Journal of Global Optimization (2005) 31: 549–565
DOI 10.1007/s10898-004-9965-1

Objective Function Features Providing Barriers to Rapid Global Optimization

M. LOCATELLI[1] and G.R. WOOD[2]
[1]*Dipartimento di Informatica, Universitá di Torino, Corso Svizzera, 185 10149 Torino, Italy*
[2]*Department of Statistics, Macquarie University, North Ryde NSW 2109, Australia (e-mail: gwood@efs.mor.edu.au)*

Abstract. The purposes of this discussion paper are twofold. First, features of an objective function landscape which provide barriers to rapid finding of the global optimum are described. Second, stochastic algorithms are discussed and their performance examined, both theoretically and computationally, as the features change. The paper lays a foundation for the later findings paper.

Mathematics Subject Classifications. 90C65, 90C30, 65K05

Key words: Local minima, Local optimization, Search region, Simulated annealing, Stochastic global optimization

1. Introduction

This paper develops some background to the question "Given an objective function f, what characteristics of an algorithm will efficiently find the global minimum of f?" In order to make some initial progress, this question requires us to describe

- The class of objective functions to be considered.
- The class of algorithms to be considered.
- A method for determining algorithm efficiency.

This paper addresses the first and second points. The class of objective functions should capture features which render a global optimization problem difficult. Intuition suggests that the number of local minima, the visibility of local minima and global minima and the orderliness of local minima will be key factors. These notions are discussed in Section 2. The broader goal is to tailor a stochastic algorithm to the features of a landscape. To this end, associated stochastic algorithms are discussed in Section 3, and first steps made to matching them to landscapes, using both theory and computation.

2. Objective Function Features

In this section we introduce features of objective functions which appear to be essential in establishing the difficulty of a global optimization problem. These features are

1. The degree of modality, also at a higher level in a sense to be specified (Section 2.1)
2. The size of the basins of attraction of local minima, again with an extension at a higher level (Section 2.2)
3. The size of improving regions and, strictly related to this, the magnitude of oscillations (Section 2.3)
4. The degree of randomness in the positions of the minima (Section 2.4).

We will omit to discuss, but note it here, another obviously important feature, namely the dimension of the search space (the well-known curse of dimensionality).

In order to raise the discussion above an abstract level these features will be illustrated with computations in Section 3.1, using test functions taken from the literature. These are the Rastrigin function

$$\text{Rastrigin}(x) = \sum_{i=1}^{n} [x_i^2 - k\cos(2\pi x_i)] + nk, \quad x_i \in [-5.12, 5.12],$$

where k is a positive parameter, and the Schwefel function

$$\text{Schwefel}(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}), \quad x_i \in [-500, 500].$$

We will also refer to the Lennard–Jones (LJ) function, which is derived from mathematical models of the energy of cluster of atoms, but also represents a very challenging global optimization problem

$$\text{LJ}(X_1, \ldots, X_N) = \frac{1}{2} \sum_{i,j=1,\ldots,N,\ i \neq j} \frac{1}{||X_i - X_j||^{12}} - \frac{2}{||X_i - X_j||^6}, \quad X_i \in R^3,$$

where N is the number of atoms and X_i represents the position of atom i.

2.1. MULTIMODALITY

A well known property on which the difficulty of a global optimization problem depends is the degree of modality of the objective function f, that is, the number of local minima of the function. The global minimum of a unimodal function can be detected by a single run of a local search routine, whereas highly multimodal functions provide more challenging problems. Inside the class of highly multimodal functions it is possible to make a further classification which depends not only on the number of local min-

ima but also on a concept of modality at a "higher level". In order to specify this concept some definitions are needed.

Given the global optimization problem $\min_{x\in X} f(x)$, $X \subset R^n$ and compact, we consider a parameter $\alpha > 0$ and introduce the oriented graph $G_\alpha = (V, \mathcal{A}_\alpha)$, where each node $v \in V$ corresponds to a local minimum of f over X (it is assumed that there are finitely many of them). Also

$$(v_1, v_2) \in \mathcal{A}_\alpha \Leftrightarrow f(v_1) \geqslant f(v_2) \text{ and } v_2 \in S(v_1, \alpha),$$

where

$$S(v_1, \alpha) = \{x \in X : |x^i - v_1^i| \leqslant \alpha,\ i = 1, \ldots, n\} \tag{1}$$

is the hypercube centred at v_1 with edge length of 2α. This means that there exists (at least) one arc between any two local minima which are close enough to each other (where the definition of close depends on the choice of the parameter α) and the direction of the arc depends on their relative function values. The set

$$\mathcal{N}_\alpha(v) = \{w \in V : (w, v) \in \mathcal{A}_\alpha \text{ or } (v, w) \in \mathcal{A}_\alpha\}$$

is the α-neighborhood of a local minimum v. Note that a more appropriate choice would have been use of a different parameter α_i for each direction $i = 1, \ldots, n$, in order to take into account the different behaviour of f with respect to the different variables. For the sake of simplicity we restrict our attention to the single parameter case. The set

$$\mathcal{D}_\alpha(v) = \{w \in V : (v, w) \in \mathcal{A}_\alpha \text{ and } (w, v) \notin \mathcal{A}_\alpha\} \subseteq \mathcal{N}_\alpha(v)$$

is the set of α-neighbours of v with a strictly lower function value. Now we are ready for the definition of local minimum at a higher level.

