Journal of Global Optimization **7:** 93–110, 1995.

Towards Pure Adaptive Search

A General Framework and a One-Dimensional Realisation

W. P. BARITOMPA and ZHANG BAOPING
Department of Mathematics, University of Canterbury, Christchurch, New Zealand

R. H. MLADINEO
Rider University, Lawrenceville, New Jersey, USA

G. R. WOOD
Department of Mathematics and Computing, Central Queensland University, Rockhampton, Queensland 4702, Australia

and

Z. B. ZABINSKY
Industrial Engineering Program, University of Washington, Seattle, Washington 98195, USA

(Received: 24 August 1992; accepted: 18 April 1995)

Abstract. The algorithm known as Pure Adaptive Search is a global optimisation ideal with desirable complexity. In this paper we temper it to a framework we term Somewhat Adaptive Search. This retains the desirable complexity, but allows scope for a practical realisation. We introduce a new algorithm termed Pure Localisation Search which attempts to reach the practical ideal. For a certain class of one variable functions the gap is bridged.

Key words: Global optimisation, stochastic, random search, localisation, complexity.

1. Introduction

The ideal of Pure Adaptive Search (PAS) has been introduced and discussed in [10] and [16]. Pure Adaptive Search occurs when we are always able to choose the next evaluation point according to a uniform distribution on the improving region, or "inside the level set", of the feasible space. In [16] it was shown that when Pure Adaptive Search is applied to global mathematical programs satisfying a Lipschitz condition, the expected number of iterations to convergence increases at most linearly in the dimension of the problem, a desirable complexity result. Convergence here occurs when the lowest known value is within a given tolerance of the global minimum. A difficulty which immediately arises is that Pure Adaptive Search appears to be hard to realise in practice. Encouragement, however, comes from the observation that several other practical random search algorithms have reported linearity in dimension, for example [13], although only for convex programs.

Pure Adaptive Search can be implemented, albeit very inefficiently, by running Pure Random Search (PRS) and accepting only those points which provide improved function evaluations. Two attempts have already been made to provide a more efficient implementation. These are the Improving Hit-and-Run algorithm [17], and the Hide-and-Seek algorithm [4]. The purpose of this paper is to approach the problem from a third perspective, which we now describe.

The central idea is to focus on an enlargement of the level set of Pure Adaptive Search. A delicate balancing act is required. The enlargement must be accessible in practice, yet small enough to retain the properties of PAS. Such an enlargement is provided by the deterministic algorithms for mathematical programs satisfying the Lipschitz condition found in [11, 12, 9, 15]. These have the property that, at each iteration, regions which cannot contain the global minima are stripped away from the domain. A general framework for such algorithms is described in [2]. Basso in [3] uses the word "localisation" for the resulting enlargement of the level set which is known to contain the global minimisers. Both [9] and [15] reduce to the well-known Piyavskii–Shubert algorithm, [11] and [12], for functions of a single variable. The localisations they provide reach towards the level set of PAS.

Is there an efficiently implementable algorithm, based on a stochastic variant of the Piyavskii–Shubert algorithm, which can realise the desirable complexity of PAS? This paper initiates a study of this question.

We begin by modifying PAS to an algorithm which we term Somewhat Adaptive Search (SAS). Somewhat Adaptive Search is a relaxation of Pure Adaptive Search, and is more likely to be efficiently implementable, yet still possesses the desirable complexity of PAS (Theorem 2.1 and Corollary 2.1). In the same breath, we modify the Piyavskii–Shubert style of algorithm to a stochastic search we term Pure Localisation Search (PLS). This algorithm chooses uniformly from the localisation. The convergence properties of PLS lie between PRS and PAS (Theorem 3.2 and Corollary 3.1). We conclude by showing that for a class of functions of a single variable, PLS does realise an SAS algorithm. This is shown in the last set of results (Theorem 4.1 and Corollary 4.1).

The layout of the paper is as follows. In Section 2 we introduce Somewhat Adaptive Search, prove the linear complexity result, and discuss the special case of "ρ-adaptive" search. In Section 3 we define Pure Localisation Search and introduce spherical and simplicial realisations of this algorithm. The link between SAS and PLS is made in Section 4, where we show, for a limited class of functions, that the above realisations of PLS are SAS. Numerical results which confirm the theoretical results, and indicate directions for future research, are given in Section 5.

2. Somewhat Adaptive Search (SAS)

Throughout we consider the global optimization problem

$$\text{(P)} \qquad \min_{x \in S} f(x)$$

where f is a real-valued function on a convex, compact full-dimensional subset S of $\mathbf{R}^n$. We denote the optimal solution by (x_*, y_*), where $x_* \in \arg\min_{x \in S} f(x)$ and $y_* = f(x_*)$. It is convenient to define $y^* = \max_{x \in S} f(x)$. We do not require that a unique minimum point should exist. If there is more than one, we choose x_* arbitrarily.

