

A new class of test functions for global optimization

Bernardetta Addis (b.addis@ingfi1.ing.unifi.it)

Dip. Sistemi e Informatica - Università di Firenze (Italy)

Marco Locatelli (locatell@di.unito.it)

Dip. Informatica - Università di Torino (Italy)

Abstract. In this paper we propose a new class of test functions for unconstrained global optimization problems for which, however, it is a priori known that the global minimum lies in the interior of a sphere centered at the origin. The class depends on some parameters through which the difficulty of the test problems can be controlled. As a basis for future comparison, we propose a selected set of these functions, with increasing difficulty, and some computational experiments with two simple global optimization algorithms.

Keywords: global optimization, test problems

1. Introduction

In the field of optimization the definition of test problems is an important and nontrivial task. Test problems should reflect the wide variety of difficulties encountered when solving practical problems and are essential in validating algorithms. In the field of Global Optimization (GO in what follows) there exists an old class of test functions, the Dixon-Szegö test set (see (Dixon-Szegö, 1978)). Unfortunately, these problems are of limited dimension and of mild difficulty (they usually have only few local minima). Therefore, testing *only* on them is not an appropriate way to validate GO algorithms. The need for new and widely recognized GO test problems emerged in a number of recent publications. Here we recall a book (Floudas-et-al, 1999), some papers (Gaviano-et-al, 2003; Lavor-Maculan, 2004; Neumaier-et-al, 2005; Pinter, 2002), and the global optimization web site (GO-site, 2005). In this paper we propose a new class of test functions depending on a limited number of user-specified parameters, through which it is possible to control the difficulty of the corresponding GO problem. Since GO includes many different classes of problems (constrained/unconstrained, differentiable/nondifferentiable, and so on), it is basically impossible to define a class of test functions depending on few parameters which covers all the possible cases. For this reason in this paper we concentrate our attention on a restricted, though relevant, subset of GO problems, namely unconstrained problems for which it is known that the global minimum is attained in the interior of a sphere centered at the origin,



© 2005 Kluwer Academic Publishers. Printed in the Netherlands.

and for which efficient local search procedures exist. In the following section we recall some features on which the difficulty of these GO problems depend. In Section 3 we introduce a set of parameters whose values have to be specified by the user and a set of parameters whose values are randomly assigned. In Section 4 we introduce some basic one-dimensional and multidimensional component functions, depending on some of the above parameters, and we generate GO test functions with the required properties by iterative procedures starting from the basic components. For any generated test function the objective function value of the global minimum will be known, but its position won't. In Section 5 we present the results of two simple GO algorithms on some selected test functions, which can be employed as a basis for future comparison.

2. Features making a GO problem difficult

In some recent papers (Baritompaa-et-al, 2005; Locatelli, 2005; Locatelli-Wood, 2005) features on which the difficulty of box-constrained GO problems (with the global minimum in the interior of the box) depends, are discussed. In particular, in (Locatelli, 2005) it is observed that the difficulty of such GO problems is not merely connected to the number of local (and not global) minima. An important factor is how "chaotic" is the position of these local minima within the search box. This observation was stimulated by the analysis of some practical box-constrained GO problems, namely molecular conformation problems such as the Lennard-Jones and Morse potential¹. Indeed, the observation of the *funneling* landscape of these functions (see e.g. (Wales-Doye, 1997)) lead to the definition of local minima at different levels in (Locatelli, 2005). Basically, a local minimum at level 1 is the standard local minimum. A local minimum at level 2 is a local minimum over a directed graph whose nodes are the local minima at level 1 and arcs are defined by a suitable neighborhood structure. In particular, in this paper a directed arc connects node X to node Y if the region of attraction of node/local minimum Y has a nonempty intersection with a small neighborhood of node/local minimum X . Local minima at level 2 are the equivalent of funnel bottoms in the terminology of the above mentioned molecular conformation problems. Efficient algorithms for the detection of local minima at level 2 exist in the literature (see e.g. (Leary, 2000; Addis-Leyffer, 2004; Addis-et-al, 2005)). These algorithms are based on local moves between neighbor local minima, i.e.

¹ strictly speaking, these problems are unconstrained ones, but it is always possible to define a box where the global minimum is guaranteed to lie

local moves on the directed graph. Local minima at level 3 are defined similarly: they are the local minima over a directed graph where nodes are all the local minima at level 2 and a directed arc connects node \mathcal{X} to node \mathcal{Y} if a small neighborhood of node/local minimum \mathcal{X} at level 2 contains at least a point within the region of attraction of node/local minimum \mathcal{Y} at level 2 (see also Figure 2).