DEFINITION 1. A point $v^* \in X$ is an α-High Level Local Minimum (α-HLLM in what follows) if it is a local minimum of f over X and if

$$\mathcal{D}_\alpha(v^*) = \emptyset.$$

Equivalently, an α-HLLM is a local minimum of f over the graph G_α. Given a definition of local minimum at a higher level, we also need a local search at a higher level, that is, an algorithm which is able to detect α-HLLM. Such an algorithm is now presented.

ALGORITHM 1.

Step 1. Randomly choose a starting point $v \in V$.

Step 2. If v is an α-HLLM, then stop. Otherwise go to Step 3.

Step 3. Explore $\mathcal{N}_\alpha(v)$ until a point $v' \in \mathcal{D}_\alpha(v)$ is detected. Then set $v = v'$ and go back to Step 2.

Of course, this is not a real algorithm for many reasons. First, it is necessary to specify how to explore the α-neighborhood $\mathcal{N}_\alpha(v)$. Second, the stopping rule "stop when an α-HLLM has been detected" is an ideal one, since it would be extremely hard to establish whether a point is an α-HLLM (it is

even *NP*-hard to establish whether a given point is a local minimum, see [8]). Finally, in a real algorithm it is more likely that α is not a fixed value but is adaptively updated during the exploration in Step 3. For the purposes of our analysis, however, this incomplete and unrealistic algorithm is what we need.

We note that Algorithm 1 always returns an α-HLLM. We may therefore need to start different runs of this algorithm before detecting a global minimum (note that global minima are always also α-HLLM, for any value of α). For this reason it is desirable that the number of α-HLLM be as low as possible. In the case of a single global minimum, the best situation appears to be the one where a single α-HLLM (exactly the global minimum) exists; in such a case a single run of Algorithm 1 immediately detects the global minimum. But it is actually always possible to have a single α-HLLM. Indeed, by choosing a sufficiently large value of α, the graph G_α is complete and the unique α-HLLM is the global minimum.

A low number of α-HLLM is not the only desirable property, since we should also take into account the cost of a run of the algorithm. If a very large value of α is chosen, then a single run is needed but the exploration of the α-neighborhoods $\mathcal{N}_\alpha$ becomes very expensive. In particular, when v is close to the global minimum, the cardinality of the improving set $\mathcal{D}_\alpha(v)$ is very small compared to that of $\mathcal{N}_\alpha(v)$ and a very large amount of time can be spent in searching for a point in $\mathcal{D}_\alpha(v)$. In order to make the exploration of $\mathcal{N}_\alpha(v)$ more efficient we need small values of α.

We therefore have two (possibly) conflicting aims: keeping the number of α-HLLM as low as possible (which may require large α values), and keeping the exploration of the α-neighbourhoods $\mathcal{N}_\alpha$ as efficient as possible. Note that this is a typical issue in the definition of neighbourhoods in combinatorial problems and our global optimization problem, once we only consider local minima, actually reduces to a combinatorial one. It is sometimes possible to choose values for α which satisfy both the aims. This is the case, for instance, with the Rastrigin function, for which, even with relatively small values of α, a unique α-HLLM exists in spite of the huge number of local minima. On the other hand, it is not the case for the Schwefel function, for which a single α-HLLM exists only with large values of α, while smaller values of α produce a very large number of α-HLLM. Functions like the Rastrigin one therefore appear to be easier to optimize than the Schwefel, in spite of the fact that all of these functions have a comparable number of local minima (if the same dimension n is chosen for all). Of course, appropriate values of α are usually not known in advance, but for the Rastrigin function such values exist and can possibly be detected by some adaptive update scheme, while for the Schwefel function such values do not even exist. Some computational results presented in Section 3.1 will support the observations above.

We can summarize the contents of this section by underlining that the difficulty of a global optimization problem not only depends on the number of local minima but also on how they are placed in the search space: highly multimodal problems where the number of α-HLLM is small even for small α values can be considered as "easy" ones within the class of highly multimodal problems.

2.2. SIZE OF BASINS OF ATTRACTION

In the previous Section we related the difficulty of a global optimization problem to the number of local minima and, in particular, to the number of local minima at a higher level. To be more precise, we should have mentioned that the difficulty is related to the size of the basin of attraction of the global minimum (the global minimum is again assumed to be unique for simplicity). Given a local search routine, the basin of attraction $A(x^*)$ of a local minimum x^* is the set of points $y \in X$ starting from which the local search routine finally returns x^*. As a measure of the size of the basin of attraction we can consider the probability that a uniformly chosen random point in X belongs to $A(x^*)$. If a function f is highly multimodal, but the size of the basin of attraction of the global minimum is very large compared to the size of each of the basins of attraction of other local minima, then the problem is not a difficult one (a local search started from a random initial point is very likely to end up in the global minimum).

