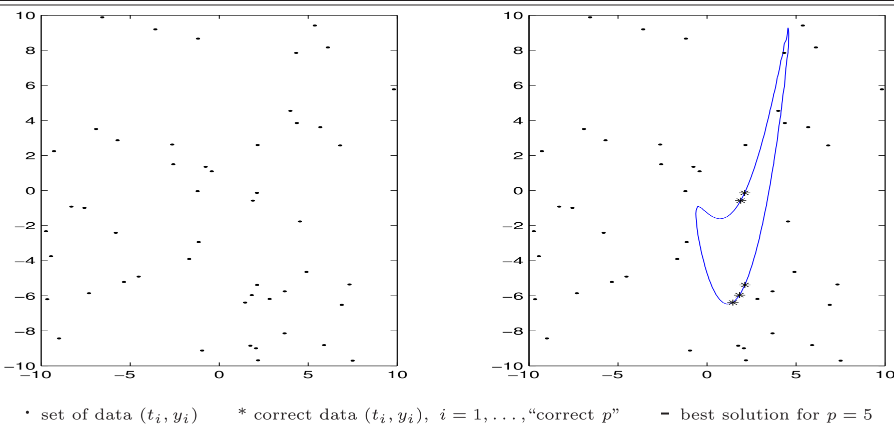


Table 12: $m = 50$, "correct p"$= 5$, $\quad x^* = (1, -3, 7)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
8	500	1018	-1.0000	-6.9994	8.9999	1.24E-05	403	step 6
9	500	1013	-1.0000	-6.9994	8.9910	6.66E-05	40	step 6
10	500	1152	-1.0000	-6.9999	9.0000	2.28E-08	760	step 6
11	500	1102	-1.0000	-6.9988	8.9963	0.0001	695	step 6
12	500	1068	-0.9979	-6.9512	9.0103	0.0646	458	step 6
13	500	1103	-0.9983	-6.8901	8.9832	0.3416	1263	step 1

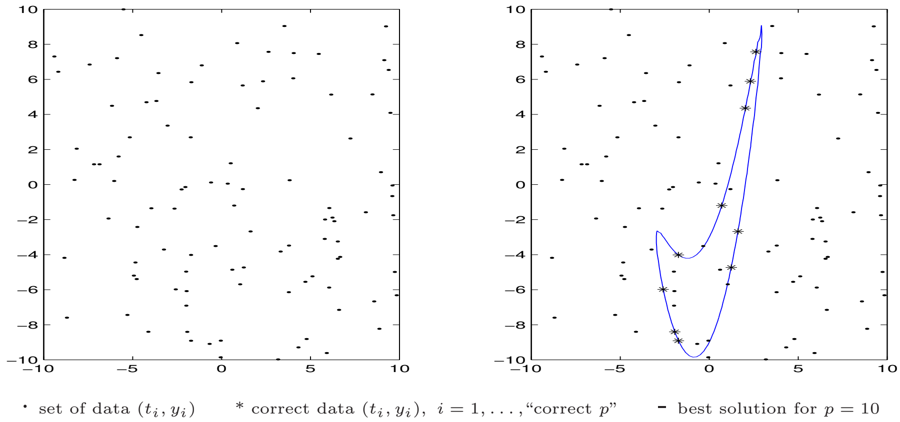


Table 13: $m = 100$, "correct p"$= 10$, $x^* = (-1, -7, 9)$

	Number of minimizations		Best solution obtained					
p	k_{max}	Successful Tunnelings	x			$f(x)$	Calls to Algorithm 1.1	Successful Step
13	100	216	0.0000	-5.9999	15.0000	1.45E-15	107	step 6
14	100	207	-0.0000	-6.0000	14.9999	1.31E-16	41	step 6
15	100	221	0.0000	-5.9999	15.0000	2.49E-16	70	step 6
16	200	427	-0.0000	-6.0001	14.7415	0.0675	169	step 6
17	200	481	-0.0108	-6.0112	15.4506	0.6014	458	step 6
18	200	450	-0.0152	-6.0666	15.1874	0.8290	515	step 6