Again, it is possible to define algorithms aimed at detecting local minima at level 3 which are based on local moves within the directed graph whose nodes are the local minima at level 2. We remark that the above description is incomplete, many details, such as the definition of "region of attraction" of local minima at level 2, have been omitted. For these details we refer to (Locatelli, 2005). We also remark that in a completely similar way we could also define local minima at higher levels. Finally, we remark that detecting local minima at a given level becomes more and more difficult as the level increases. What we would like to recall here is the relation between the difficulty of a GO problem and the first level at which only one or few local minima exist. The easiest GO problems are those with one or few local minima (the standard ones or local minima at level 1). In these cases a single or few runs of a local search procedure started at random initial points is able to reach the global minimum². Problems with a large number of local minima are more difficult to solve. In this case the essential feature to establish the difficulty of the problem is not the number of local minima but their position. If many local minima (at level 1) exist but they are placed in a regular way in the search space, in particular in such a way that a single or few local minima at level 2 exist, then this case can be efficiently solved e.g. by any algorithm based on local moves between standard local minima. If not only many local minima at level 1 exist, but also many local minima at level 2, then the problem becomes more difficult. As previously mentioned, in the latter case we can employ more time-consuming algorithms based on local moves between local minima at level 2, aiming at detecting local minima at level 3. Of course, as we increase the number of local minima at level 3, we further increase the difficulty of the GO problem. Based on the above discussion, our intention is to introduce a class of test functions for which, by specifying few parameters, the user is able to define test functions with a given number of local minima at levels 1, 2 and 3 (we could also consider

² Of course, the classical golf-hole situation, where the global minimum has a very narrow region of attraction, is always possible, but here we omit to deal with this pathological case

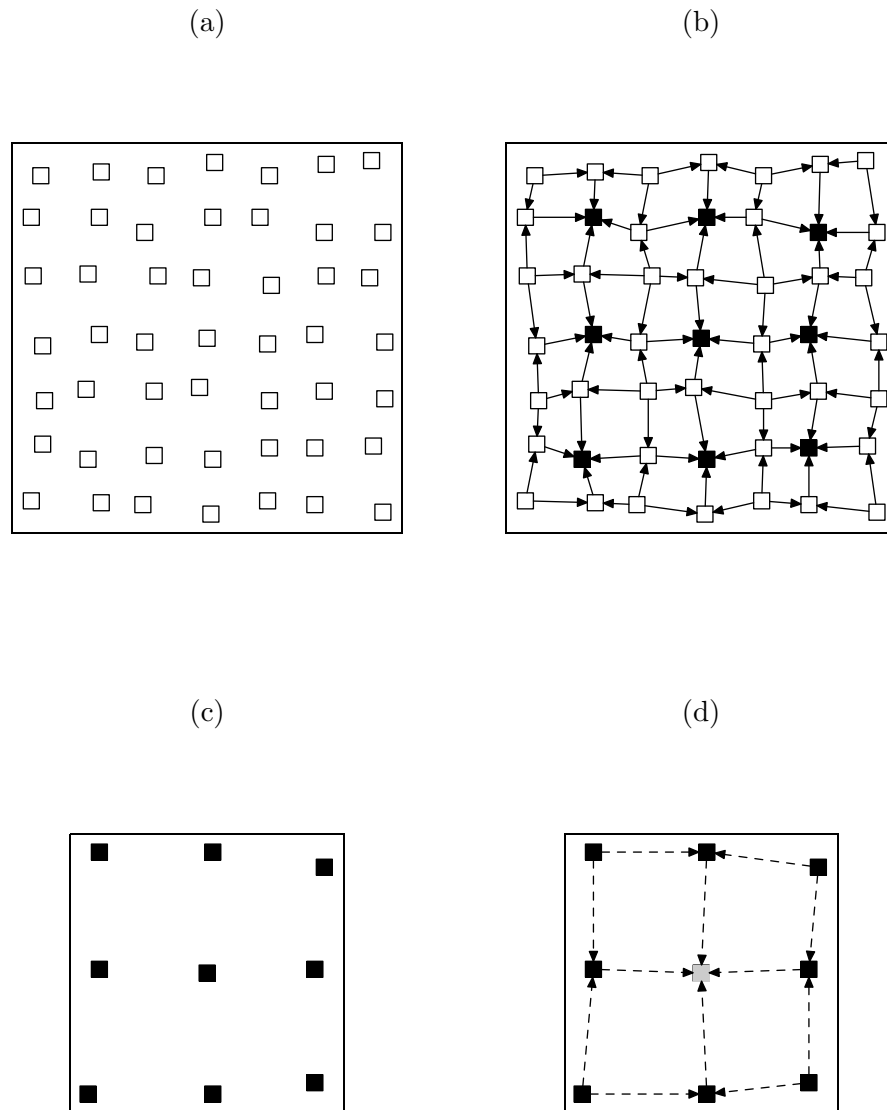


Figure 1. (a) local minima at level 1 (b) neighborhood structure between local minima at level 1 (c) local minima at level 2 (d) neighborhood structure between local minima at level 2: the grey square is the unique local minimum at level 3

higher levels but level 3 is already enough to define very challenging GO test functions), thus controlling the difficulty of the problem.