Now we can extend the definition of the size of the basin of attraction to α-HLLM: given an α-HLLM v^*, the size of its basin of attraction is the probability that, by starting from a random initial point in V, Algorithm 1 will end up in v^*. Again, if for small values of α (values for which the exploration of the α-neighbourhoods is still efficient) the number of α-HLLM is large but the size of the basin of attraction of the α-HLLM corresponding to the global minimum is also very large, then the problem is "easy" because only a single run of Algorithm 1 is very likely to be needed.

For instance, the easier of the two functions in Figures 1 and 2 is the one in Figure 1. They both have (for suitable choice of α) only two α-HLLM (v_1^* and v_2^* in the figures) and even though it is not explicitly computed, it is clear that the size of the basin of attraction of the α-HLLM corresponding to the global minimum is larger for the function in Figure 1 than for the function in Figure 2.

2.3. IMPROVING REGIONS AND MAGNITUDE OF OSCILLATIONS

Algorithm 1 only moves from local minima to local minima by exploring the α-neighbourhood $\mathcal{N}_\alpha(v)$ of a local minimum v until a better local minimum, that is, a local minimum in $\mathcal{D}_\alpha(v)$, is detected. It is possible to

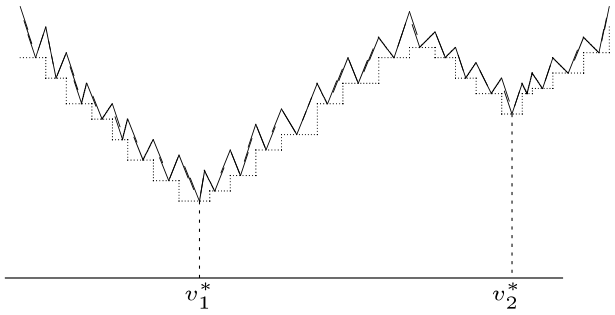

Figure 1. A function with two α-HLLM and a large basin of attraction of the global minimum v_1^*.

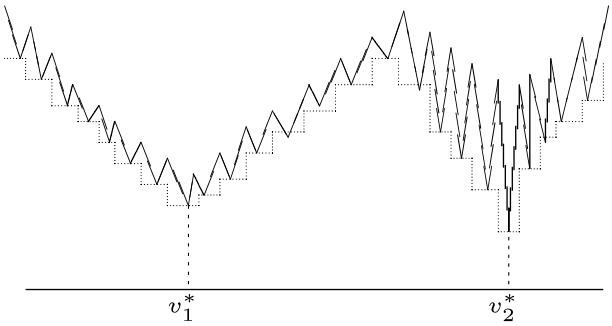

Figure 2. A function with two α-HLLM and a small basin of attraction of the global minimum v_2^*.

change it in such a way that at each iteration the α-neighbourhood $\mathcal{N}_\alpha(y)$ of a general point in $y \in X$ (not necessarily a local minimum) is explored until a better point $z \in \mathcal{D}_\alpha(y)$ (again, not necessarily a local minimum) is detected. (Note that the definitions of $\mathcal{N}_\alpha$ and $\mathcal{D}_\alpha$ can be easily extended to general points.) In Algorithm 1, the efficiency of the exploration step is related to the measure of $\mathcal{D}_\alpha(y)$ relative to $\mathcal{N}_\alpha(y)$: if the measure of $\mathcal{D}_\alpha(y)$ is very small compared to that of $\mathcal{N}_\alpha(y)$, then we are very likely to spend a long time before detecting a point in $\mathcal{D}_\alpha(y)$ (unless the problem has some structure which enables us to favour the detection of improving points with respect to worsening ones). Furthermore, the positions of the local minima and the choice of α are the only features influencing the relative measure of $\mathcal{D}_\alpha$ with respect to $\mathcal{N}_\alpha$. When we consider the algorithm moving between general points, however, we also have to take into account the behaviour of the function in the regions around the local minima.

A common way to control this behaviour in test functions is by introducing a parameter which determines the magnitude of the oscillations of the function. This is the case, for instance, with the parameter k appearing in the Rastrigin function: as k increases, the magnitude of the oscillations

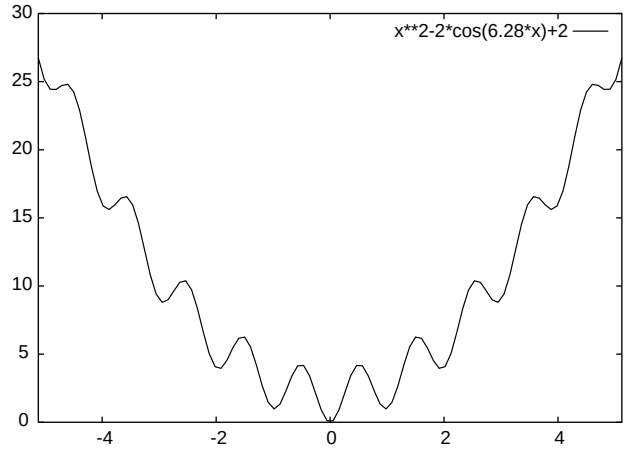

Figure 3. The Rastrigin function with $k = 2$.