We will consider stochastic sequential search procedures whose aim is to locate (x_*, y_*). We use a number of random variables and denote them using upper case letters, following standard statistical practice. The sample path of evaluation points we denote $X_1, X_2, \ldots$ and the associated function values $Y_1, Y_2, \ldots$. Epoch $i > 1$ is said to be a *record* of the sequence $\{Y_k\}$ if $Y_i < \min\{Y_1, \ldots, Y_{i-1}\}$ or $Y_i = y_*$. For technical reasons it is convenient to include the latter condition. Epoch $i = 1$ is always considered to be a record. The corresponding value Y_i is called a *record value*. For $k \geq 1$ we let $R(k)$ denote the epoch of the kth record value of the sequence of evaluations. For $k \geq 2$, the number of iterations from the $(k-1)$th to the kth record is denoted by I_k. Thus $I_k = R(k) - R(k-1)$.

At times we refer to a specific realisation of a random variable by using corresponding lower case letters. For example $x_1, x_2, \ldots$, a realisation of $X_1, X_2, \ldots$, will denote a particular list, or sample path, of evaluation points in a given search procedure. Note that the random variable I_k is a function of such a sample path, whence $I_k(\omega)$ refers to its value at the sample path denoted by ω.

The expected value of a random variable V is denoted by $E(V)$. Two sequences of random variables are said to be stochastically equivalent if they have equal joint distributions.

We pause to recall the definition of the PAS algorithm for solving problem (P):

DEFINITION 2.1. Pure Adaptive Search (PAS)

Initial Step: Set $k = 0$ and $S_0 = S$.

Iterative Step: Increment k
(i) Select evaluation point.
Choose x_k uniformly on S_{k-1} if non-empty,
else set $x_k = x_{k-1}$
Set $w_k = f(x_k)$.
(ii) Update the improving region.
Set $S_k = f^{-1}(-\infty, w_k)$.

Stopping Criterion: Stop if a stopping criterion is met, else return to the iterative step.

Mindful of the virtues of PAS, but aware of the impossibility of achieving it in practice, we now define a new class of algorithms. This is an attempt to keep these virtues while at the same time allowing room to construct practical algorithms. The algorithms require that two conditions should hold. The first allows the algorithm

to mark time between records, but not for too long, while the second insists that the quality of the records be as good as those of PAS. The first condition gives the space needed to implement the algorithm, while together they ensure that the "linearity in dimension" drawcard of PAS is retained.

DEFINITION 2.2. A stochastic sequential search algorithm for solving (P) is termed Somewhat Adaptive Search (SAS) if the following two conditions are satisfied:

(i) There exists a bound $\beta \geq 1$ such that $E(I_k) \leq \beta$ for all k, and

(ii) $\{Y_{R(k)} : k = 1, 2, \ldots\}$ is stochastically equivalent to $\{W_k : k = 1, 2, \ldots\}$, the sequence of record values of PAS.

In [16], for a non-constant function, Zabinsky and Smith define the *relative improvement* associated with an evaluation $y > y_*$ as $z = (y^* - y)/(y - y_*)$. In the context of this paper "relative value" would be a better name. We choose to retain, however, the original authors' terminology. In order to state our main theorem, we extend the language of [16] to:

$N_{SAS}(z)$ = the number of iterations of SAS achieving a relative improvement of z or less

$N^*_{SA}(y)$ = the number of iterations of SAS required to achieve a value of y or lower.

The wording of the above must be heeded carefully. The first expression is the number of iterations *achieving* something, while the second is the number *required* before something is achieved. The following relation is easy to show and worth noting:

$$E(N^*_{SAS}(y)) = 1 + E(N_{SAS}(z))$$

when $z = (y^* - y)/(y - y_*)$.

The corresponding expressions for PAS we denote by $N_{PAS}(z)$ and $N^*_{PAS}(y)$.

THEOREM 2.1. *For SAS applied to problem (P), we have*

$$E\left[N^*_{SAS}(y)\right] \quad \leq \quad \beta\, E\left[N^*_{PAS}(y)\right].$$

COROLLARY 2.1. *Consider all global optimisation problems (P) over a convex feasible region in R^n with diameter at most d, and all functions Lipschitz with parameter at most M. Suppose an algorithm is SAS for this class of problems, and the bound in Definition 2.2 (i) is at most B. Then*

$$E\left[N^*_{SAS}(y)\right] \quad \leq \quad B + \left[B \ln(Md/(y - y_*))\right] n.$$

That is, the bound is a linear function of the dimension n of the problem.

Proof of Theorem. For a constant function some of the above terms are undefined. In this case it is easy to check that SAS is actually PAS and the theorem follows trivially. For a non-constant function, since the $Y_{R(k)}$ are stochastically equivalent to the W_k it follows that

$$N_{SAS}(z) = I_1 + I_2 + \ldots + I_{N_{PAS}(z)}.$$

$$\begin{aligned} \text{Then} \quad E\left[N_{SAS}(z)\right] &= E\left\{E\left[I_1 + \ldots + I_{N_{PAS}(z)} \middle| N_{PAS}(z) \text{ fixed}\right]\right\} \\ &\leq E\left[\beta N_{PAS}(z)\right] \\ &= \beta E\left[N_{PAS}(z)\right]. \end{aligned}$$

Converting this into a result about N^*_{SAS}, we see

$$\begin{aligned} E\left[N^*_{SAS}(y)\right] &\leq E\left[N^*_{SAS}(y)\right] - 1 + \beta \\ &= E\left[N_{SAS}\left(\frac{y^* - y}{y - y_*}\right)\right] + \beta \\ &\leq \beta\, E\left[N_{PAS}\left(\frac{y^* - y}{y - y_*}\right)\right] + \beta \\ &= \beta\, E\left[N^*_{PAS}(y)\right], \quad \text{as required.} \end{aligned}$$