3. User-defined and random parameters

In this section we introduce the parameters which will be employed to define the test functions. Some of them have to be specified by the user, some others are randomly chosen by the test generator.

User-defined parameters

- n : the *number of basic variables*. Note that basic variables are not the unique variables of the test functions, other variables will be later introduced to combine different components of the test functions.
- L_2 : it is constrained to belong to the interval $[1, 2^{n+1} - 1]$ and controls the *number of local minima at level 2*.
- L_3 : the *number of local minima at level 3*. In view of the high difficulty introduced by local minima at level 3, we set the upper bound \sqrt{n} for this value; then, L_3 is constrained to belong to the interval $[1, \sqrt{n}]$.
- $K_i, i = 1, \dots, n$: the *oscillation frequencies*. They basically control the number of local minima in the one-dimensional components by which the test functions are made up; they are constrained to belong to the interval $[10, 20]$. The user has the option of fixing them all equal to a given value $K \in [10, 20]$ or to let the value of each of them be randomly chosen in the above interval. In order to guarantee a high enough variability among the K_i values, these are not uniformly sampled over the interval $[10, 20]$ but each of them is sampled with probability 0.5 in the interval $[10, 12.5]$ and with the same probability in the interval $[17.5, 20]$. We remark that the random choice of these values is a further source of difficulty because it introduces a different scaling of the variables. In what follows we will also denote by \bar{K} the average $\sum_{i=1}^n K_i/n$ of these values.
- H : the *oscillation width*. It controls the height of the barriers between neighbor local minima in the one-dimensional components; it is constrained to belong to the interval $[10, 30]$.

Random parameters

- $c_1 \in [-3.5, -2.0]$ defines the position of the first local minimum at level 2 in the one-dimensional components with two local minima at level 2 (see Section 4.1).
- $c_2 \in [2.0, 3.5]$ defines the position of the second local minimum at level 2 in the one-dimensional components with two local minima at level 2 (see Section 4.1).
- $p_i \in \{0, 1\}$: for each i corresponding to a one-dimensional component of the test function, p_i defines the position (c_1 or c_2) of the lowest local minimum at level 2 if the one-dimensional component has two local minima at level 2, or of the unique local minimum at level 2 if the one-dimensional component has a single local minimum at level 2 (see Section 4.1).

As it will become clear in what follows, the overall dimension d of the test functions will be defined as follows

$$d = n + \nu(L_2) + L_3 - 2,$$

where $\nu(L_2)$ is the number of ones in the binary code of L_2 . Although we can view the problems as unconstrained ones, it can be guaranteed that the global minimum lies within the sphere $S(0, 5\sqrt{d})$ centered at the origin and with radius $5\sqrt{d}$.

4. Building the test functions

Our aim is to build test functions with the required properties, in particular with the required number of local minima at levels 2 and 3. Before defining these functions, we need to underline that the concept of neighbor local minima at any level, which is essential to define local minima at the upper level, is not trivial to define and it is often algorithmic dependent (see (Locatelli, 2005)). Here minimum A is considered close to minimum B if it is possible to start a descending path towards B in a point “close” to A , i.e. within some distance r from A . Distance r should be chosen in such a way that two requirements are fulfilled: on one hand the neighborhood of a local minimum should be small enough in order to allow its efficient exploration, on the other hand r should be large enough to reduce as much as possible the number of local minima at the upper level. A very small value of r allows a very

small (possibly empty) neighborhood of the minima but the number of minima at the upper level is very large, while a very large value of r allows a small number of local minima at the upper level (possibly only one) but makes the neighborhood of the minima extremely large. As it will become clear during the description of the test functions, here we employ distance $r_1 = \frac{10}{\min_i K_i}$ to define the neighborhood of local minima at level 1 (standard local minima), and the distance $r_2 = c_2 - c_1$ to define the neighborhood of local minima at level 2.

4.1. THE ONE-DIMENSIONAL COMPONENTS

The first step towards the definition of the test functions is the introduction of some basic one-dimensional components with multiple minima. Following a common practice in the definition of test functions with many local minima (see e.g. the definition of the Rastrigin test function in (Törn-Žilinskas, 1989)), multiple minima are obtained through an *oscillation* term based on the cosine function

$$O_{c_1, c_2}^{K, H}(x) = -H \cos \left[2\pi \left\lceil \frac{K(c_2 - c_1)}{10} \right\rceil \frac{(x - c_1)}{c_2 - c_1} \right] + H.$$

This term has (approximately) K local minima (all with value equal to 0) in the interval $[-5, 5]$, two of which are in c_1 and c_2 .