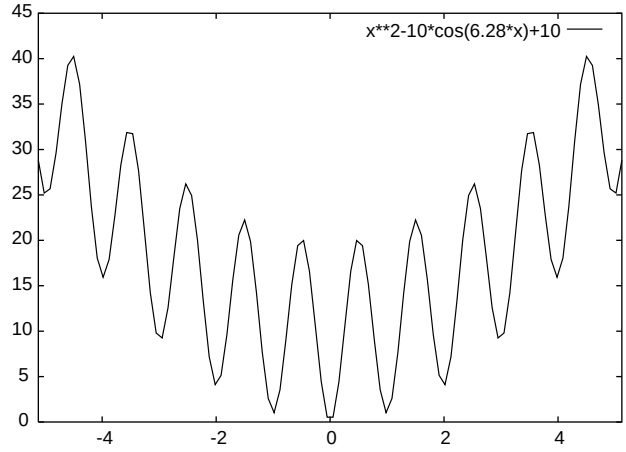

Figure 4. The Rastrigin function with $k = 10$.

of the function also increases and the regions of improvement shrink (see Figures 3 and 4 for the cases $k = 2$ and $k = 10$). An analogous role is played by the parameter $\mu > 0$ in the following class of functions $g : [0, \infty) \to [0, \infty)$, given by

$$g(x) = g_t(x) \quad \text{for} \quad x \in [t - 1/2, t + 1/2], \ t \in N, \tag{2}$$

where

$$g_t(x) = \begin{cases} -2\mu x + t(1 + 2\mu) & x \in [t - 1/2, t], \\ 2(1 + \mu)x - t(1 + 2\mu) & x \in [t, t + 1/2]. \end{cases}$$

Here N denotes the natural numbers. It is easy to see that for $\alpha = 1$ the relative measure of $\mathcal{D}_\alpha(t)$ with respect to $\mathcal{N}_\alpha(t)$ is, for any $t \in N$, equal to

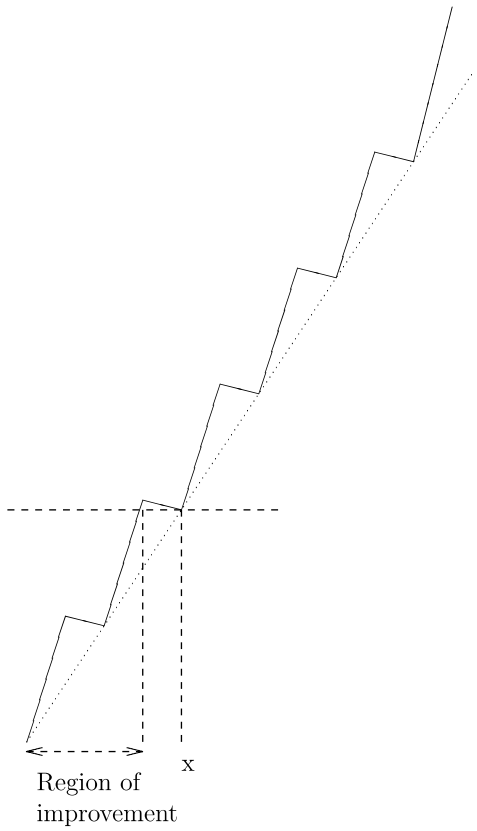

Figure 5. Region of improvement for the point x (small oscillations).

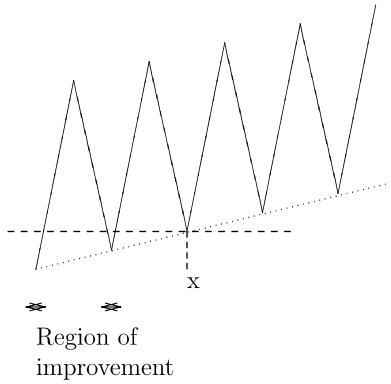

Figure 6. Region of improvement for the point x (large oscillations).

$1/(4(1+\mu))$. Therefore, for small values of μ (Figure 5) we have smaller oscillations and larger regions of improvement than for large μ values (Figure 6).