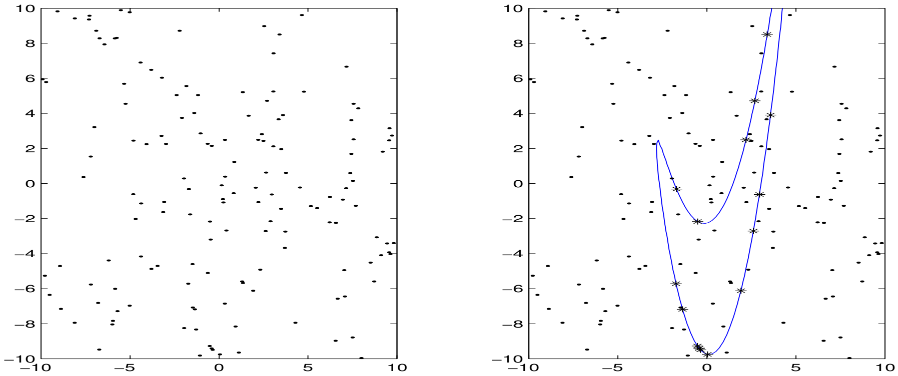


Table 14: $m = 150$, "correct p"$= 15$ $x^* = (0, -6, 15)$

5.6 Finding hidden ellipses

Here the hidden patterns are ellipses:

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

Consequently, 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 given in Tables 15, 16 and 17 and the corresponding pictures.

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
8	1819	4181	0.0000	3.9999	8.9999	5.0000	1.73E-13	5063	step 6
9	500	1151	0.0000	3.9999	9.0000	4.9999	3.18E-13	1587	step 6
10	500	1091	0.0000	3.9999	9.0000	4.9999	2.39E-13	993	step 6
11	500	1218	-0.0229	4.0050	9.1158	4.9385	0.0006	1601	step 6
12	500	1226	-0.0798	4.0292	9.1170	4.9312	0.0008	1229	step 6
13	500	1183	0.3137	4.2246	8.1414	6.0566	0.0033	303	step 6

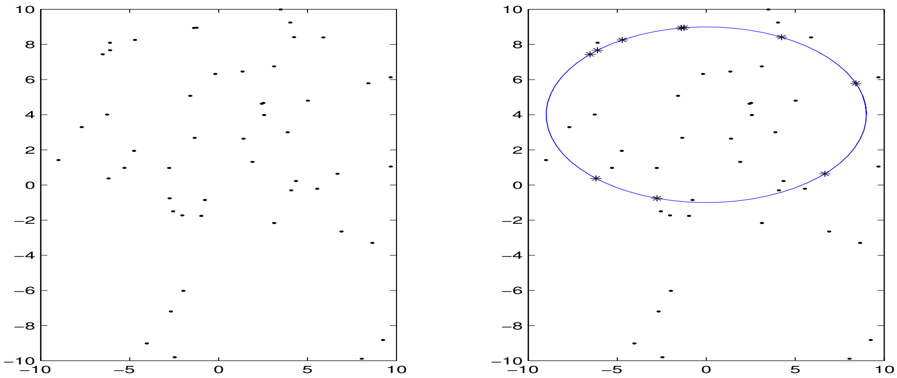


Table 15: $m = 50$, "correct p" $= 10$, $x^* = (0, 4, 9, 5)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
18	1000	3008	0.0053	-2.0021	9.0073	6.9925	7.07E-6	1085	step 6
19	1000	2954	0.0007	-1.9962	8.9942	6.9996	6.00E-6	3057	step 1
20	6000	17667	-0.0030	-2.0029	8.9964	6.9997	2.58E-6	23201	step 6
21	1000	2966	0.0069	-1.9846	8.9753	6.9872	7.76E-5	3914	step 6
22	1000	2950	0.0298	-1.9911	8.9318	7.0416	0.0002	229	step 6
23	1000	2859	0.0005	-1.9889	8.9043	7.0428	0.0004	1728	step 6