Once the oscillation term has been introduced, we are ready to introduce the two types of one-dimensional components through which the test functions are built. The first type, the one-dimensional component with a single local minimum at level 2, is defined as follows

$$s_{p, c_1, c_2, K}(x) = \gamma_{p, c_1, c_2}(x) + O_{c_1, c_2}^{K, H}(x)$$

where:

$$\gamma_{p, c_1, c_2}(x) = \begin{cases} 0.5(x - c_2)^2 + 2 & \text{if } p = 0 \\ 0.5(x - c_1)^2 + 2 & \text{if } p = 1 \end{cases}$$

Note that the sum of the unimodal (actually convex) term γ and of the oscillation term O guarantees that this one-dimensional component has a unique local minimum at level 2 at $x = c_1$ (if $p = 1$) or $x = c_2$ (if $p = 0$) with value equal to 2.

The second type of one-dimensional component has two local minima at level 2 in $x = c_1$ and $x = c_2$ respectively with value 0 and 1 if $p = 0$ or with value 1 and 0 if $p = 1$. It is obtained through the sum of a bimodal function, denoted by ξ , with two local minima in $x = c_1$ and $x = c_2$, and the oscillation term O

$$d_{p, c_1, c_2, K}(x) = \xi_{p, c_1, c_2}(x) + O_{c_1, c_2}^{K, H}(x).$$

The bimodal function ξ is defined as follows:

$$\xi_{p,c_1,c_2}(x) = \begin{cases} \xi^a(x) = \sum_{r=0}^3 \alpha_r (x - c_1)^r & \text{if } -5 \leq x \leq 0 \\ \xi^b(x) = \sum_{r=0}^3 \beta_r (x - c_2)^r & \text{if } 0 \leq x \leq 5 \end{cases}$$

The equations to find the parameters α_r and β_r , $r = 0, 1, 2, 3$ are

$$\begin{aligned} \frac{d}{dx} \xi^a(c_1) = 0 \quad \frac{d}{dx} \xi^a(0) = 0 \quad \xi^a(c_1) = p \quad \xi^a(0) = 5 \\ \frac{d}{dx} \xi^b(c_2) = 0 \quad \frac{d}{dx} \xi^b(0) = 0 \quad \xi^b(c_2) = 1 - p \quad \xi^b(0) = 5 \end{aligned}$$

Such equations guarantee that: ξ has continuous first derivative, has a minimum in $x = c_1$ with value p , a minimum in $x = c_2$ with value $1 - p$, and a maximum in $x = 0$ with value 5.

4.2. BASIC FUNCTION WITH 2^m LOCAL MINIMA AT LEVEL 2

Next step is the introduction of basic n -dimensional components with 2^m ($m = 0, 1, \dots, n$) local minima at level 2. The easiest way to obtain these functions is by summing up one-dimensional components with one or two local minima at level 2. However, this way we would get separable functions. In order to avoid that explicit or implicit exploitation of the separability (e.g. by moves along single or few coordinates) leads to very good but misleading results, we remove separability. This is easily obtained e.g. by a one-to-one linear transformation of the original variables. In particular we consider the following distance-preserving transformation. Let $\mathbf{x} = \{x_1, \dots, x_n\}$ and

$$\mathbf{w}(x_1, \dots, x_n) = \mathbf{A}\mathbf{x} \tag{1}$$

where \mathbf{A} is a $n \times n$ orthonormal matrix obtained by randomly generating a nonsingular matrix \mathbf{A}' and then orthonormalizing it through the Gram-Schmidt process. Next we define the basic component with 2^m local minima at level 2, $0 \leq m \leq n$, as follows:

$$F_m(x_1, \dots, x_n) = \sum_{i=1}^m d_{c_1, c_2, p_i, K_i}(w_i(x_1, \dots, x_n)) + \sum_{i=m+1}^n s_{c_1, c_2, p_i, K_i}(w_i(x_1, \dots, x_n))$$

This function has a global minimum whose value is $2(n-m)$. The values w_i , $i = 1, \dots, n$, at the global minimum point are:

$$w_i = \begin{cases} c_1 & \text{if } (p_i = 0 \text{ and } i \leq m) \text{ or } (p_i = 1 \text{ and } i > m) \\ c_2 & \text{otherwise} \end{cases}$$

and once these values are known we can derive from (1) also the values of the original x_i variables.