In view of the already remarked importance, for the efficiency of the exploration step, of the relative measure of $\mathcal{D}_\alpha$ with respect to $\mathcal{N}_\alpha$, increasing the magnitude of oscillations (and, consequently, shrinking the regions of improvement) has a strong (negative) impact on the algorithm moving between general points. On the other hand, the impact is definitely less strong for Algorithm 1. We would even expect Algorithm 1 to be unaffected by a variation in the magnitude of oscillations because it only uses the positions of the local minima, not what happens around them. While this is true in theory, in practical implementations the magnitude of oscillations has some impact even on algorithms moving only between local minima. This fact will be further discussed in Section 3.1.

2.4. DEGREE OF RANDOMNESS IN THE POSITIONS OF HIGH LEVEL LOCAL MINIMA

A run of Algorithm 1 returns an α-HLLM. If the number of α-HLLM is large or, more precisely, if the basin of attraction of the one corresponding to the global minimum is very small, we can expect to have to run Algorithm 1 many times before detecting the global minimum. We may wonder whether the results of the different runs should simply be forgotten or should be collected and combined in order to assist in the detection of new minima. In some cases collecting and combining α-HLLM may lead to the detection of the global minimum in a much faster way than by merely running Algorithm 1 many times until the global minimum is detected. This is strictly related to the degree of randomness in the positions of α-HLLM.

Consider Figure 7 where the four α-HLLM (for some suitable choice of α, say $\alpha = 200$) of the Schwefel function for $n = 2$ are displayed (see Figure

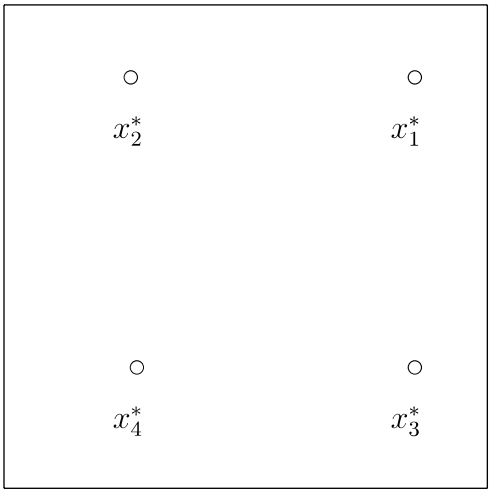

Figure 7. Locations of four α-HLLM of the Schwefel function, for $n = 2$.

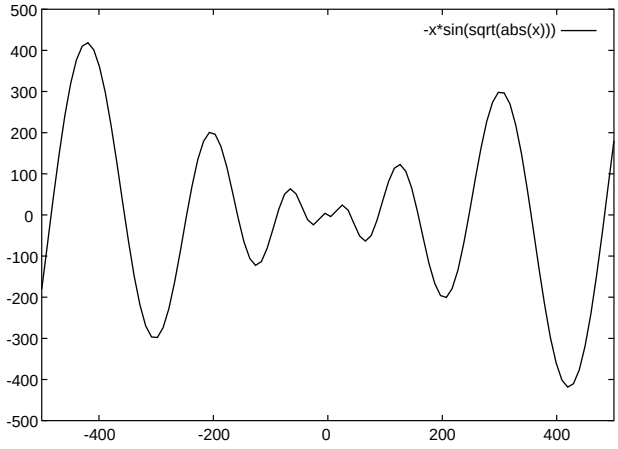

Figure 8. The one-dimensional Schwefel function.

8 for the one-dimensional Schwefel function). It is immediately clear that the α-HLLM are not randomly positioned. We can profit from this in order to speed up the detection of the global minimum. For instance, assume that Algorithm 1 has already returned the α-HLLM x_2^* and x_3^*. Then by combining them, for instance using the crossover operation of genetic algorithms (taking one component from x_2^* and the other one from x_3^*), we can immediately detect the global minimum x_1^*. The regular pattern followed by the α-HLLM of the Schwefel function is a consequence of the separability of this function.

More generally, each time the positions of α-HLLM follow a regular pattern, we may hope to increase efficiency in detecting the global minimum by exploiting the information given by the set of already detected α-HLLM. Of course, the regular pattern followed by the positions of α-HLLM is not usually known and, consequently, how to combine previously detected α-HLLM is not immediately evident. If some regularity is present, however, we may hope to detect and exploit it. This would not be possible if the positions of the α-HLLM were basically random as in Figure 9.

3. Results

We turn now to further development of algorithms and then examine how they perform as objective function landscapes change.

3.1. THEORETICAL RESULTS

In this section we discuss how the features of objective functions introduced so far affect algorithm performance, using some theoretical results.

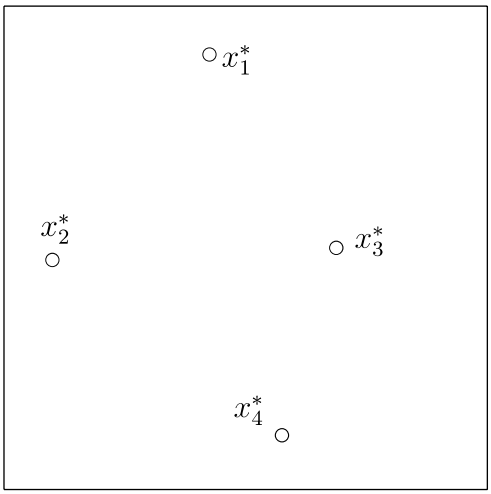

Figure 9. Locations of four irregularly positioned α-HLLM.

