Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 8, No. 1, pp. 103–119, January 2012

THE MESH ADAPTIVE DIRECT SEARCH ALGORITHM FOR PERIODIC VARIABLES*

Charles Audet and Sébastien Le Digabel

Abstract: This work analyzes constrained blackbox optimization in which the functions defining the problem are periodic with respect to some or all the variables. We show that the natural strategy of mapping trial points into the interval defined by the period in the Mesh Adaptive Direct Search (Mads) framework can be easily done in practice, and the convergence analysis does not suffer any loss if some minor algorithmic conditions are met. The proposed strategy is tested on a nonsmooth classification problem and on a biobjective portfolio selection problem for which Mads is repeatedly used on single-objective problems.

Key words: *mesh adaptive direct search algorithms (*Mads*), pattern search algorithms, periodic functions, blackbox optimization, convergence analysis*

Mathematics Subject Classification: *90C30, 90C56, 49J52, 49N20*

1 Introduction

The need for blackbox optimization methods typically arises when the functions defining the objective and the constraints are supplied by a computer code. The code reads an input vector $x \in \mathbb{R}^n$ and after performing some computations, returns the objective and constraints function values. In some cases, the code fails to evaluate and terminates abruptly; such termination is caused by hidden constraints [14]. Direct search methods are designed for the situation where the analytical expression and the internal structure of the blackbox are not exploitable.

Pattern Search (Gps) [32], Mesh Adaptive Direct Search (Mads) [10], Nelder-Mead [29] and Direct [19] are examples of direct search methods that do not attempt to model the functions.

The present paper studies particular instances of blackbox optimization problems in which additional structure about some or all variables is known and can be exploited. Consider the optimization problem

$$\min_{x \in \Omega} \quad f(x) \tag{1.1}$$

where Ω is a subset of $\mathbb{R}^n$ and the function f is periodic with respect to the variables $x_i \in [\ell_i, u_i[$ for $i \in \{1, 2, \ldots, \overline{n}\}$, with $\overline{n} \leq n$. Formally, this signifies that

$$f\left(x + m(u_i - \ell_i)e_i\right) = f(x)$$

for every integer m and for every $i \leq \overline{n}$, where e_i denotes the standard i-th coordinate vector in $\mathbb{R}^n$. For simplicity, we use the term *periodic variables* for the variables x_i, $i \leq \overline{n}$. For example, such variables, may correspond to hours in the day, days in the week, or angles.

ISSN 1348-9151

A way to deal with periodic variables is to explicitly add the bound constraints $\ell_i \leq x_i \leq u_i$ for $i \leq \overline{n}$. In the smooth case, these constraints can be handled by the extreme barrier approach, consisting in rejecting infeasible trial points, while making sure that the local exploration directions conform to the boundary of the bound constraints [23]. Another approach is based on an augmented Lagrangian [24]. In the nonsmooth case, the extreme barrier can be used if enough directions are considered [2, 10]. Alternatively, these constraints can be handled by a filter [9], a progressive barrier [11] or a progressive-to-extreme barrier [12]. Penalty methods are also used in [25, 26]. However, all these approaches introduce artificial bounds on the periodic variables, and consequently, may generate trial points converging to an artificial local solution.

Another way to deal with periodic variables consists in treating them as being unbounded. However, this approach may not be possible in the cases where the blackbox expects values inside the interval $[\ell_i, u_i]$. Furthermore, omitting these bounds poses difficulties in the algorithmic parameters that are automatically chosen, such scales associated to individual variables.

As suggested in [3] for scaling purposes in the design of space trajectories, a periodic variable x_i can be treated by adding or subtracting an integer multiple of the period $u_i - \ell_i$ so that the resulting value lies in the interval $[\ell_i, u_i[$. This ensures that a local descent that moves x_i to one of its bound will not get stuck at that bound, but instead, would move the value of x_i towards the opposite bound. This strategy was also proposed in [17, Section 7.4.1.3], and is appropriate for some classes of direct search methods that do not rely on a mesh, but require a form of sufficient descent [20, 27].

Section 2 shows how to adapt this idea to the MADS framework. The resulting class of algorithms is called $\overline{\text{MADS}}$. We give precise conditions under which our new approach ensures the same hierarchical nonsmooth convergence results as in MADS. We also detail an example that illustrates the necessity of these conditions.

Finally, this new method is tested in Section 3 on two test problems with our NOMAD software. The user of the software simply needs to flag and specify the bounds of the periodic variables and the algorithmic conditions necessary to handle the periodicity will be automatically satisfied. First, a classification problem expressed as a new blackbox problem, and then a biobjective portfolio selection problem from finance. In both cases, the proposed strategy to handle periodic variables allows the algorithm to escape from suboptimal solutions created by the artificial bounds on the variables.

2 Mesh Adaptive Direct Search with Periodic Variables

2.1 The Mesh Adaptive Direct Search Algorithm

A key to the convergence of direct search algorithms such as GPS [8, 32] and MADS [10] is that all trial points generated by the algorithm lie on a conceptual mesh which is defined below. The mesh at iteration k is constructed using three elements: the set of previously visited points $V_k \subset \mathbb{R}^n$, a parameter $\Delta_k^m \in \mathbb{R}_+$ that dictates the coarseness of the mesh (the superscript m stands for the word *mesh*), and finally, a finite set of directions forming a matrix $D \in \mathbb{R}^{n \times n_D}$. Only this last element, D, remains the same over all iterations k.

There are two restrictions on the matrix D. First, D must be a positive spanning matrix [7, 15], i.e., nonnegative linear combinations of its columns must span $\mathbb{R}^n$. Second, it must be possible to write D as the product GZ of some nonsingular generating matrix $G \in \mathbb{R}^{n \times n}$ by an integer matrix $Z \in \mathbb{Z}^{n \times n_D}$. This last condition is used in the convergence analysis of GPS [32] and shown to be a necessary condition in [6]. The mesh can be formally

defined as follows.

Definition 2.1. At iteration k, the current mesh is defined to be the union

$$M_k \quad = \quad \bigcup_{x \in V_k} \{x + \Delta_k^m Dz : z \in \mathbb{N}^{n_D}\}$$

where V_k is the set of points where the objective function and constraints have been evaluated by the start of iteration k. $V_0 = \{x_0\}$ contains the starting point.

Each iteration of MADS is composed of a SEARCH and a POLL steps. Both steps generate a finite number of trial points lying on the mesh in an attempt to improve the current incumbent solution. The SEARCH step is optional and may generate trial points anywhere on the mesh. It is usually defined by users with knowledge of the problem, although one can use generic search strategies such as Latin-Hypercube sampling [30]. The POLL step explores mesh points in the neighborhood of the current incumbent solution. Both steps can terminate opportunistically as soon as a new incumbent is found. If such an improvement is made during the SEARCH step, the POLL is skipped at this iteration.