4.3. COMBINING BASIC COMPONENTS

Let F_{m_2} and F_{m_1} be two basic components with $m_2 > m_1$. We want to build a new function G_t , $t = 2^{m_1} + 2^{m_2}$, by combining F_{m_1} and F_{m_2} , with t local minima at level 2. Given the combination operator \times , the combination operation is

$$G_t = F_{m_1} \times F_{m_2}$$

and its result is a function of the n basic variables x_1, \dots, x_n and of the *auxiliary* variable y :

$$G_t(x_1, \dots, x_n, y) = \begin{cases} G_t^a(x_1, \dots, x_n, y) + O_{-2.5, 2.5}^{\bar{K}, F_{m_1} + F_{m_2}}(y) & \text{if } y \leq 0 \\ G_t^b(x_1, \dots, x_n, y) + O_{-2.5, 2.5}^{\bar{K}, F_{m_1} + F_{m_2}}(y) & \text{if } y > 0 \end{cases}$$

where we recall that \bar{K} is equal to $\sum_{i=1}^n K_i/n$, while

$$G_t^a(x_1, \dots, x_n, y) = \sum_{r=0}^3 \mu_r (y + 2.5)^r$$

and

$$G_t^b(x_1, \dots, x_n, y) = \sum_{r=0}^3 \varphi_r (y - 2.5)^r.$$

Parameters μ_r and φ_r , $r = 0, 1, 2, 3$, are functions of the basic variables x_1, \dots, x_n whose values can be obtained by solving respectively the following systems:

$$\begin{cases} \frac{d G_t^a}{d y}(x_1, \dots, x_n, -2.5) = 0 \\ \frac{d G_t^a}{d y}(x_1, \dots, x_n, 0) = 0 \\ G_t^a(x_1, \dots, x_n, -2.5) = F_{m_1}(x_1, \dots, x_n) \\ G_t^a(x_1, \dots, x_n, 0) = 2(F_{m_1}(x_1, \dots, x_n) + F_{m_2}(x_1, \dots, x_n)) \end{cases} \quad (2)$$

and

$$\begin{cases} \frac{d G_t^b}{d y}(x_1, \dots, x_n, 2.5) = 0 \\ \frac{d G_t^b}{d y}(x_1, \dots, x_n, 0) = 0 \\ G_t^b(x_1, \dots, x_n, 2.5) = F_{m_2}(x_1, \dots, x_n) \\ G_t^b(x_1, \dots, x_n, 0) = 2(F_{m_1}(x_1, \dots, x_n) + F_{m_2}(x_1, \dots, x_n)) \end{cases} \quad (3)$$

From the above equations function G_t is built in such a way that it is equivalent to F_{m_1} for $y = -2.5$ and to F_{m_2} for $y = 2.5$. For each *fixed* x_1, \dots, x_n , function G_t has two minima in $y = -2.5$ and $y = 2.5$. Every local minimum $(\bar{x}_1, \dots, \bar{x}_n, \bar{y})$ with $\bar{y} \neq -2.5, 2.5$ can not be a

local minimum at level 2 for G_t . Indeed, consider the one-dimensional function

$$u(y) = G_t(\bar{x}_1, \dots, \bar{x}_n, y).$$

We have that \bar{y} is also a local minimum of u but any local minimum y' of u closer to -2.5 with respect to \bar{y} if $\bar{y} \leq 0$, or closer to 2.5 if $\bar{y} > 0$, has a function value lower than $u(\bar{y})$. Note that $(\bar{x}_1, \dots, \bar{x}_n, y')$ is not necessarily a local minimum for G_t but it is close to $(\bar{x}_1, \dots, \bar{x}_n, \bar{y})$ and it has a lower function value thus belonging to the region of attraction of a better local minimum. Then, we have that the local minima at level 2 of function G_t are all and only those of function F_{m_1} , obtained by taking $y = -2.5$, plus all those of function F_{m_2} , obtained by taking $y = 2.5$. The oscillating term $O_{-2.5, 2.5}^{\bar{K}, F_{m_1} + F_{m_2}}(y)$ has the effect of introducing a barrier separating the local minima at level 2 of the two basic functions F_{m_1} and F_{m_2} (note that this term also depends on variables x_1, \dots, x_n through F_{m_1} and F_{m_2}). The global minimum for G_t has the same value as the global minimum of F_{m_2} , while the global minimum point has the same $x_i, i = 1, \dots, n$ coordinates as the global minimum of F_{m_2} and $y = 2.5$. We also remark that the combination operator \times is not a commutative one.

Once we have defined a procedure to combine two basic component functions F_{m_1} and F_{m_2} , we can generalize the combination operator \times in such a way that each of its two arguments can either be one the basic component functions F_m or the result of previous combination operations. The details of the generalization are analogous to those for the combination of the two basic component functions F_{m_1} and F_{m_2} with just a slight difference. In order to describe this difference let us consider the case of the combination of a basic component function F_m with a function G_t obtained by previous combinations. We notice that function F_m only depends on the basic variables x_1, \dots, x_n , while function G_t depends on the basic variables *and* some auxiliary variables, say y_1, \dots, y_r . In order to make both functions depending on the same set of variables, we modify as follows the basic component function:

$$\tilde{F}_m(x_1, \dots, x_n, y_1, \dots, y_r) = F_m(x_1, \dots, x_n) + \sum_{j=1}^r \left[(y_j - 2.5)^2 + O_{-2.5, 2.5}^{\bar{K}, H}(y_j) \right]. \quad (4)$$

Note that \tilde{F}_m still has 2^m local minima at level 2, has the same global minimum value as F_m , and its global minimum point has the same x_1, \dots, x_n coordinates as the global minimum point of F_m while the y_j coordinates, $j = 1, \dots, r$, are all equal to 2.5 .