In order to do this we first propose a practical implementation of Algorithm 1.

ALGORITHM 2.

Step 1. Generate a uniformly distributed point $y \in X$ and start a local search from y. Let v be the detected local minimum.

Step 2. Generate a uniformly distributed point $z \in S(v, \alpha)$ ($S(v, \alpha)$ is defined in (1)) and start a local search from z. Let v' be the detected local minimum.

Step 3. If $f(v') < f(v)$, then set $v = v'$.

Step 4. If a stopping rule is satisfied, then stop. Otherwise go back to Step 2.

Algorithm 2 is not an exact implementation of Algorithm 1. First of all, since we do not usually have the opportunity of checking whether a point is an α-HLLM, the stopping rule is generally "stop if, after a given number of iterations, no improving point has been detected". Moreover, the local minimum v' generated in Step 2 does not necessarily belong to $\mathcal{N}_\alpha(v)$. The α-neighborhood explored in Step 2 is slightly different from $\mathcal{N}_\alpha(v)$ because it contains any local minimum whose basin of attraction has a nonempty intersection with $S(v, \alpha)$. In spite of these differences, Algorithm 2 can be considered a reasonable approximation of Algorithm 1. We also underline at this point that Algorithm 2 has not been introduced here merely for the purposes of our analysis, but has been very succesfully applied for the minimization of Lennard–Jones clusters (see [2, 5, 6]). Moreover, the Variable Neighbourhood Search algorithm, discussed for example in [3], is very

similar to Algorithm 2, the only difference being that at each iteration neighbourhoods of increasing size are explored, not just a single neighbourhood.

In Section 2.3 we discussed the possibility of exploring not only the local minima in the α-neighbourhood of the current point but also of exploring any point in this neighbourhood. We observed that such an algorithm is very sensitive to the increase in the magnitude of oscillations (and, consequently, the decrease of the size of the regions of improvement). This is not the case for Algorithm 1, which ignores what happens in the region around the local minima and only cares about the positions of the local minima. In order to probe this further, we now present a practical implementation of an algorithm moving between general points and apply it and Algorithm 2 to a simple example.

ALGORITHM 3.

Step 1. Generate a uniformly distributed point $y \in X$ and start a local search from y. Let v be the detected local minimum.

Step 2. Generate a uniformly distributed point $z \in S(v, \alpha)$.

Step 3. If $f(z) < f(v)$, then start a local search from z. Let v' be the detected local minimum and set $v = v'$.

Step 4. If a stopping rule is satisfied, then stop. Otherwise go back to Step 2.

Compared to the algorithm discussed in Section 2.3, a local search is now started as a point in $\mathcal{D}_\alpha(v)$ is detected (see Step 3). This modification simplifies the analysis (we only need to consider regions of improvement of local minima) without affecting the conclusions.

Now let us consider Algorithms 2 and 3 with $\alpha = 1$ (again, as soon as α is large enough, different values would not affect the conclusions) applied to the following problem

$$\min_{x \in [0,d]} g(x) \quad d \in N, \tag{3}$$

where g is the function defined in (2). Let the starting point at Step 1 be $y = d$ for both algorithms. How long do we have to wait before detecting the global minimum $v^* = 0$? We note that the probability of detecting a better local minimum is equal to 1/4 for Algorithm 2 if we assume that the basin of attraction of each local minimum $v = t$, $t \in N$, is the interval $[t - 1/2, t + 1/2]$. Note that this probability does not depend on μ, the parameter which controls the magnitude of oscillations. The probability of detecting a better point in Algorithm 3 is equal to $1/(4(1 + \mu))$, as already disussed in Section 2.3. The expected number of iterations before detecting the global minimum is therefore $4d$ for Algorithm 2 and $4(1 + \mu)d$ for Algorithm 3. Of course, the cost of an iteration is larger for Algorithm 2

than for Algorithm 3 because in the former a local search is always started at each iteration.

These simple results clearly show what was already remarked in Section 2.3: Algorithm 2 is not influenced by parameter μ, which controls the magnitude of the oscillations, but this parameter does have a strong impact on Algorithm 3. This suggests that local searches should be more often employed when the function displays large oscillations. In some sense, local searches allow us to remove the noise, i.e. the oscillations, which prevent us from reaching the improving region even when we are already close to it. It could be argued that the reason for the failure of Algorithm 3 for large oscillations is that it accepts a new point only when it strictly improves on the current one, whereas in some cases it may be profitable to accept worsening points. This is what is done, for instance, in simulated annealing algorithms.