□

Proof of Corollary. When the conditions of Corollary 2.1 hold, it was shown in [16, Theorem 5.3] that $E[N^*_{PAS}(y)] \leq 1 + [\ln(Md/(y - y_*))]n$. Coupled with the result of Theorem 2.1, this gives the statement in the corollary. □

We pause to discuss a special case of SAS, namely "ρ-adaptive search". Informally, a search is ρ-adaptive, for some ρ, with $0 \leq \rho \leq 1$, if at each iteration the probability that it behaves as PAS is at least ρ. The letter ρ is chosen as a reminder of "record".

DEFINITION 2.3. Let $0 \leq \rho \leq 1$. A stochastic sequential search for solving (P) is termed *ρ-adaptive* if for each iteration k, and for all sample paths $x_1, \ldots, x_{k-1}$,

$$P\left[X_k \text{ is distributed uniformly in the improving region} \mid x_1, \ldots, x_{k-1}\right] \geq \rho$$

In this framework, PRS becomes 0-adaptive, and PAS 1-adaptive. This language gives us a way of describing a spectrum of algorithms between these two extremes. Our next result shows that a ρ-adaptive algorithm, with $\rho > 0$, is always SAS.

THEOREM 2.2. *A ρ-adaptive algorithm, with $0 < \rho \leq 1$, is SAS, with $\beta = 1/\rho$.*

Proof. The definition of ρ-adaptivity ensures at any iteration k, and independent of the sample path, the probability of a record is greater than or equal to ρ. Thus $E(I_k)$ is less than or equal to the mean of a geometric distribution, with parameter ρ,

or $E(I_k) \leq 1/\rho$, for all k. Condition ii), that the $Y_{R(k)}$ are stochastically equivalent to the RW_k, follows via a straighforward modification of [16, Lemma 3.1]. □

This completes the setting up of the attainable ideal.

3. Pure Localisation Search (PLS)

We turn our attention now to a new and readily implemented algorithm for solving problem (P). In Theorem 3.1 we show that the records it produces are stochastically equivalent to those of PAS, while in Corollary 4.1 we show that it realises a SAS algorithm on a particular class of functions of a single variable. In spirit, the algorithm is a probabilistic analogue of the well-known Piyavskii–Shubert algorithm and its higher dimensional extensions, for example [9, 15]. We present the algorithm in a general setting initially.

The central idea is the following. An exact tracking of the level set of PAS is an impossible task. Tracking a superset of it is not. Certain "removal" algorithms in the literature (for example [11, 12, 9, 15]), while deterministic, do yield a localisation for the level set at each iteration.

DEFINITION 3.1. Pure Localisation Search (PLS)

Initial Step: Set $k = 0$ and $L_0 = S$.
Set $\alpha_0 = \infty$.

Iterative Step: Increment k
(i) Select evaluation point.
Choose x_k uniformly on L_{k-1} if non-empty,
else set $x_k = x_{k-1}$.
Set $y_k = f(x_k)$.
(ii) Update localisation.
Set $\alpha_k = \begin{cases} y_k, & \text{if } y_k < \alpha_{k-1} \\ \alpha_{k-1}, & \text{otherwise.} \end{cases}$
Set $L_k = L_{k-1} - R_k$, where the removal region R_k is such that $R_k \subseteq S - f^{-1}(-\infty, \alpha_k)$.

Stopping Criterion: Stop if a stopping criterion is met, else return to the iterative step.

We define $S_y = f^{-1}(-\infty, y)$. Observe that the special case of PLS with $R_k = \phi$ is PRS, while PLS becomes PAS when $R_k = S - S_{\alpha_k}$, so $L_k = S_{\alpha_k}$. It follows readily from Theorem 3.2 of this section that PLS converges with probability one.

An important observation concerning any PLS is that $L_k \supseteq S_{\alpha_k}$, that is, the localisation always contains the improving set.

That PLS always has the second property of SAS is shown in the next theorem.

THEOREM 3.1. *For PLS applied to the global optimisation problem (P), the stochastic process of PLS record values is equal in distribution to the stochastic process of PAS record values. That is*

$$\{Y_{R(k)} : k = 1, 2, \ldots\} \quad \sim \quad \{W_k : k = 1, 2, \ldots\}.$$

Proof. The proof is an extension of [16, Lemma 3.1]. First we show that the conditional distributions are equal. Let k be any iteration and take $y_* \leq y < y' \leq y^*$. Note that

$$P[W_{k+1} < y | W_k = y'] = P[Y_{R(k)+j} < y | Y_{R(k)} = y' \text{ and } Y_{R(k)+1} \geq y' \ldots Y_{R(k)+j-1} \geq y' \text{ and } Y_{R(k)+j} < y']$$

since both sides equal $\lambda(S_y)/\lambda(S_{y'})$. Here λ denotes Lebesgue measure on $\mathbf{R}^n$. Then we have