4.4. A FUNCTION WITH L_2 LOCAL MINIMA AT LEVEL 2

We are finally ready to define a procedure returning a function with a given number L_2 ($1 \leq L_2 \leq 2^{n+1} - 1$) of local minima at level 2. Let $\ell_m \ell_{m-1} \dots \ell_1 \ell_0$ be the binary code of L_2 . Let

$$J = \{j : \ell_j = 1\} = \{j_0, \dots, j_k\} \quad j_0 < \dots < j_k$$

Then we can employ the following procedure.

Initialization Set $G = F_{j_0}$ and $h = 1$.

Step 1 If $h > k$, then STOP and return G , otherwise go to Step 2.

Step 2 Set

$$G = G \times \tilde{F}_{j_h} \quad h = h + 1$$

(where \tilde{F}_{j_h} is the modification of the basic component F_{j_h} as in (4) with the additional y_1, \dots, y_{h-1} auxiliary variables) and go back to Step 1.

We remark that the resulting function has exactly L_2 local minima at level 2, the same global minimum value as F_{j_k} and a global minimum whose x_1, \dots, x_n coordinates are the same as those of F_{j_k} , while the y_1, \dots, y_k coordinates are all equal to 2.5.

4.5. A FUNCTION WITH L_3 LOCAL MINIMA AT LEVEL 3

Function G as built in the previous subsection has L_2 local minima at level 2 but it is possible to define a neighborhood structure between these minima in such a way that there is a unique local minimum at level 3. Therefore, the next step is to define functions with a given number L_3 of local minima at level 3. In order to obtain this function, first we can build L_3 different components G^j , $j = 1, \dots, L_3$, each one with L_2 local minima at level 2 and a single local minimum at level 3. These functions are obtained by the previously seen procedure but by introducing different random parameters p_i^j , $i = 1, \dots, n$, $j = 1, \dots, L_3$. This way the position of the global minimum (which is also the unique local minimum at level 3) is, very likely, different for each G^j component (there are 2^n different possible positions for the global minimum). Next we define a combination operator \times , similar to the one introduced in Section 4.3, between two components G^{j_1} and G^{j_2}

$$\Gamma = G^{j_1} \times G^{j_2}$$

and its result is a function of the n basic variables x_1, \dots, x_n , of the *auxiliary* variables y_1, \dots, y_k and of the further auxiliary variable z :

$$\Gamma(x_1, \dots, x_n, y_1, \dots, y_k, z) = \begin{cases} \Gamma^a(x_1, \dots, x_n, y_1, \dots, y_k, z) + O_{-2.5, 2.5}^{\bar{K}, G^{j_1} + G^{j_2}}(z) & \text{if } z \leq 0 \\ \Gamma^b(x_1, \dots, x_n, y_1, \dots, y_k, z) + O_{-2.5, 2.5}^{\bar{K}, G^{j_1} + G^{j_2}}(z) & \text{if } z > 0 \end{cases}$$

where:

$$\Gamma^a(x_1, \dots, x_n, y_1, \dots, y_k, z) = \sum_{r=0}^3 \eta_r (z + 2.5)^r$$

and

$$\Gamma^b(x_1, \dots, x_n, y_1, \dots, y_k, z) = \sum_{r=0}^3 \tau_r (z - 2.5)^r$$

Parameters η_r and τ_r , $r = 0, 1, 2, 3$, are functions of the variables x_1, \dots, x_n and y_1, \dots, y_k and can be obtained by the following equations, similar to (2) and (3):

$$\begin{cases} \frac{d}{dz} \Gamma^a(x_1, \dots, y_k, -2.5) = 0 \\ \frac{d}{dz} \Gamma^a(x_1, \dots, y_k, 0) = 0 \\ \Gamma^a(x_1, \dots, y_k, -2.5) = G^{j_1}(x_1, \dots, y_k) + 1 \\ \Gamma^a(x_1, \dots, y_k, 0) = 2(G^{j_1}(x_1, \dots, y_k) + G^{j_2}(x_1, \dots, y_k)) + 2 \end{cases}$$

and

$$\begin{cases} \frac{d}{dz} \Gamma^b(x_1, \dots, y_k, 2.5) = 0 \\ \frac{d}{dz} \Gamma^b(x_1, \dots, y_k, 0) = 0 \\ \Gamma^b(x_1, \dots, y_k, 2.5) = G^{j_2}(x_1, \dots, y_k) \\ \Gamma^b(x_1, \dots, y_k, 0) = 2(G^{j_1}(x_1, \dots, y_k) + G^{j_2}(x_1, \dots, y_k)) + 2 \end{cases}$$