Simulated annealing algorithms can be efficient when oscillations are not too large, as in Figure 5, but are inefficient when there are large oscillations, as in Figure 6. Indeed, as observed in [9], at a fixed temperature T the value of the density at a local minimum v in stationary conditions is proportional to

$$\exp\left\{-\frac{f(v)}{T}\right\}, \tag{4}$$

while the expected time to climb a barrier of height h which separates two local minima is

$$\exp\left\{\frac{h}{T}\right\}. \tag{5}$$

It follows from (4) that in order to make the probability of being close to a local minimum v_1 much larger than the probability of being close to a neighbouring local minimum v_2, with $f(v_2) > f(v_1)$, we need a value of T which is small compared to the difference $f(v_2) - f(v_1)$. It follows from (5), however, that a value of T which is large with respect to h is needed in order to keep the expected time needed to climb the hill low.

For function g the difference in the function values of two neighbouring local minima is fixed and equal to one, but the height of the barrier increases as μ increases. For large values of μ it becomes impossible to find a temperature T which both strongly favours the better local minimum and keeps the expected time needed to climb the barrier between them small. In Figure 5 (corresponding to a small μ value), for each local minimum the barrier which has to be climbed in order to reach the neighbouring local minimum with a better function value is small compared to the one which has to be climbed in order to reach the neighbouring local minimum with a worse function value. In Figure 6 (corresponding to a large μ value), however, the two barriers have almost the same height and it is not

possible to find a temperature which makes it easy to climb the barrier towards the better local minimum (which is also in the direction of the global minimum) and difficult to climb the barrier towards the worse local minimum (which is in the opposite direction of the global minimum).

3.2. COMPUTATIONAL RESULTS

We now turn to some practical numerical findings. The results in the previous section for Algorithm 2 have been obtained by assuming that the local minimum returned by the local search is the one reached from the starting point along a line of decreasing function values. So, for instance, if the starting point is $z = t + 1/4$ the detected local minimum is $v' = t$. As a consequence of this assumption the results of the algorithm applied to problem (3) are not influenced by the value of μ. It has already been briefly remarked in Section 2.3, however, that real local search routines are sometimes more clever and able to jump over some local minima and end in better ones. For instance, by starting again at $z = t + 1/4$ we may end up with $v' = t - 1$ instead of t. The ability to jump over worse local minima seems to be more evident when the magnitude of oscillations is not too large, as we will now show by applying Algorithm 2 to some test functions.

For the Rastrigin function, Algorithm 2 has been applied with

$$\alpha = 0.5\sqrt{\frac{\ell}{10}}, \tag{6}$$

where ℓ is the edge length of the hypercube X representing the feasible region of the problem ($\ell = 10.24$ for the Rastrigin function). The average results (over 1000 random tests) for $n = 20$ are reported in Table 1. (The local search routine employed is a limited memory BFGS and the number of gradient evaluations is the same as the number of function evaluations.) These results show that, in spite of the fact that the number of local minima is the same for $k = 2$ and $k = 10$ and that the minima are almost in the same positions, the magnitude of oscillations does have some impact on the performance of the algorithm. This is probably because with small oscillations it is easier for the local search routine to jump over some local

Table 1. Average number of local searches and function evaluations needed to detect the global optimum of the Rastrigin function with $n = 20$, for $k = 2$ and $k = 10$

		Local searches	Function evaluations
k	2	97	1094
	10	388	4680

minima and end up in better ones, while this becomes more and more difficult as the magnitude of oscillations is increased.

For the Rastrigin function Algorithm 2, applied with large α values, displays very bad performance. For instance, it was not possible to detect the global minimum of the Rastrigin function with $k = 2$ even after 100,000 local searches. For such values there is a single α-HLLM but a strong inefficiency introduced in the exploration of α-neighborhoods. On the other hand, it has already been observed that these functions are such that a single α-HLLM exists, even for smaller values of α, for which α-neighborhoods can be explored efficiently. The results obtained for them with α given by (6) are a practical confirmation of this observation. We emphasize again that appropriate α values are generally not known in advance and that usually some adaptive scheme must be incorporated in a real algorithm in order to detect them.

We know from Section 2.1 that appropriate α values do not exist for all functions, that is, values for which both the number of α-HLLM is small and the exploration of α-neighborhoods can be done in an efficient way. In particular, in that section we mentioned the case of the Schwefel function. For large values of α a single α-HLLM exists, but if we decrease the value of α the number of α-HLLM immediately increases to 2^n (see Figure 7 for the case $n = 2$). The greater difficulty presented by this function compared to the Rastrigin one is computationally confirmed by the fact that for $n = 20$ not even 1000 runs of Algorithm 2 with different α values were able to return the global minimum.

In Section 2.4 we noticed that although in some cases the number of α-HLLM is very large it may be the case that they are positioned according to some regular pattern. In such cases it may be possible to increase the efficiency of Algorithm 2 by incorporating a step which collects and combines the results of multiple runs of Algorithm 2. The structure of the resulting algorithm is now presented.