$$\begin{aligned}
&P[Y_{R(k+1)} < y | Y_{R(k)} = y'] \\
&= P[Y_{R(k)+1} < y | Y_{R(k)} = y'] + P[Y_{R(k)+1} \geq y' \text{ and } \\
&\quad Y_{R(k)+2} < y | Y_{R(k)} = y'] + \cdots \\
&= P[Y_{R(k)+1} < y' | Y_{R(k)} = y'] \cdot P[Y_{R(k)+1} < y | Y_{R(k)} = y' \text{ and } Y_{R(k)+1} < y'] \\
&\quad + P[Y_{R(k)+1} \geq y' | Y_{R(k)} = y'] \cdot P[Y_{R(k)+2} < y' | Y_{R(k)} = y' \text{ and } Y_{R(k)+1} \geq y'] \cdot \\
&\quad P[Y_{R(k)+2} < y | Y_{R(k)} = y' \text{ and } Y_{R(k)+1} \geq y' \text{ and } Y_{R(k)+2} < y'] + \cdots \\
&= [p_1 + (1 - p_1)p_2 + (1 - p_1)(1 - p_2)p_3 + \cdots] \, P[W_{k+1} < y | W_k = y']
\end{aligned}$$

where $p_i = P[Y_{R(k)+i} < y' \mid Y_{R(k)} = y' \text{ and } Y_{R(k)}, \ldots, Y_{R(k)+i-1} \geq y']$, the probability that the first record after the kth record occurs at epoch $R(k) + i$. Here, from the key feature of PLS, we have $S_{y'} \subset L_{R(k)+i-1}$ so for all i,

$$p_i = \lambda(S_{y'})/\lambda(L_{R(k)+i-1}) \geq \lambda(S_{y'})/\lambda(S).$$

So the sequence $\{p_i\}$ is bounded away from zero, and by an elementary argument $p_1 + (1 - p_1)p_2 + \cdots = 1$. Thus

$$P[Y_{R(k+1)} < y | Y_{R(k)} = y'] \quad = \quad P[W_{k+1} < y | W_k = y'].$$

We now use induction to show that the unconditional distributions are equal. By convention, $R(1) = 1$ and from the definition of PLS, $P[Y_1 < y] = P[W_1 < y]$ for all y, for $y_* \leq y \leq y^*$. Hence $Y_{R(1)} \sim W_1$.

Now consider $k > 1$ and suppose that $Y_{R(i)} \sim W_i$ for $i = 1, 2, \ldots, k$. Then, for all $y_* \leq y \leq y^*$, we have

$$\begin{aligned}
P[Y_{R(k+1)} < y] &= \int_{y_*}^{y^*} P[Y_{R(k+1)} < y | Y_{R(k)} = t] \mathrm{d}F_{Y_{R(k)}}(t) \\
&= \int_{y_*}^{y^*} P[W_{k+1} < y | W_k = t] \mathrm{d}F_{W_k}(t) \\
&= P[W_{k+1} < y]
\end{aligned}$$

The second equality follows using the equality of conditional distributions and the induction hypothesis. By induction it follows that the two sequences are equal in marginal distribution, hence in joint distribution, as required. □

The way in which PLS is sandwiched between PRS and PAS is made clear in the next theorem and its immediate corollary.

THEOREM 3.2. *Fix f as in problem (P), and fix a relative improvement level, $z > 0$. Let $N_{PAS}(z)$, $N_{PLS}(z)$ and $N_{PRS}(z)$ be the number of iterations of PAS, PLS and PRS respectively, achieving a relative improvement of z or less. Then*

$$P[N_{PAS}(z) < k] \quad \geq \quad P[N_{PLS}(z) < k] \quad \geq \quad P[N_{PRS}(z) < k].$$

COROLLARY 3.1.

(i) $E[N_{PAS}(z)] \leq E[N_{PLS}(z)] \leq E[N_{PRS}(z)]$

(ii) $N^p_{PAS}(z) \leq N^p_{PLS}(z) \leq N^p_{PRS}(z)$ *where* $N^p_{alg}(z)$ *is the number of iterations of the algorithm required to achieve a relative improvement of z with probability not less than p.*

Proof of Theorem. Let y correspond to the relative improvement of z. Then

$$\begin{aligned} P[N_{PAS}(z) < k] &= P[W_k \leq y] \\ &= P[Y_{R(k)} \leq y], \quad \text{by Theorem 3.1} \\ &\geq P[N_{PLS}(z) < k] \end{aligned}$$

since if PLS achieves a relative improvement of z before the kth iteration then $Y_k \leq y$, whence $Y_{R(k)} \leq y$.