The mesh size parameter update rules are such that at iteration k, there exists an integer r_k, positive or negative, such that $\Delta_k^m = \Delta_0^m \tau^{r_k}$ (which implies that $r_0 = 0$), where $\tau = \frac{p}{q} > 1$ is a fixed rational number, and where $p \geq 2$ and $q \geq 1$ are two integers. The mesh size parameter is reduced ($\Delta_{k+1}^m < \Delta_k^m$) only when iteration k fails to generate a new incumbent solution. At the end of each iteration, the next mesh size parameter is set to be the product of an integer power of τ by the current one: $\Delta_{k+1}^m = \tau^{w_k} \Delta_k^m$, where w_k is a finite integer, positive or negative. Note that this implies that $\tau^{r_{k+1}} = \tau^{w_k} \tau^{r_k}$ i.e. $r_{k+1} = w_k + r_k$.

A general description of MADS is summarized in Figure 1. The interested reader is invited to consult [10] for a complete description of the algorithm. In the present document, we only highlight the elements necessary for the convergence analysis of $\overline{\text{MADS}}$, the periodic version of MADS presented in the next section.

A GENERAL MADS ALGORITHM

- INITIALIZATION: Let x_0 be a starting point, G and Z be the matrices used to define D, and τ be the rational number used to update the mesh size parameter.
  Set the iteration counter $k \leftarrow 0$.
- SEARCH AND POLL STEPS: Perform the SEARCH and POLL steps (or only part of them) until an improved mesh point x_{k+1} is found on the mesh M_k (see Definition 2.1), or until all trial points are visited.
- UPDATES: Update Δ_{k+1}^m and V_{k+1}.
  Set $k \leftarrow k + 1$ and go back to the SEARCH and POLL steps.

Figure 1: High-level description of the MADS algorithm.

2.2 Modification to the MADS Algorithm for Periodic Variables

We propose a modification to the algorithm in Figure 1 in order to treat periodic variables. This modification is more elaborate than simply bounding the periodic variables and is based on the following notation. Let Π be the $n \times n$ diagonal matrix with the periods or zeroes on

the diagonal:

$$\Pi_{i,i} = \begin{cases} u_i - \ell_i & \text{if } i \leq \overline{n}, \\ 0 & \text{otherwise.} \end{cases}$$

In order to simplify the presentation, we modify the traditional MADS notation by adding a bar over each element that we wish to carry over to the new algorithm. $\overline{\text{MADS}}$ translates a trial point $t \in M_k$ generated by the SEARCH or POLL step of MADS as follows:

$$\bar{t} \;=\; t + \Pi\rho \;\in\; \mathbb{R}^n \tag{2.1}$$

where $\rho \in \mathbb{Z}^n$ satisfies $\bar{t}_i = t_i + \rho_i(u_i - \ell_i) \in [\ell_i, u_i[$ for the periodic variables $i \leq \overline{n}$ and $\rho_i = 0$ for the non-periodic variables $i > \overline{n}$. The translated point $\bar{t}$ becomes the new trial point for $\overline{\text{MADS}}$. It is easy to see that $f(\bar{t}) = f(t)$ and that for each periodic variable $i \leq \overline{n}$, there is a unique integer ρ_i that satisfies $\bar{t}_i \in [\ell_i, u_i[$.

2.3 Necessary Algorithmic Conditions

This section describes an example showing that the strategy exposed in 2.2 is not acceptable without additional algorithmic conditions. This example shares similarities with the one in [6] showing the necessity of the assumption that D be the product of a non-singular matrix with an integer matrix.

Example 2.2. Consider the optimization problem in $\mathbb{R}$ with the continuously differentiable periodic function $f(x) = \sin(2x)$ on $\Omega = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right[$. The minimizer is at $\hat{x} = -\frac{\pi}{4}$. On the closed interval, there is a local minimizer at $x = \frac{\pi}{2}$.

Let us apply a $\overline{\text{MADS}}$ algorithm with parameters $x_0 = \pi - 3$, $\Delta_0^m = 1$, $G = 1$, $Z = D = [-1,\ 1]$, and $\tau = 2$. In addition, $w_k = -1$ if no improvement has been made at iteration k, and $w_k = 0$ otherwise. The SEARCH step of the $\overline{\text{MADS}}$ algorithm is defined as follows. At iteration k, if $\Delta_k^m = 1$ and if there exist positive integers a_k and b_k such that $x_k = b_k\pi - a_k > 0$, then the SEARCH generates the trial point

$$t_k \;=\; x_k - ((\lceil 1/x_k \rceil - 1)a_k + 1)\Delta_k^m \tag{2.2}$$

where $\lceil 1/x_k \rceil \in \mathbb{N} \setminus \{0\}$ denotes the smallest integer larger than $1/x_k$. This trial point t_k belongs to the current mesh M_k since $((\lceil 1/x_k \rceil - 1)a_k + 1))$ is an integer. The SEARCH step is opportunistic in the sense that the iteration terminates as soon as a new incumbent solution is generated. The period of the variable is π and setting $\rho = (\lceil 1/x_k \rceil - 1)b_k$ moves t_k to

$$\overline{t_k} \;=\; t_k + (\lceil 1/x_k \rceil - 1)b_k\pi \;=\; \lceil 1/x_k \rceil \times x_k - 1.$$

The trial point $\overline{t_k}$ belongs to the open interval $]0, x_k[$ since

$$0 \;=\; (1/x_k) \times x_k - 1 \;<\; \overline{t_k} \;=\; \lceil 1/x_k \rceil\, x_k - 1 \;<\; (1/x_k + 1)\, x_k - 1 \;=\; x_k\ .$$

Accordingly to (2.1), $\overline{t_k}$ is the $\overline{\text{MADS}}$ trial point (instead of t_k). This leads to $f(t_k) = f(\overline{t_k}) = \sin(2\overline{t_k}) < \sin(2x_k) = f(x_k)$ and thus $x_{k+1} = \overline{t_k}$ is the next incumbent solution.

By induction on k, the entire sequence of iterates $\{x_k\}$ is monotone decreasing and converges to the origin, which does not satisfy any sort of optimality condition.

In order to avoid the behavior detailed in this example while applying the strategy described in 2.2 and to keep the convergence properties of MADS, we add the following algorithmic conditions and analyze them in the next section.