ALGORITHM 4.

Step 1. Set $C = \emptyset$.

Step 2. Run Algorithm 2 and let v be the returned point.

Step 3. Set $C = C \cup \{v\}$. Combine in some way (to be specified) the points in C and return a new set of points A. If the stopping rule is satisfied, then stop. Otherwise go back to Step 2.

Discussion of how to combine points in C in order to get a new set of points A goes beyond the scope of this paper, but, for instance, the crossover operations of genetic algorithms are one possibility. Indeed, the Schwefel function has been defined as "the genetic algorithm playground". Excellent computational results are reported in [7], although it should be

pointed out that these results have been obtained by modifying very few variables of current solutions at each iteration (and not all of them as in Step 2 of Algorithm 2), thus explicitly exploiting the separability of the objective function. While the regular pattern of the α-HLLM is obvious for the Schwefel function, in some other cases it is not known whether such a regular pattern exists. For instance, the α-HLLM of the Lennard-Jones functions may follow some regular pattern (the successful application of genetic algorithms to these problems suggests that, see for example [1, 4]) but gaining insight about these patterns appears not a trivial task.

Finally, we notice that while there does not seem to exist in the literature a unique class of test functions in which all the features described in the previous sections are incorporated, most do appear across standard test functions. For instance, parameter k in the Rastrigin function controls the magnitude of the oscillations, and the Rastrigin function is an example of a function for which a single α-HLLM can be obtained even for small α-values. On the other hand, the Schwefel function is an example where this is not possible. What seems to be lacking in the test function literature is a parameter which controls the size of basins of attraction of different α-HLLM. What would be needed is a parameter which produces both the landscape in Figure 1 (with a large basin of attraction for the α-HLLM corresponding to the global minimum) and the landscape in Figure 4 (with a small basin of attraction for the α-HLLM corresponding to the global minimum). A partial answer is provided by Lennard–Jones functions. Although they do not contain a parameter which explicitly controls the size of the basins of attraction, they provide a wide range of cases as N, the number of atoms, varies. The range goes from easy cases (for example, $N = 13$) where a single α-HLLM exists, to very hard cases (for example, $N = 38,75–77,98,102–104$) where the basin of attraction of the α-HLLM corresponding to the global minimum is very narrow (see for example the discussion on multiple-funnel energy landscapes in [2]).

4. Conclusion

Four features of objective functions which strongly influence the ease of global optimization have been isolated, as follows. The first and second involve the concept of α high level local minima (Section 2.1).

1. The existence of a local search parameter α which gives rise to both cheap neighbourhood exploration (favoured by small α) and a small number of α-HLLM (favoured by large α). (If a compromise exists, the problem is easier.)
2. The size of the basin of attraction of the α-HLLM corresponding to the global minimum. (The larger the basin, the easier is the problem.)

3. The amplitude of oscillations. (The higher the amplitude, the greater the need for local search.)
4. Pattern in the locations of the local minima. (The more apparent the pattern, the easier is the problem.)

Some algorithms have been described and the influence of these features on their performance discussed.

To conclude, the problem raised and partially discussed has been ''Given a particular objective function, how should the parameters of a stochastic algorithm be chosen to ensure that it converges as efficiently as possible?''

Acknowledgements

The authors are grateful to Prof. I. Bomze for some very helpful discussions. The authors would like to thank the Marsden Fund of the Royal Society of New Zealand for support of this research.

References

1. Barron, C., Gomez, S. and Romero, D. (1999), The optimal geometry of Lennard–Jones clusters. *Computer Physics Communications* 123, 87–96.
2. Doye, J.P.K. (in press), Physical perspectives on the global optimization of atomic clusters. In: Pinter, J. (ed.), *Global Optimization – Selected Case Studies*.
3. Hansen, P. and Mladenović, N. (2001), Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130, 449–467.
4. Hartke, B. (1999), Global cluster geometry optimization by a phenotype algorithm with niches: Location of elusive minima, and low-order scaling with cluster size. *Journal of Computational Chemistry* 20, 1752.
5. Leary, R.H. and Doye, J.P.K. (1999), New tetrahedral global minimum for the 98-atom Lennard–Jones cluster. *Physical Review. E* 60, R6320–R6322.
6. Leary, R.H. (2000), Global optimization on funneling landscapes. *Journal of Global Optimization* 18, 367–383.
7. Mühlenbein, H. and Schlierkamp-Voosen, D. (1993), Predictive models for the Breeder Genetic Algorithm. *Evolutionary Computation* 1, 25–49.
8. Pardalos, P. and Schnitger, G. (1988), Checking local optimality in constrained quadratic programming is *NP*-hard. *Operations Research Letters* 7, 33–35.
9. Sorkin, G.B. (1991), Efficient simulated annealing on fractal energy landscapes. *Algorithmica* 6, 367–418.