In order to show that $P[N_{PLS}(z) < k] \geq P[N_{PRS}(z) < k]$ we now show that $P[N_{PLS}(z) > k-1] \leq P[N_{PRS}(z) > k-1]$. Note that $i = N_{PLS}(z) + 1$ is the first epoch such that $x_i \in S_y$. Thus, for any k, $k = 1, 2, \ldots,$

$$\begin{aligned} P[N_{PLS}(z) > k-1] &= P[x_1 \notin S_y \text{ and } x_2 \notin S_y \ldots \text{ and } x_k \notin S_y] \\ &= P[x_1 \notin S_y]P[x_2 \notin S_y | x_1 \notin S_y] \ldots \\ &\quad P[x_k \notin S_y | x_1, \ldots, x_{k-1} \notin S_y] \end{aligned}$$

Consider a PLS sample path with first $j-1$ domain points not in S_y, where $j \in \{1, \ldots, k\}$. Then $\alpha_{j-1} \geq y$, so $S_{\alpha_{j-1}} \supseteq S_y$, or $L_{j-1} \supseteq S_y$. It follows that

$$P[x_j \in S_y | x_1, \ldots, x_{j-1} \notin S_y] \quad = \quad \lambda(S_y)/\lambda(L_{j-1}) \quad \geq \quad \lambda(S_y)/\lambda(S).$$

Hence,

$$P[x_j \in S_y | x_1, \ldots, x_{j-1} \notin S_y] \quad \geq \quad \lambda(S_y)/\lambda(S)$$

as it is the average of the above term over all initial segments $x_1, \ldots, x_{j-1}$ of PLS sample paths with domain points not in S_y. Thus

$$P[N_{PLS}(z) > k-1] \quad \leq \quad [1-\lambda(S_y)/\lambda(S)]^k \quad = \quad P[N_{PRS}(z) > k-1]. \quad \square$$

4. Linking the Ideal of SAS to the Reality of PLS

Earlier we noted that PRS is a trivial PLS but clearly not SAS. Is it possible to find a natural PLS algorithm and a class of functions which achieves SAS? This section is devoted to showing that the answer is, surprisingly, yes.

Natural PLS algorithms for Lipschitz continuous functions

When f is Lipschitz continuous with Lipschitz bound M, two realisations of PLS immediately arise. The first is a stochastic analogue of the method in [9], the second a stochastic analogue of the method in [15]. In the former,

$$R_k = \bigcup_{i=1,\ldots,k} B_i$$

where B_i is the ball of radius $(y_i - \alpha_k)/M$, centred at x_i. In the latter,

$$R_k = \bigcup_{i=1,\ldots,k} C_i$$

where C_i is a standard simplex of radius $(y_i - \alpha_k)/M$, centred at x_i. We call the former *Spherical* PLS, and the latter *Simplicial* PLS. Note that when $n = 1$ these two realisations of PLS reduce to the same algorithm.

In [15] the term "bracket" was used to describe the $n + 1$ dimensional region known to contain the points on the graph at the global minima. This bracket can be used to implement PLS. The projection of this bracket onto the domain is what we term the localisation. When $n = 1$, both Spherical and Simplicial PLS yield a bracket composed of disjoint similar triangles. For higher dimensions Simplicial PLS produces a union of overlapping, but similar, simplexes. For Spherical PLS the bracket is more complicated to describe.

We call both the above realisations, LPLS for "Lipschitz PLS". Note that they are non-trivial to implement. When $n = 1$ it is necessary to store a linked list of the intervals which comprise L_k. Selecting the next evaluation point is performed by choosing a random number in [0,1] and moving through the intervals to the x_k value. Updating the localisation involves an updating of the linked list. For $n > 1$, Spherical PLS has the virtue of producing a tighter localisation than Simplicial PLS, since the removed ball always contains the removed simplex. On the other hand, choosing x_k in Spherical PLS has so far been achieved through an acceptance-rejection approach, whereas with Simplicial PLS a linked list of simplex tops can be stored, and a procedure similar to the $n = 1$ case used to find x_k, see [14, 15].

A simple class of functions of one variable

We begin by defining a function of a single variable which we call, for obvious reasons, the (upside down) "witch's hat". For $h \in [0, 1]$ we define the witch's hat of height h to be $w_h(x) = \min(|x|, h)$, for $x \in [-1, 1]$. Note that w_h is Lipschitz continuous, with Lipschitz constant $M = 1$. It will be convenient to call the graph of w_h on $[-1, -h] \cup [h, 1]$ the "brim", and w_h on $[-h, h]$ the "cap". We now define the class of functions, $\mathcal{C}_h$, as all those one variable Lipschitz continuous functions with $M = 1$ which agree with w_h on $[-h, h]$ and elsewhere on $[-1, 1]$ lie above w_h.

When LPLS with $M = 1$ is applied to a function in $\mathcal{C}_h$, the localisation eventually becomes the best possible, the inverse image of all values less than or equal to the best known, that is, $f^{-1}(-\infty, \alpha_k]$. Note that for functions in C_h, when $\alpha_k \leq h$, this is just the interval $[-\alpha_k, \alpha_k]$. This is formalised in the following theorem.

THEOREM 4.1. *Let f be any function in $\mathcal{C}_h$, and for LPLS with $M = 1$, let N be the number of iterations until the localisation becomes the level set. Then*

$$E[N] < 6 + 26/h.$$

Before presenting the proof, we proceed to the consequence:

COROLLARY 4.1. *For functions in $\mathcal{C}_h$, LPLS with $M = 1$ is SAS, with $\beta = 6 + 26/h$.*

Proof. Partition the sample paths as $\cup_{i=1}^{\infty} \Omega_i$, where Ω_i is the set of all sample paths for which the localisation becomes the level set at the *ith* iteration. Then for each $\omega \in \Omega_i$, *a fortiori* $I_k(\omega) \leq i$ for all k, since for such sample paths there are at most i iterations from any one record to the next. Thus

$$E(I_k) \quad \leq \quad \sum_{i=1}^{\infty} P[\Omega_i]\, i \quad = \quad E[N]$$

for any k. It follows that condition (i) of SAS is satisfied with $\beta = 6 + 26/h$, using Theorem 4.1. Condition (ii) of SAS follows at once from Theorem 3.1. □

Proof of Theorem 4.1. Take $f \in \mathcal{C}_h$. A typical situation which would arise when running LPLS on f, once an evaluation is found less than h, is shown in Figure 1.