Condition 2.3. The mesh size parameter $\Delta_k^m = \tau^{r_k}\Delta_0^m$ is bounded above at every iteration, i.e., there exists an integer $r_{\max} \geq 0$ such that $\Delta_k^m \leq \tau^{r_{\max}}\Delta_0^m$ for every k.

Condition 2.4. The matrix $\frac{1}{\Delta_0^m}G^{-1}\Pi$ is integer.

Condition 2.3 already holds for the two existing MADS implementations, LTMADS [10] and ORTHOMADS [2]. In both cases Δ_k^m is bounded above by Δ_0^m for all k. However, these implementations consider another parameter called the *poll size parameter* Δ_k^p which can grow to larger values allowing trial points to be further away after successful iterations. The convergence analysis of MADS relies on the assumption that all trial points belong to a bounded subset of $\mathbb{R}^n$ and as a consequence, the poll size parameter will be bounded above.

A simple way to satisfy Condition 2.4 is to choose an integer j and let G be a diagonal matrix such that

$$G_{i,i} = \begin{cases} \frac{u_i - \ell_i}{j\Delta_0^m} & \text{if } i \leq \overline{n}, \\ \frac{1}{j\Delta_0^m} & \text{otherwise.} \end{cases} \tag{2.3}$$

This condition is violated in Example 2.2 as $\frac{1}{\Delta_0^m}G^{-1}\Pi = \pi$ is not an integer.

2.4 Convergence Analysis

The problem with the previous example is that although the points t_k lie on the same mesh M_k, the trial points $\overline{t_k}$ do not lie on a mesh. These trial points converge to a point that does not satisfy any optimality conditions.

In the present section, we show that under Conditions 2.3 and 2.4, all trial points $\overline{t_k}$ lie on another mesh called $\overline{M_k}$, constructed using three elements: the set of previously visited points V_k, a parameter $\overline{\Delta_k^m}$ that dictates the coarseness of the mesh, and a finite set of directions forming a matrix $\overline{D}$.

The set of visited points V_k is the same as the one used in the definition of M_k. The other two elements differ. The $n \times 2n$ direction matrix is a maximal positive basis $\overline{D} = G[I_n \ -I_n]$ where I_n is the identity matrix and G is the same nonsingular matrix used to define D. The parameter $\overline{\Delta_k^m}$ is defined as follows:

$$\begin{cases} \overline{\Delta_0^m} = \dfrac{\Delta_0^m}{p^{r_{\max}}q^{r_{\max}}} & \text{for } k = 0, \text{ and} \\ \\ \overline{\Delta_{k+1}^m} = p^{w_k}\overline{\Delta_k^m} & \text{for } k \geq 1 \end{cases}$$

where r_{max} is the nonnegative integer from Condition 2.3, $w_k \in \mathbb{Z}$ is the integer used for the mesh size update ($\Delta_{k+1}^m = \tau^{w_k}\Delta_k^m$) and $p \geq 2$ and $q \geq 1$ are the integers such that $\tau = \frac{p}{q}$. This leads to the following lemma:

Lemma 2.5. *Under Condition 2.3, the mesh size parameter at iteration k satisfies*

$$\overline{\Delta_k^m} = \frac{\Delta_k^m}{p^{r_{\max}}q^{r_{\max}-r_k}} = \frac{\Delta_0^m}{p^{r_{\max}-r_k}q^{r_{\max}}} \ .$$

Proof. By definition, the result is true for $k = 0$. By induction, suppose that the result is true for some value of k. Then, for $k+1$, $r_{k+1} = r_k + w_k$, and therefore, using $\tau = \frac{p}{q}$,

$$\begin{aligned}\overline{\Delta_{k+1}^m} \;=\; p^{w_k}\overline{\Delta_k^m} \;&=\; \frac{\Delta_k^m}{p^{r_{\max}-w_k}q^{r_{\max}-r_k}} \\ &=\; \frac{\Delta_{k+1}^m\tau^{-w_k}}{p^{r_{\max}-w_k}q^{r_{\max}-r_k}} \\ &=\; \frac{\Delta_{k+1}^m}{p^{r_{\max}}q^{r_{\max}-r_k-w_k}} \;=\; \frac{\Delta_{k+1}^m}{p^{r_{\max}}q^{r_{\max}-r_{k+1}}}\;.\end{aligned}$$

Substituting $\Delta_{k+1}^m = \left(\frac{p}{q}\right)^{r_{k+1}} \Delta_0^m$ completes the proof. □

This notation allows us to define the following mesh through $\overline{D}$:

$$\overline{M_k} \;=\; \bigcup_{x\in V_k} \left\{x + \overline{\Delta_k^m}\;\overline{D}\overline{z} : \overline{z} \in \mathbb{N}^{n_D}\right\}.$$

Figure 2 illustrates meshes in $\mathbb{R}^2$ for periodic variables on the interval $[0, 2\pi]$. The mesh size parameter of the two first plots satisfies Condition 2.4 since the ratio of the period over the mesh size parameter is rational : $\frac{4}{3}$. The ratio for the last two plots is equal to π, and therefore Condition 2.4 is not satisfied. This is illustrated in the last plot where the mesh $\overline{M_k}$ is dense on the domain $[0, 2\pi] \times [0, 2\pi]$.

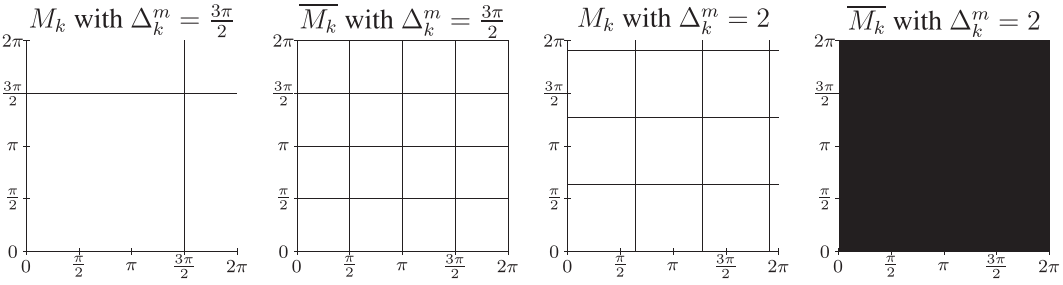

Figure 2: Examples of meshes in $\mathbb{R}^2$. In the first three figures, the mesh consists of the intersection of all lines, and in the last one the mesh is dense in the entire domain.