Function Γ is built in such a way that it is equivalent to $G^{j_1} + 1$ for $z = -2.5$ and to G^{j_2} for $z = 2.5$. Then, we have that function Γ has two local minima at level 3, $2L_2$ local minima at level 2, its global minimum value is the same as the one for G^{j_2} , while its global minimum point has the same x_i , $i = 1, \dots, n$, and y_j , $j = 1, \dots, k$, coordinates as the global minimum of G^{j_2} and $z = 2.5$. Similarly to Section 4.3, we can generalize the combination operation in such a way that each of its two arguments can either be one the basic component functions G^j (or even better its modification \tilde{G}^j completely analogous to (4) with z variables in place of the y variables) or the result of previous combination operations.

Now, the following procedure, again similar to the one described in Section 4.3, returns a function with L_3 local minima at level 3.

Initialization Set $\Gamma = G^1$ and $h = 2$.

Step 1 If $h > L_3$, then STOP and return Γ , otherwise go to Step 2.

Step 2 Set

$$\Gamma = \Gamma \times \tilde{G}^h \quad h = h + 1$$

and go back to Step 1.

Function Γ has L_3 local minima at level 3, L_3L_2 local minima at level 2, its global minimum value is the same as the one for G^{L_3} and its global minimum point has the same x_1, \dots, x_n and y_1, \dots, y_k coordinates as G^{L_3} , while the z_1, \dots, z_{L_3-1} coordinates are all equal to 2.5.

5. Computational experiments

If we are given a class of global optimization problems, our aim is to find algorithms which are able to solve them efficiently. If we want to propose a class of challenging test functions the situation is somehow reversed: we are given existing algorithms and we would like to find problems on which these algorithms are not efficient in order to stimulate the search of new techniques. Of course it is impossible to test all the existing GO algorithms. Therefore, for those willing to test their GO algorithms on the proposed test functions, we make available a C++ class, downloadable at the web site (GOL, 2005), through which the functions, together with their gradients, are computed. Any feedback is more than welcome and at the same web site all updates based on the received feedbacks will be indicated.

Although we can not test all GO algorithms, we can at least propose, as a basis for future comparison, the results obtained by two simple GO algorithms, both based on multiple local searches, on some test functions of increasing difficulty from the class proposed in this paper. The first algorithm is the very simple Multistart algorithm, where at each iteration a random point within the sphere $S(0, 5\sqrt{d})$ is sampled and a local search is started from it. Tests with Multistart are basically only performed to experimentally confirm that even clever local search procedures (like the limited memory BFGS employed in this paper) are trapped by the many local minima of these functions. The second algorithm is Monotonic Basin Hopping (see (Leary, 2000)), denoted by MBH in what follows:

MBH

0. Let `MaxNoImprove` be a fixed parameter and X_0 be a random initial local minimum; set $h, k = 0$.

1. Let Z_{k+1} be obtained by random sampling in the sphere $S(X_k, r)$ of radius r , and Y_{k+1} be the local minimum reached by starting a local search from Z_{k+1} ;
2. if $f(Y_{k+1}) < f(X_k)$, then set $X_{k+1} = Y_{k+1}$ and $h = 0$, otherwise set $X_{k+1} = X_k$ and $h = h + 1$;
3. if $h \geq \text{MaxNoImprove}$, then STOP, otherwise set $k = k + 1$ and go back to 1.

In spite of its simplicity, MBH turned out to be extremely efficient for very challenging global optimization problems, like molecular conformation problems. It can be viewed as a local search at level 2, because if the perturbation r is appropriately chosen, MBH is able to detect quite efficiently local minima at level 2. Of course the choice of the random perturbation is a key issue. Tuning r is not trivial, but since our aim is to propose test functions which are challenging for MBH even when its parameters are carefully chosen, we always report the best results obtained with different choices of r . It is still possible that tuning the value of r more carefully for every specific test, better results can be obtained but the current choice appears to be reasonably good. The parameter r is chosen as a function of \bar{K} , i.e. $\frac{r_0}{\bar{K}}$. The `MaxNoImprove` parameter has been set to a very large value (10^5) in order to be reasonably sure that a local minimum at level 2 has been reached.