The localisation, L_k, consists of four parts:

1. The level set, A_k.
2. A finite union of intervals, B_k, under the brim.
3. One interval, C_k^l, to the left of A_k and under the cap.
4. One interval, C_k^r, to the right of A_k and under the cap.

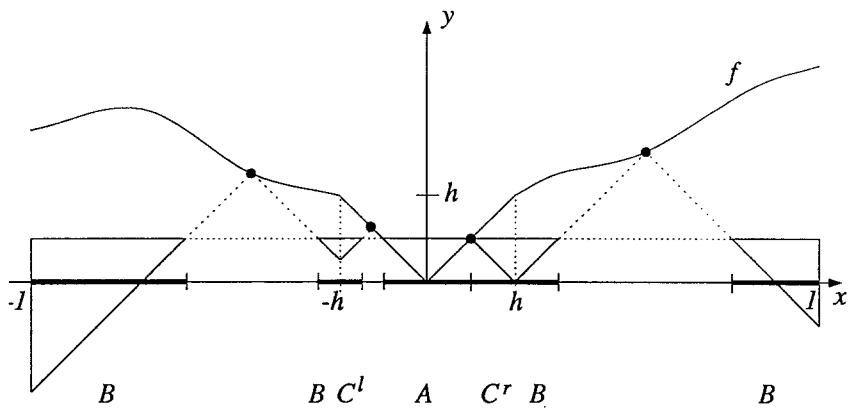

Fig. 1. Running LPLS on $f \in C_h$: the four parts A, $B = B_1 \cup B_2 \cup B_3 \cup B_4$, C^l and C^r of the localization are shown. The situation illustrated is cap separated.

These sets are indicated in Figure 1. Denote the total length of these sets by a,b,c^l, and c^r respectively. The maximum vertical extent of part of the bracket is called its *depth*. We say at any stage that the run is *cap separated* if there has been an evaluation under the cap both to the left and to the right of the origin.

Four facts are needed in the final proof of the theorem. We present these now. The first three are readily shown; we give a proof for the fourth.

FACT 1. *If the run at the kth iteration is cap separated, then the depth of the bracket over the complement of A_k is less than or equal to $b + c^l + c^r$.*

FACT 2. *Denote by d the depth of the bracket over the complement of A_k. Then $P[B_{k+1} \cup C^l_{k+1} \cup C^r_{k+1} = \phi \mid x_1, \ldots, x_k$, and that $x_{k+1} \in A_k] \geq (a - 2d)/a$ where a and d are the values after the kth iteration.*

FACT 3. $P[C^l_{k+1} = \phi \mid x_1, \ldots, x_k$, and that $x_{k+1} \in C^l_k] \geq 1/2$.

FACT 4. *Consider $t \in (0, h)$. If α_k, the lowest evaluation immediately after the kth iteration, is less than t, then the number of further iterations under the brim is less than or equal to $2\left\lceil \frac{1-h}{h-t} \right\rceil$, where $\lceil x \rceil$ is the least integer greater than or equal to x.*

Proof of Fact 4. Suppose that after k iterations we have $\alpha_k < t < h$. Let $z_1, z_2, \ldots$ be the later iterations of LPLS which are in [h,1]. If $i < j$ then $z_j \notin \overline{B}_{h-t}(z_i)$, the closed interval of radius $h - t$ centred at z_i, so $\{\overline{B}_{(h-t)/2}(z_i) : i = 1, 2, \ldots\}$ is a mutually disjoint collection of closed intervals whose union is a subset of $[h - (h-t)/2, 1 + (h-t)/2]$. It follows that the collection must be finite, having say m elements, and furthermore, that $m(h - t) < 1 - t$. Thus m is less than or equal to the biggest integer less than $(1-t)/(h-t) = (1-h)/(h-t) + 1$.

This is $\left\lceil \frac{1-h}{h-t} \right\rceil$. Fact 4 then follows by doubling this figure. □

The heart of the proof of the theorem rests in recognizing that if we count N_t, the number of iterations until the lowest known evaluation is less than t, and also the number of subsequent iterations, N_p, until we can be sure that the localisation is the level set, then $N_t + N_p$ is greater than or equal to N.

Following the iteration N_t at which $\alpha_{N_t} < t$, we define five types of "progress" event which can occur. These are:

"P_1" Cap separation occurs for the first time at the $(k+1)th$ iteration.