Theorem 2.6. *Under Conditions 2.3 and 2.4, every trial point generated at iteration k of* $\overline{\text{MADS}}$ *belongs to the mesh $\overline{M_k}$.*

Proof. Consider the point $t_k = x_k + \Delta_k^m Dz$ with $z \in \mathbb{N}^{n_D}$, lying on the mesh M_k, and translated to $\overline{t_k} = t_k + \Pi\rho_k$ where $\rho_k \in \mathbb{Z}^n$ and $(\overline{t_k})_i \in [\ell_i, u_i[$ for every $i \le \overline{n}$. Then

$$\begin{aligned}\overline{t_k} \;&=\; x_k + \Delta_k^m Dz + \Pi\rho_k \\ &=\; x_k + \overline{\Delta_k^m}\left(p^{r_{\max}}q^{r_{\max}-r_k}Dz + \Pi\rho_k/\overline{\Delta_k^m}\right) \\ &=\; x_k + \overline{\Delta_k^m}\left(p^{r_{\max}}q^{r_{\max}-r_k}Dz + \frac{p^{r_{\max}-r_k}q^{r_{\max}}}{\Delta_0^m}\Pi\rho_k\right) \\ &=\; x_k + \overline{\Delta_k^m}\left(p^{r_{\max}}q^{r_{\max}-r_k}GZz + GG^{-1}\frac{p^{r_{\max}-r_k}q^{r_{\max}}}{\Delta_0^m}\Pi\rho_k\right) \\ &=\; x_k + \overline{\Delta_k^m}G\left(p^{r_{\max}}q^{r_{\max}-r_k}Zz + \frac{p^{r_{\max}-r_k}q^{r_{\max}}}{\Delta_0^m}G^{-1}\Pi\rho_k\right)\end{aligned}$$

belongs to $\overline{M_k}$ since Condition 2.4 ensures that $\left(p^{r_{\max}} q^{r_{\max}-r_k} Zz + \frac{p^{r_{\max}-r_k} q^{r_{\max}}}{\Delta_0^m} G^{-1}\Pi\rho_k\right)$ is an integer vector. □

Corollary 2.7. *Under Conditions 2.3 and 2.4, the entire convergence analysis of* MADS *holds for* $\overline{\text{MADS}}$.

Proof. The convergence analysis of MADS relies on the fact that all trial points t_k lie on a mesh M_k. The translated trial points $\overline{t_k}$ do not necessarily lie on the mesh M_k, but, Theorem 2.6 ensures that they all lie on the conceptual mesh $\overline{M_k}$. Furthermore, Lemma 2.5 ensures that $\Delta_k^m \to 0$ if and only if $\overline{\Delta_k^m} \to 0$. It follows that the same convergence analysis presented in MADS can be used, with $\overline{M_k}$ playing the role of M_k. □

The MADS convergence analysis relies on the assumption that all trial points lie in a bounded set. This assumption is satisfied in particular when the level sets of the objective function are bounded. In the present context, it is trivial that all translated trial points $\bar{t}_k$ are such that the i-th periodic variable lies in the bounded interval $[\ell_i, u_i[$. The a posteriori assumption that the trial points belong to a bounded set is replaced by Condition 2.4, which can be verified a priori, in particular by setting G as in Equation (2.3).

3 Numerical Results

The aim of this section is to test numerically the proposed strategy to handle periodic variables on two test problems. We first compare the strategy on MADS and GPS [32] with coordinate directions for a classification problem, then on BIMADS [13], the biobjective version of MADS, for a portfolio selection problem [33]. For reproducibility, executions of MADS are performed using the ORTHOMADS [2] instantiation of MADS. We did not consider LTMADS [10] since it uses randomness to generate new trial points, which complicates the comparisons.

Numerical tests are conducted with the version 3.4 of NOMAD [1, 22], which includes periodic variables and biobjective optimization. Functions for both problems are periodic with period 2π since all variables represent angles in spherical coordinates. A slight difference between the periodic and aperiodic settings is that the aperiodic version considers bounds of $[0,\ \pi]$ for the first variable (first angle of the spherical coordinates) while the periodic version keeps bounds of $[0,\ 2\pi]$ to respect the 2π period for all variables.

The numerical tests use the default parameter values of NOMAD. More precisely, the initial mesh size parameter Δ_0^m is set to 1, and the other mesh parameters are $\tau = 4$ with ORTHOMADS and $\tau = 2$ with GPS, and $w_k = 1$ whenever iteration k is successful, and $w_k = -1$ otherwise. The matrix G is automatically set to $G = \frac{1}{10}\Pi$ as in Equation (2.3) with $j = 10$ and $\overline{n} = n$, and the mesh directions $D = GZ$ are defined using $Z = [I_n\ -I_n]$. It follows that Condition 2.3 is satisfied since $\frac{1}{\Delta_0^m} G^{-1}\Pi = 10 I_n$. The source code of NOMAD and of both test problems as well as the test logs are available for download from [1].

3.1 A Hyperplane Separation Problem

This section first describes the original formulation of the problem and gives a new blackbox formulation on which NOMAD is applied.

3.1.1 Problem Presentation

Let Q be a finite set of points in $\mathbb{R}^n$ partitioned into $Q = X \cup Y$ with $X \cap Y = \emptyset$. The objective is to identify a hyperplane of equation $a^T x = b$, with $a \in \mathbb{R}^n$ a normalized vector and $b \in \mathbb{R}$, that separates the sets X and Y as best as possible. In the present paper, we achieve this by minimizing the number of points that are misclassified, i.e. located on the "wrong" side of the hyperplane. The corresponding mathematical model may be formally written as

$$\begin{array}{ll} \min\limits_{a\in\mathbb{R}^n, b\in\mathbb{R}} & g(a,b) \\ \text{subject to} & \|a\| = 1 \end{array} \quad \text{, where} \quad g(a,b) = \big|\{x \in X : a^T x < b\}\big| + \big|\{x \in Y : a^T x > b\}\big| \,. \tag{3.1}$$

Observe that points located exactly on the hyperplane are not misclassified. A review on the more general subject of classification can be found in [4]. Problem (3.1) is useful in the field of automatic classification: After a hyperplane is decided for a sample of data whose properties are known, it can be applied to some unknown new data in order to classify them. For example, it may be applied to differentiate healthy cells from ill cells for the detection of tumors (problem Cancer in Uci Repository of Machine Learning Databases [5]).