Seven different test functions have been considered. The parameters defining these functions can be downloaded at the web site (GOL, 2005). As expected, Multistart was unable to detect the global minimum using a number of local searches double with respect to MBH even in the easiest (with respect to MBH) case (test function with $L2 = L3 = 1$, $H = K = 10$). Results for MBH are reported in Table 5. On each test function 1000 runs have been performed. For each function we report the number of successes over the 1000 runs and the average number of local searches per run (always excluding in each run those performed during the last 10^5 iteration where no improvement is observed). MBH always solves the cases with $L2 = L3 = 1$. It can be seen that increasing the number of local minima (i.e. increasing parameter K from 10 to 20) only slightly worsens the performance of MBH, while different scaling, due to random selection (within the interval $[10, 20]$) of the K_i values, is a more serious source of difficulty (the average number of local searches increases) because the random generation of the point Z_{k+1} over a sphere centered at the current record X_k (Step 1. of MBH) does not take into account the different scaling of the variables. As the values L_2 and L_3 are increased, we

Table I. Results of MBH over the seven proposed test functions

Function	n	K_i	H	L_2	L_3	successes	average #LS
Test 1	50	$K_i = 10, \forall i$	10	1	1	1000	1517
Test 2	50	$K_i = 20, \forall i$	10	1	1	1000	2393
Test 3	50	K_i random in [10, 20]	10	1	1	1000	5271
Test 4	30	$K_i = 10, \forall i$	10	10	1	82	1444
Test 5	30	$K_i = 10, \forall i$	10	25	1	30	1810
Test 6	30	$K_i = 10, \forall i$	10	25	4	17	1781
Test 7	30	$K_i = 10, \forall i$	10	100	4	3	1867

observe a clear decrease of the performance of MBH (MBH gets trapped at a local minimum at level 2 and is unable to escape from it when this is not the global minimum). Notice that MBH reaches in a relatively fast time a local minimum at level 2 (it has been observed that MBH always stops at a local minimum at level 2 and the number of local searches per run is never very large) but then, as previously observed, is unable to escape from it.

6. Conclusion

In this paper we have proposed a class of test functions for unconstrained global optimization problems. The difficulty of these problems can be controlled by an appropriate choice of some parameters. A web site ((GOL, 2005)) is maintained where users will be able to download a C++ class of the test functions, post their comments and get the parameters defining a set of seven test functions with increasing difficulty for which we also make available the results obtained by two simple GO algorithms as a basis of comparison.

Acknowledgements

The authors acknowledge support from Progetto FIRB ‘‘Ottimizzazione Non Lineare su Larga Scala’’.

References

- Addis B., Locatelli M., Schoen F., Local optima smoothing for global optimization, to appear in *Optimization Methods and Software* (2005)
- Addis B., Leyffer S., A trust-region algorithm for global optimization, submitted, Preprint ANL/MCS-P1190-0804, Argonne National Laboratory, Mathematics and Computer Science Division (2004)
- Baritomba W.P., Dr M., Hendrix E.M.T., Noakes L., Pullan W.J. and Wood G.R., Matching stochastic algorithms to objective function landscapes, to appear in *Journal of Global Optimization* (2005)
- Dixon L.C.W., Szegö G.P., *Towards Global Optimization 2*, North-Holland, Amsterdam, The Netherlands (1978)
- Floudas, C. A., Pardalos, P. M., Adjiman C. J., Esposito W. R., Gumus Z. H., Harding S. T., Klepeis J. L., Meyer C. A., Schweiger C. A (1999). *Handbook of Test Problems in Local and Global Optimization*, volume 33 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht.
- Gaviano M., Kvasov D.E., Lera D., Sergeev Y.D., Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization, *ACM Transactions on Mathematical Software*, 29, 469-480 (2003)
- Global Optimization web site by A. Neumaier,
<http://www.mat.univie.ac.at/~neum/glopt/test.html>
- Global Optimization Laboratory web site,
http://globopt.dsi.unifi.it/gol/test_functions
- Lavor C., Maculan N., A function to test methods applied to global minimization of potential energy of molecules, *Numerical Algorithms*, 35, 287-300 (2004)
- Leary R. H., Global optimization on funneling landscapes, *J. Global Optim.*, 18, 367-383, (2000).
- Locatelli M., On the multilevel structure of global optimization problems, *Computational Optimization and Applications*, 30, 5-22 (2005)
- Locatelli, M., Wood G.R., Objective function features providing barriers to rapid global optimisation, to appear in *Journal of Global Optimization* (2005)
- Neumaier A., Shcherbina O., Huyer W., Vinko T., A comparison of complete global optimization solvers, *Mathematical Programming*, 103(2), 335-356 (2005)
- Pinter J.D., Global Optimization: software, test problems and applications, in *Handbook of Global Optimization, Volume 2*, Pardalos P.M. and Romeijn H.E., eds., Kluwer Academic Publishers, Dordrecht Boston London (2002)
- Törn, A. and Žilinskas, A. *Global Optimization*, Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, (1989)
- D. J. Wales and J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, *J. Phys. Chem. A*, 101, 5111-5116, (1997).