"P_2" $C_k^l \neq \phi$ and $C_{k+1}^l = \phi$

"P_3" $C_k^r \neq \phi$ and $C_{k+1}^r = \phi$

"P_4" $x_{k+1} \in B_k$

"P_5" $B_{k+1} \cup C_{k+1}^l \cup C_{k+1}^r = \phi$

Informally, a progress step is a movement towards the localisation becoming the level set, progress step five. Note that steps one, two and three can occur only once, while step four can occur at most $2\lceil (1-h)/(h-t) \rceil$ times. Thus, once there has been $2\lceil (1-h)/(h-t) \rceil + 3$ progress steps following iteration N_t, the localisation must equal the level set. If we let

$$\begin{aligned} N_t &= \text{the number of iterations, } k, \text{ until } \alpha_k < t, \quad \text{and} \\ N_p &= \text{the number of iterations following the } N_t^{th} \text{ iteration} \\ &\quad \text{to achieve } 2\lceil (1-h)/(h-t) \rceil + 3 \text{ progress steps,} \end{aligned}$$

we have $N \leq N_t + N_p$, so $E[N] \leq E[N_t] + E[N_p]$.

Certainly $E[N_t]$ is smaller for LPLS than PRS on f. For PRS on f, the distribution of the number of iterations until a value less than or equal to t is geometric, with probability t. Thus $E[N_t] \leq 1/t$.

We conclude the proof by showing that once we have $\alpha_{N_t} < t$, then the probability of a progress step is always at least 1/6. The distribution of N_p is negative binomial, so $E[N_p] \leq 6(2\lceil (1-h)/(h-t) \rceil + 3)$, whence

$$E[N] \leq \frac{1}{t} + 6\left(2\left\lceil \frac{1-h}{h-t} \right\rceil + 3\right) = 18 + \frac{1}{t} + 12\left\lceil \frac{1-h}{h-t} \right\rceil.$$

Putting $t = h/2$ and removing the "least integer greater than or equal" symbols demonstrates the statement in the theorem.

In order to show that the probability of progress is always greater than or equal to 1/6, we consider three cases. We suppose we have an initial segment of $x_1, \ldots, x_k$, and $\alpha_k \leq t$.

Case 1. The bracket is not cap separated. Then

$$
\begin{aligned}
&P[\text{progress at } (k+1)st \text{ iteration}] \\
&\geq P[P_1] + P[x_{k+1} \in C_k^l \text{ and } P_2] + P[x_{k+1} \in C_k^r \text{ and } P_3] + P[P_4] \\
&\geq \frac{a/2}{a + c^l + c^r + b} + \frac{c^l/2}{a + c^l + c^r + b} + \frac{c^r/2}{a + c^l + c^r + b} \\
&\quad + \frac{b}{a + c^l + c^r + b} \geq \frac{1}{2}.
\end{aligned}
$$

Case 2. The bracket is cap separated, and $\dfrac{a}{a + c^l + c^r + b} \geq \dfrac{2}{3}$. Then

$$
\begin{aligned}
&P[\text{progress at } (k+1)st \text{ iteration}] \\
&\geq P[x_{k+1} \in A_k \text{ and } P_5] + P[x_{k+1} \in C_k^l \text{ and } P_2] + P[x_{k+1} \in C_k^r \text{ and } P_3] \\
&\quad + P[P_4] \\
&\geq \frac{a - 2(c^l + c^r + b)}{a + c^l + c^r + b} + \frac{c^l/2}{a + c^l + c^r + b} + \frac{c^r/2}{a + c^l + c^r + b} \\
&\quad + \frac{b}{a + c^l + c^r + b} \\
&= \frac{a - 3/2(c^l + c^r + b) + b/2}{a + c^l + c^r + b} \\
&\geq \frac{a}{a + c^l + c^r + b} - \frac{3}{2}\frac{c^l + c^r + b}{a + c^l + c^r + b} \\
&\geq \frac{2}{3} - \frac{3}{2}\cdot\frac{1}{3} = \frac{1}{6}.
\end{aligned}
$$

Case 3. The bracket is cap separated, and $\dfrac{a}{a + c^l + c^r + b} < \dfrac{2}{3}$. Then

$$
\begin{aligned}
&P[\text{progress at } (k+1)st \text{ iteration}] \\
&\geq P[x_{k+1} \in C_k^l \text{ and } P_2] + P[x_{k+1} \in C_k^r \text{ and } P_3] + P[P_4] \\
&\geq \frac{c^l/2}{a + c^l + c^r + b} + \frac{c^r/2}{a + c^l + c^r + b} + \frac{b}{a + c^l + c^r + b} \\
&\geq \frac{1}{2}\cdot\frac{c^l + c^r + b}{a + c^l + c^r + b} \\
&\geq \frac{1}{2}\cdot\frac{1}{3} = \frac{1}{6}.
\end{aligned}
$$

□

5. Numerical Results

This section gives empirical support to results in the previous sections. Although the emphasis of this paper is on showing LPLS is an effectively implementable stochastic variant of the Piyavskii–Shubert algorithm which is SAS in a particular one-dimensional setting, we begin with a comparison of the various algorithms with the Piyavskii–Shubert algorithm. The second set of numerical tests show the bounds proved in Theorem 4.1 and Corollary 4.1 are conservative. The last tests show that for the higher dimensional analogue of the witch's hat, Simplicial PLS keeps some of the similarities from the one dimensional case. Evidence, however, suggests that even in this simple case the number of function evaluations to convergence is not linear with dimension.