Marcotte *et al.* [28] propose a mixed integer program (Mip) for problem (3.1) where the binary variables express if the data points are misclassified or not. This formulation is difficult to solve when the number of points grows large. The Mip formulation of the Diabetes problem (whose complete name is Pima Indians Diabetes in the Uci databases) is not solvable by Cplex 10.11 [18] on our hardware: it was stopped after 12 hours, due to memory limitation (branching tree of size 3.2 GB) with a current best solution of 130 misclassified points. Hence, this problem can be classified as difficult and has been chosen for our tests in Section 3.1.2. In order to be treated with direct search solvers such as Nomad, the problem is transformed into a blackbox. The technical presentation of this transformation is detailed in the appendix.

3.1.2 Results

Nomad is applied to the Diabetes instance from the Uci databases. All seven variables represent angles. The default Nomad parameters are used and it terminates when the mesh size parameter reaches a precision of 10^{-13}, the default NOMAD precision.

As the choice of the starting point is important, several executions are performed from 100 starting points generated from a Latin-Hypercube sample [31]. The periodic and aperiodic versions are compared with the OrthoMads and Gps directions for a total of 400 runs. Table 1 summarizes these 400 executions. The best classification was obtained with both periodic algorithms for a total of 148 misclassified points.

Figure 3 shows performance profiles allowing to analyze graphically these results. Performance profiles have been introduced in [16] and are adapted to the classification problem as follows: The four plots correspond to the different tested strategies and summarize 100 executions from a different starting point. For a given Alpha value (horizontal axis), a point belonging to one of these plots indicates the number of instances on which the corresponding method gave a value within Alpha% of the best obtained solution equal to 148. For example the coordinates $(10, 20)$ mean that 20 instances out of 100 gave a solution within 10% of 148, i.e. less than or equal to 162.

The two periodic plots begin at the coordinates $(0, 1)$ meaning that both methods terminated once with the value 148. The two aperiodic plots begin at $(0, 0)$ meaning that all solutions obtained with the aperiodic methods are worse than 148. From both the Table 1

and Figure 3, it is clear that the periodic versions $\overline{\text{MADS}}$ and $\overline{\text{GPS}}$ dominate the aperiodic algorithms MADS and GPS.

Table 1: Results for the 400 runs from 100 different starting points for four different methods.

Method	Best	Worst	Average	Median	Std Dev
Periodic OrthoMADS	148	181	163	163	7.86
Aperiodic OrthoMADS	150	227	174	175	15.60
Periodic GPS	148	182	163	162	8.29
Aperiodic GPS	150	237	178	177	21.00

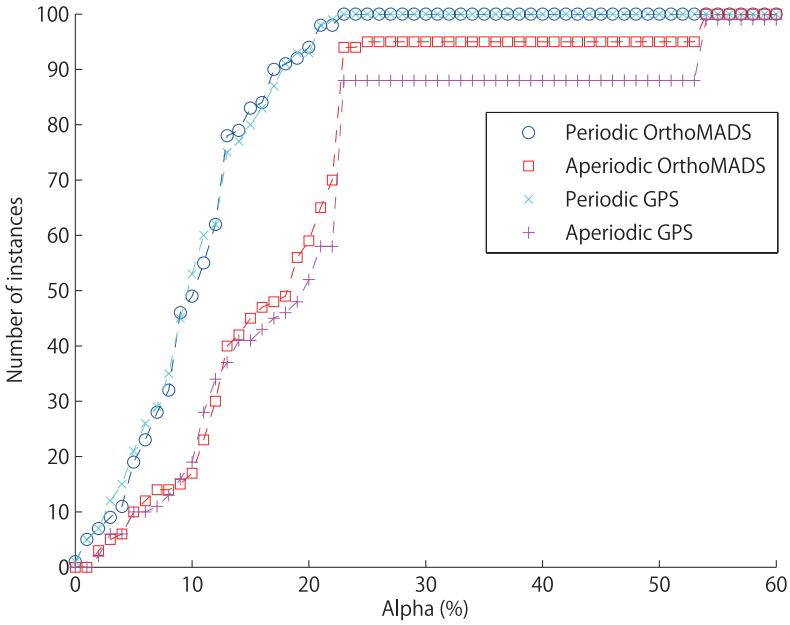

Figure 3: Performance profiles for the 400 runs from 100 different starting points. Plots close to the upper left portion of the graph are the best.

The effect of the treatment of periodic variables presented in this paper is visualized in Figure 4 showing the evolution of the seventh variable for both MADS and $\overline{\text{MADS}}$ during the first 1000 evaluations of a particular instance. This instance is the one with the starting point that gave the best result with $\overline{\text{MADS}}$. In both the aperiodic and periodic cases, the algorithm drives the variable to small values, but in the periodic case, θ_7 crosses over to take values near 2π. The algorithm converges then to a solution with a large seventh angle.

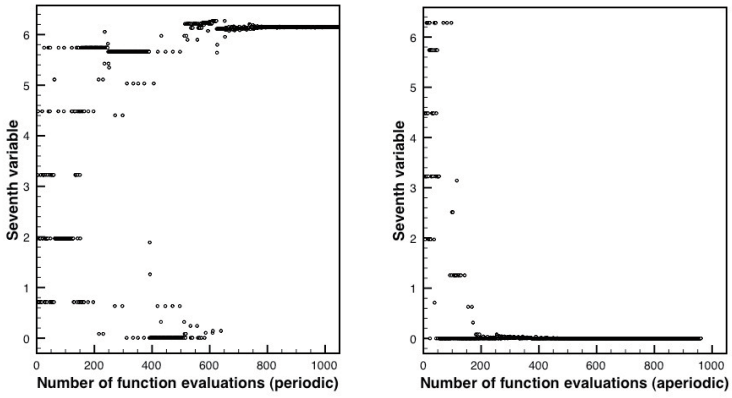

Figure 4: Values of the seventh angle with periodic variables (left) and without periodic variables (right).

3.2 A Biobjective Problem from Finance

This portfolio selection problem with five assets and two objective functions is taken from [21]. The objective is to maximize the mean return and the skewness of the portfolio for a fixed variance of one. This last constraint can be implicitly expressed with a change of variables to spherical coordinates. The problem then becomes non-linear and BiMads [13], the biobjective version of Mads, was used in [33] to solve it. BiMads consists in sequentially launching several instances of Mads with different single-objective formulations, in order to approximate the Pareto front. The Pareto front is the set of undominated points, i.e. points such that no other point in the front has better values for both the objective functions. We denote by $\overline{\text{BiMads}}$ the algorithm identical to BiMads except that it treats the periodic variables using $\overline{\text{Mads}}$ instead of Mads.

As in the classification problem of 3.1, the change of variables reduces the five assets to four variables in $[0, 2\pi[$ except for the the first angle of the aperiodic version which lies in $[0, \pi[$. The two objective functions are periodic with respect to these angles. There are no other constraints.