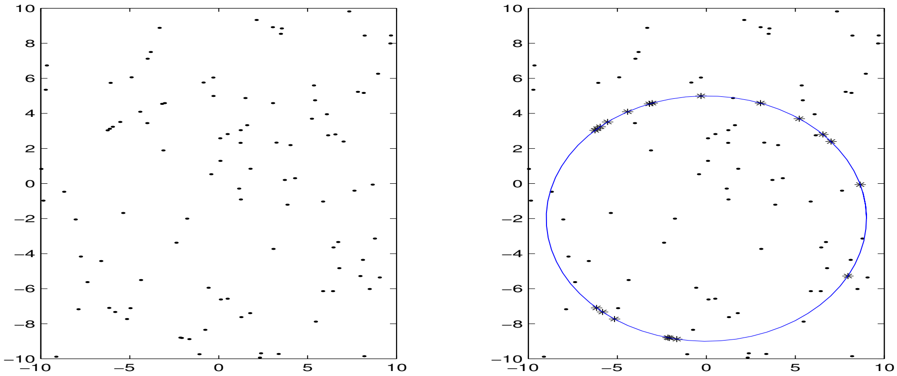


Table 16: $m = 100$, "correct p"$= 20$, $x^* = (0, -2, 9, 7)$

	Number of minimizations		Best solution obtained						
p	k_{max}	Successful Tunnelings	x				$f(x)$	Calls to Algorithm 1.1	Successful Step
28	1000	3494	0.9665	-0.0014	7.9936	9.0046	7.43E-5	1660	step 6
29	1000	3428	0.9982	-0.0219	7.9965	9.0000	2.35E-5	1959	step 6
30	1000	3311	1.0024	-0.0007	7.9990	8.9920	3.85E-6	1492	step 6
31	1000	3388	1.0151	0.0113	7.9558	8.9870	0.0002	333	step 6
32	1000	3376	1.0444	0.0142	8.0054	9.0047	0.0003	4149	step 6
33	1000	3338	1.0001	0.0151	8.0310	9.0197	0.0004	1761	step 6

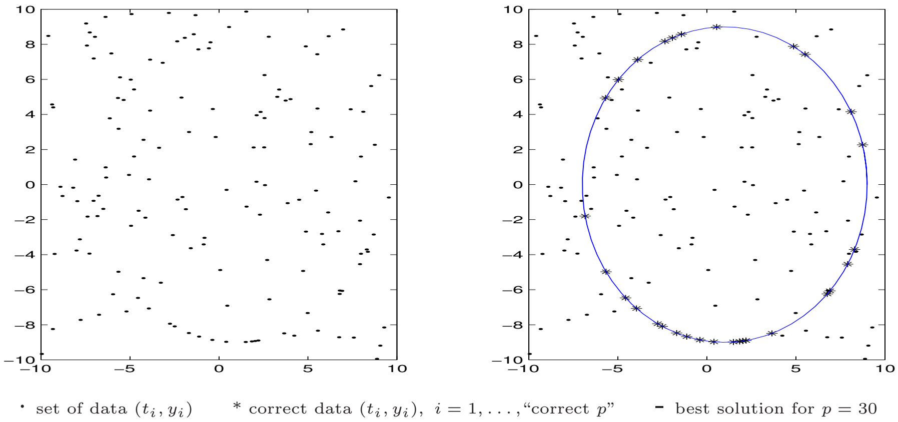


Table 17: $m = 150$, "correct p"$= 30$, $x^* = (1, 0, 8, 9)$

6 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 [1] 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 efficiency of Algorithm 1.1, our global strategy incorporates multiple starts with discarding strategies and a one-dimensional tunneling strategy 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) makes we think that we are not far from discovering a quite 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 (see, for example [21]) 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 curve are not completely clear. Although we fill that Lissajous curve satisfy many of these (non-formulated) criteria, we would like to finish this contribution pointing out the relevance of future theoretical research on this subject.

References

[1] 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).

[2] 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, pp. 560-574 (2000).

[3] L. L. Cavalli-Sforza. Genes, peoples, and languages. *Proceedings of the National Academy of Sciences of the United States of America* 94, pp. 7719-7724 (1997).

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

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

[6] R. Gronner and L. Manheim. Hidden patterns. *Arch. Gen. Psychiat.* 17, pp. 248 (1967).

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

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

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

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

[11] K. P. Joshi, A. Joshi and Y. Yesha. On using a warehouse to analyze web logs. *Distributed and Parallel Databases* 13, pp. 161-180 (2003).

[12] H. C. Koh and S. K. Leong. Data mining applications in the context of casemix. *Annals Academy of Medicine Singapore* 30, pp. 41-49 (2001).

[13] J. Kropp. A neural network approach to the analysis of city systems. *Applied Geography* 18, pp. 83-96 (1998).

[14] A. V. Levy and A. Montalvo. The tunneling algorithm for the global minimization of functions. *SIAM J. Sci. Stat. Comp.* 6, pp. 15-29 (1985).

[15] M. May and L. Ragia, Spatial subgroup discovery applied to the analysis of vegetation data. *Lecture Notes in Artificial Intelligence* 2569, pp. 49-61 (2002).

[16] 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, pp. 13-16 (2004).

[17] 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, pp. 1703-1724 (2002).

[18] N. H. Pronko and L. Manheim. Hidden patterns - Studies in psychoanalitic literary criticism. *Psychol. Rec.* 17, pp. 283-& (1967).

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

[20] 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, pp. 659-& (1996).

[21] Strongin. ESTE E' UM PAPER DE UM RUSSO QUE EU VI NO COMPUTER JOURNAL EM 1975. O PAPER E' DOS ANOS 60, CREIO. NAO CONSEGUI ENTRAR EN LA WEBOFSCIENCE PARA ENCONTRA-LO.

[22] J. W. Weaver. Hidden patterns in Joyce 'Portrait of the artist as a young man'. *South Atlantic Bulletin* 41, pp. 63-63 (1976).