Nomad is used with three different budgets of 100, 500, and 1000 evaluations from the starting point $x_0 = (\pi/2, \pi, \pi, \pi)^T$ in the center of the domain. Note that the choice of the starting point is not as important as in the single-objective case since BiMads changes this parameter between single-objective executions. The Latin Hypercube sampling that Nomad uses by default for biobjective optimization is disabled in order to remove randomness. All other Nomad parameters are set to the default values and in particular OrthoMads directions are considered.

Table 2 shows the numbers of single-objective executions and of undominated points. Figure 5 shows the two Pareto front approximations generated by BiMads with or without periodic variables and for the three budgets of evaluations. As we maximize the two objective functions, a good approximation of the Pareto front has to be as far as possible from the axes. The figure clearly shows the dominance of the front generated while using periodic variables for 100 and 500 evaluations, even if less undominated points are produced. With a budget of 500 evaluations, there are even points of the periodic front that dominate the entire aperiodic front. With 1000 evaluations, the two methods generate two similar

approximations, but the periodic version gives 16 more points. The true Pareto front is of course identical for both periodic and aperiodic formulations. But the aperiodic formulation introduces artificial bounds on the variables, thereby creating local solutions, or makes the true Pareto front inaccessible to the MADS algorithm.

Table 2: Numbers of single-objective executions and of undominated points for the six different biobjective optimizations.

Method	Evaluations	Single-obj. executions	Undominated points
Periodic	100	5	6
Aperiodic	100	5	23
Periodic	500	12	61
Aperiodic	500	12	100
Periodic	1000	16	127
Aperiodic	1000	16	143

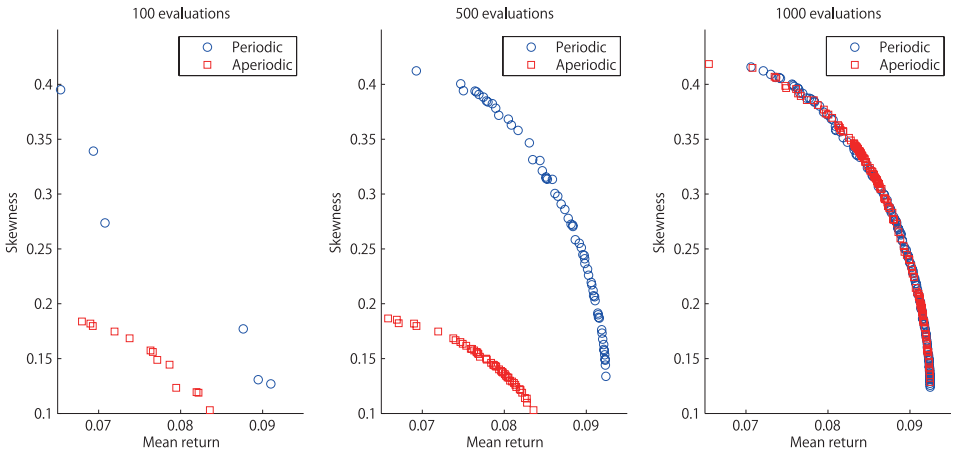

Figure 5: Pareto front approximation with 100, 500, and 1000 function evaluations, with or without periodic variables. Mean return and skewness are the two objective functions to maximize. Best points are far from the origin.

Figure 6 gives insight of why $\overline{\text{BIMADS}}$ outperforms BIMADS for the 500 evaluations budget. The figure plots the values of the third angle generated by the algorithm versus the number of function evaluations. With BIMADS, this variable barely moves from 0, while $\overline{\text{BIMADS}}$ allowed the variable to vary significantly and access more interesting values.

4 Discussion

This paper examined the natural strategy of handling periodic variables by mapping trial points into the interval defined by the period. This strategy requires algorithmic conditions

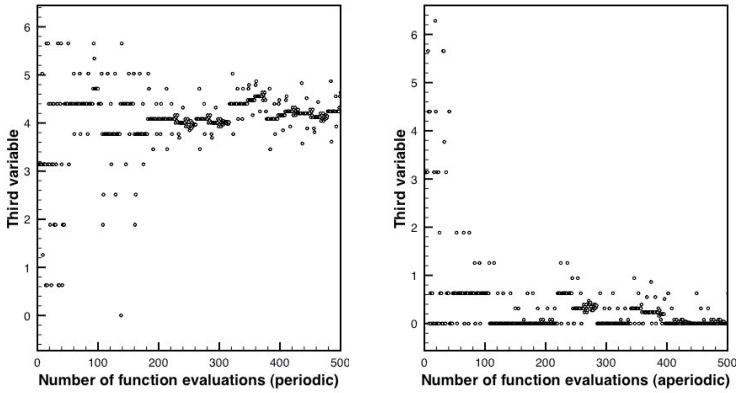

Figure 6: Values of the third angle for 500 evaluations, with periodic variables (left), and without periodic variables (right).

involving the mesh size parameter which must be bounded above at every iteration (Condition 2.3), the initial mesh size parameter Δ_0^m, the set of directions used to define the mesh $D = GZ$, and the period of the variables Π (Condition 2.4). In practice, these conditions are satisfied by our implementation, and in theory, the new method inherits the entire MADS convergence analysis.

The strategy to handle periodic variables applies to any instantiation of the MADS class of algorithm. We have tested it on the GPS and ORTHOMADS instantiations of MADS on a new blackbox formulation of a classification problem. We have also tested it on a more general algorithm called BIMADS that uses MADS to solve subproblems. In all the numerical experiments conducted here, the new strategy outperforms the approach of treating the periodic variables as bound constrained variables. Removing the artificial bound constraints on periodic variables allows the direct search algorithm to find better local solutions.

Acknowledgments

The authors would like to thank Andy Conn and two anonymous referees for their helpful remarks.

References

[1] M.A. Abramson and C. Audet and G. Couture and J.E. Dennis, Jr. and S. Le Digabel, The NOMAD project, Software available at http://www.gerad.ca/nomad.

[2] M.A. Abramson, C. Audet, J.E. Dennis, Jr. and S. Le Digabel, OrthoMADS: A deterministic MADS Instance with orthogonal directions, *SIAM Journal on Optimization* 20 (2009) 948–966.

[3] B. Addis, A. Cassioli, M. Locatelli and F. Schoen, A global optimization method for the design of space trajectories, *Computational Optimization and Applications* 48 (2011) 635–652.

[4] J. Adem and W. Gochet, Mathematical programming based heuristics for improving LP-generated classifiers for the multiclass supervised classification problem, *European Journal of Operational Research*, 168 (2006) 181–199.

[5] A. Asuncion and D.J. Newman, UCI Machine learning repository, http:// archive.ics.uci.edu/ml/, University of California, Irvine, School of Information and Computer Sciences, 2007.

[6] C. Audet, Convergence results for pattern search algorithms are tight, *Optimization and Engineering* 5 (2004) 101-122.

[7] C. Audet, A short proof on the cardinality of maximal positive bases, *Optimization Letters.* 5 (2011) 191–194.

[8] C. Audet and J.E. Dennis, Jr., Analysis of generalized pattern searches, *SIAM Journal on Optimization*, 13 (2003) 889–903.

[9] C. Audet and J.E. Dennis, Jr., A pattern search filter method for nonlinear programming without derivatives, *SIAM Journal on Optimization* 14 (2004) 980–1010.

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

[11] C. Audet and J.E. Dennis, Jr., A progressive barrier for derivative-free nonlinear programming, *SIAM Journal on Optimization* 20 (2009) 445–472.

[12] C. Audet and J.E. Dennis, Jr. and S. Le Digabel, Globalization strategies for mesh adaptive direct search, *Computational Optimization and Applications* 46 (2010) 193–215.

[13] C. Audet, G. Savard and W. Zghal, Multiobjective optimization through a series of single-objective formulations, *SIAM Journal on Optimization* 19 (2008) 188–210.

[14] T.D. Choi and C.T. Kelley, Superlinear convergence and implicit filtering, *SIAM Journal on Optimization* 10 (2000) 1149–1162.

[15] C. Davis, Theory of positive linear dependence, *AMJ* 76 (1954) 733–746.

[16] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, 91 (2002) 201–213.

[17] P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[18] ILOG Inc., CPLEX 10.11, 2006.

[19] D.R. Jones, C.D. Perttunen and B.E. Stuckman, Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Application* 79 (1993) 157–181.

[20] T.G. Kolda and R.M. Lewis and V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, *SIAM Review* 45 (2003) 385–482.

[21] T.Y. Lai Portfolio selection with skewness: a multiple-objective approach, *Review of Quantitative Finance and Accounting* 1 (1991) 293–305.

[22] S. Le Digabel, Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm, *ACM Transactions on Mathematical Software* 37 (2011) 44:1–44:15.

[23] R.M. Lewis and V. Torczon, Pattern search algorithms for bound constrained minimization, *SIAM Journal on Optimization* 9 (1999) 1082–1099.

[24] R.M. Lewis and V. Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Optimization* 12 (2002) 1075–1089.

[25] G. Liuzzi and S. Lucidi, A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function, *SIAM Journal on Optimization* 20 (2009) 1–29.

[26] G. Liuzzi, S. Lucidi and M. Sciandrone, Sequential penalty derivative-free methods for nonlinear constrained optimization, *SIAM Journal on Optimization* 20 (2010) 2614–2635.

[27] S. Lucidi and M. Sciandrone, A derivative-free algorithm for bound constrained optimization, *Computational Optimization and Applications* 21 (2002) 119–142.

[28] P. Marcotte, G. Marquis and G. Savard, A new implicit enumeration scheme for the discriminant analysis problem, *Computers and Operations Research* 22 (1995) 625–639.

[29] J.A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.* 7 (1965) 308–313.

[30] M. Stein, Large sample properties of simulations using Latin hypercube sampling, *Technometrics* 29 (1987) 143–151.

[31] B. Tang, Orthogonal array-based latin hypercubes, *Journal of the American Statistical Association* 88 (1993) 1392–1397.

[32] V. Torczon, On the Convergence of Pattern Search Algorithms, *SIAM Journal on Optimization* 7 (1997) 1–25.

[33] W. Zghal and C. Audet and G. Savard, A new multi-objective approach for the portfolio selection problem with skewness, in *Advances in Quantitative Analysis of Finance and Accounting*, C.F. Lee (ed.), Airiti Press, 2011.

Appendix: Blackbox Formulation of the Hyperplane Separation Problem

This appendix presents a blackbox formulation of the hyperplane separation problem studied in Section 3.1 and usable by any direct search solver. The main component of the blackbox takes the vector $a \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$ as input and counts the number of misclassified points $g(a, b)$. With little computational effort, the blackbox may be modified so that it computes the minimum of $g(a, b)$ and $g(-a, -b)$. This means that for given values of a and b, it identifies the best strategy for choosing the sides associated with X and Y. We introduce $g^+(a, b) = \min\{g(a, b), g(-a, -b)\}$. The first step in the blackbox elaboration consists in reformulating Problem (3.1) as

$$\min_{\substack{a\in\mathbb{R}^n,\\ \|a\|=1}} \left(\min_{b\in\mathbb{R}} \; g^+(a, b) \right). \tag{4.1}$$

The second step uses spherical coordinates to express the normalized vector $a \in \mathbb{R}^n$. Define $a : [0, \pi[\times [0, 2\pi[^{n-2} \to \mathbb{R}^n$ with $\theta \mapsto a(\theta)$ to be the transformation into Euclidian coordinates from the spherical coordinates θ in dimension n. Let $m(\theta) = \min_{b \in \mathbb{R}} g^+(a(\theta), b)$ denote the least number of misclassified points that can be achieved for a given value of θ. Problem (4.1) can then be reformulated as

$$\begin{array}{ll} \min\limits_{\theta \in \mathbb{R}^{n-1}} & m(\theta) \\ \text{subject to} & \theta_1 \in [0\ \pi[\\ & \theta_i \in [0\ 2\pi[\ , \quad i = 2, 3, \ldots, n-1 \end{array} . \tag{4.2}$$

Note that spherical coordinates allow to modify the upper bound of the first variable from π to 2π and in that case all variables are periodic.

For a given value of $\theta \in \mathbb{R}^{n-1}$, evaluation of $m(\theta)$ requires finding the optimal value of b, which is computed as follows: Let I be the set of indices of the points to be classified: $Q = \{x^{(i)} \in \mathbb{R}^n : i \in I\}$. For every $i \in I$, let $m^{(i)} = g^+(a(\theta), b^{(i)})$ denote the number of misclassified points with respect to the hyperplane passing through $x^{(i)}$: $a(\theta)^T x = b^{(i)}$, where $b^{(i)} = a(\theta)^T x^{(i)}$. The blackbox then sets $m(\theta) = \min\{m^{(i)} : i \in I\}$, a nonnegative integer value bounded above by $|I|/2$. The complexity of the evaluation is of the order of the number of points $|I|$.

Figure 7 shows a plot of the function m on the DIABETES instance with respect to the seventh angle while fixing the six others. It is a step function, and may cause direct search algorithms to get stuck on a step. In order to allow direct search methods to discriminate between points on a step, we add a perturbation $\delta(\theta) \in [0, 1[$ to the value $m(\theta)$.

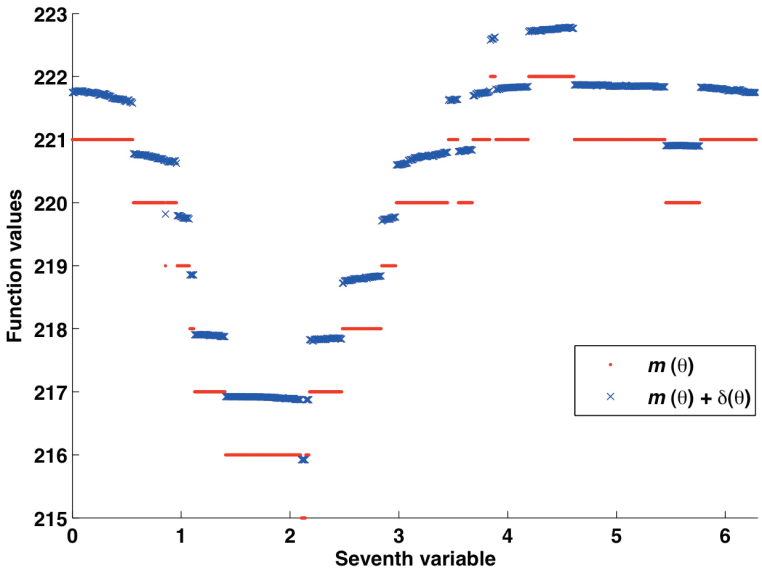

Figure 7: Representation of the blackbox functions $m(\theta)$ and $m(\theta) + \delta(\theta)$ for the data set DIABETES of the classification problem. All variables are fixed to 2.5, except the seventh which varies between 0 and 2π.

The perturbation $\delta(\theta)$ is constructed as follows. During each blackbox execution, we consider the $r = \min\{n^2, |I|\}$ points with the smallest values of $m^{(i)}$. The indices of these

points are sorted in the set $\mathcal{J}(\theta) = \{i_1, i_2, \ldots, i_r\}$ where $m^{(i_k)} \leq m^{(i_\ell)}$ if $k \leq \ell$. It follows that the first element of $\mathcal{J}(\theta)$ satisfies $m^{(i_1)} = m(\theta)$. Next, the set $\mathcal{J}(\theta)$ is partitioned with respect to the different values of $m^{(i)}$: $\mathcal{J}(\theta) = \bigcup_{z=m^{(i_1)}}^{m^{(i_r)}} \mathcal{J}_z(\theta)$ with $\mathcal{J}_z(\theta) = \{i \in \mathcal{J}(\theta) : m^{(i)} = z\}$. The perturbation is then fixed to

$$\delta(\theta) = 1 - \frac{1}{r} \sum_{z=m^{(i_1)}}^{m^{(i_r)}} \frac{|\mathcal{J}_z(\theta)|}{2^{z-m^{(i_1)}}} \ . \tag{4.3}$$

Since $1 \leq |\mathcal{J}_{f(\theta)}(\theta)| \leq r$ and $m^{(i_r)} \geq z \geq m^{(i_1)}$, the perturbation satisfies $0 \leq \delta(\theta) < 1$ and therefore $m(\theta) = \lfloor m(\theta) + \delta(\theta) \rfloor$. It follows that minimizing $f(\theta) = m(\theta) + \delta(\theta)$ is equivalent to minimizing $m(\theta)$.

The perturbed function f is plotted in Figure 7, on the same graph as m, for the DIABETES instance. The purpose of this heuristic perturbation is to assign different values to trial points that share the same value of $m(\theta)$. One can observe from the figure that a descent method applied to θ_7 would often lead to the minimizer.

An illustrative example of the computation of the functions m, δ and f is shown in Figure 8, for a set of 7 points in dimension 2, with θ set to $\frac{3\pi}{4}$. The numbers of misclassified points for the dashed hyperplanes are $(m_1, m_2, \ldots, m_7) = (2, 2, 3, 2, 1, 1, 2)$, and $f(\frac{3\pi}{4}) = 1$. A total of $r = 4$ points are considered in the determination of the perturbation and their indices $\mathcal{J}(\frac{3\pi}{4}) = \{5, 6, 1, 2\}$ are partitioned as $\mathcal{J}_1(\frac{3\pi}{4}) = \{5, 6\}$ and $\mathcal{J}_2(\frac{3\pi}{4}) = \{1, 2\}$. The perturbation is then

$$\delta\left(\frac{3\pi}{4}\right) \ = \ 1 - \frac{1}{4}\sum_{z=1}^{2} \frac{|\mathcal{J}_z(\theta)|}{2^{z-1}} \ = \ 1 - \frac{1}{4}\left(\frac{2}{1} + \frac{2}{2}\right) \ = \ \frac{1}{4}.$$

The blackbox returns the value $f(\frac{3\pi}{4}) = m(\frac{3\pi}{4}) + \delta(\frac{3\pi}{4}) = 1.25$ indicating that there is only one misclassified point.

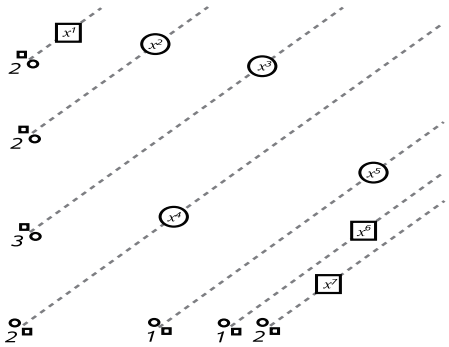

Figure 8: Computation of the blackbox functions m, θ and f, with $n = 2$, $q = 7$, and $\theta = \frac{3\pi}{4}$. The sets X and Y are represented by circles and squares, respectively.

Manuscript received 4 March 2010
revised 14 July 2010
accepted for publication 27 September 2010

Charles Audet
GERAD and Département de mathématiques et de génie industriel, Ecole Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville, Montréal, Québec H3C 3A7 Canada
E-mail address: Charles.Audet@gerad.ca

Sébastien Le Digabel
GERAD and Département de mathématiques et de génie industriel, Ecole Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville, Montréal, Québec H3C 3A7 Canada
E-mail address: Sebastien.Le.Digabel@gerad.ca