Comparison of PAS, LPLS, PRS and the Piyavskii–Shubert algorithm in dimension one

Generally speaking, compared with LPLS and PRS, Piyavskii–Shubert usually takes fewer function evaluations. For functions with a large number of nearly equal global minima, however, LPLS can on average require less work than the Piyavskii–Shubert algorithm to attain modest accuracy. Theorem 3.2 is empirically verified. It is interesting to note that the work required by the Piyavskii–Shubert algorithm and its stochastic variant, LPLS, is very similar to that required by the theoretical PAS.

Two random selections of functions were made. We obtain the number of iterations until the global minimum is found to a specified tolerance, using the various algorithms.

The first selection consisted of 69 Lipschitz continuous functions with M=1 that usually had a small number of local minima, generally one. These were produced by an obvious modification of a procedure due to Graf *et al.* described in [6, pp.240–241]. Figure 2 shows that for this class the algorithms ranked from best to worst are PAS, Piyavskii–Shubert, LPLS and PRS. In this and the following example, for each function and each of PAS, LPLS and PRS, 100 runs were averaged.

In [5] Chuyan and Sukharev showed, under a mild condition, that results from adaptive stochastic global optimisation algorithms are in a certain sense no better than results from non-adaptive stochastic global optimisation algorithms. At first glance this appears to contradict the histograms shown for LPLS and PRS in Figure 2, since it is evident that the average behaviour of LPLS is far better than that for PRS. Chuyan and Sukharev, however, prove the equality of a "best worst-case" performance measure, for adaptive and non-adaptive algorithms. Figure 2 compares average behaviour, with objective functions drawn randomly using the Graf generation process. A much earlier paper, with results of the Chuyan and Sukharev type, dealing with deterministic algorithms and Lipschitz continuous functions, is that of Archetti and Betro [1].

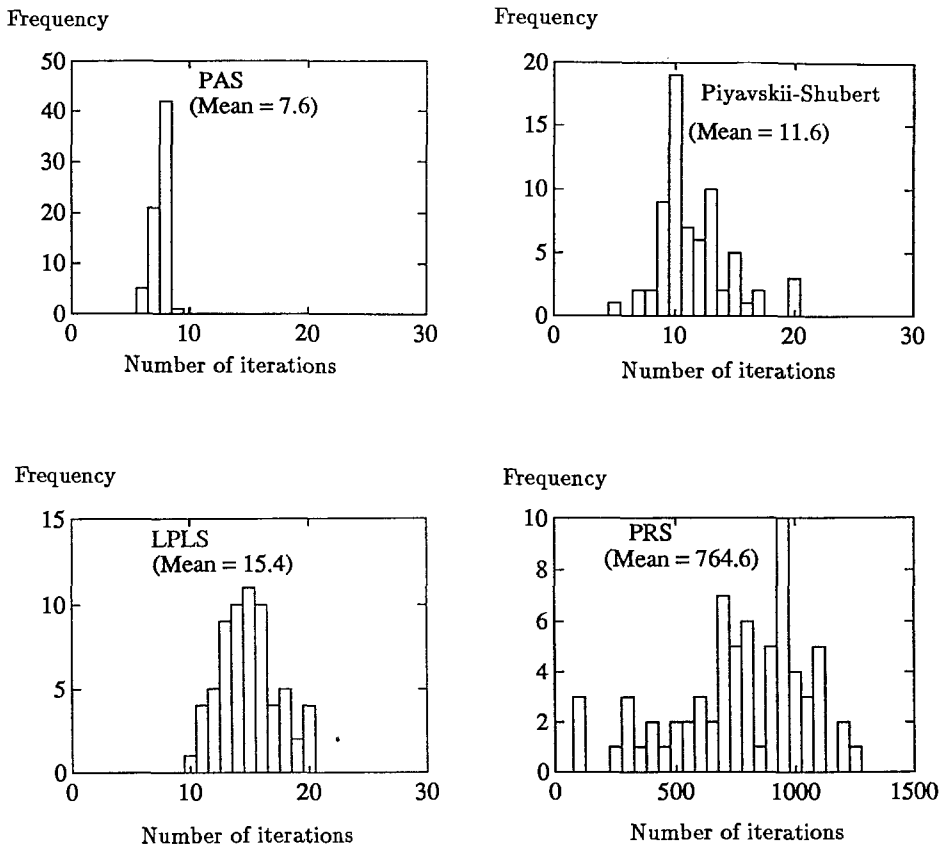

Fig. 2. Histograms showing the mean number of iterations to convergence, with $\epsilon = 0.0005$, for the 69 Graf-generated functions and the four algorithms. Note the different horizontal scale for PRS.

The second selection consisted of 50 Lipschitz continuous functions with M=1, of the form $(1/A)\sin(Ax+B)$ where A and B were randomly chosen. In this selection all of the functions have between one and eight global minima. Table I shows that for this class the algorithms ranked from best to worst are PAS, LPLS, Piyavskii–Shubert, and PRS for modest accuracy, but PAS, Piyavskii–Shubert, LPLS and PRS when greater accuracy is required.

TABLE I. Mean number of evaluations to convergence, with the two relative accuracy levels, for the 50 sinusoidal functions

ϵ	PAS	Piyavskii–Shubert	LPLS	PRS
$0.1/A$	3.0	5.6	5.5	7.6
$0.01/A$	4.2	9.6	11.2	23.9

OBSERVED AND THEORETICAL RESULTS FOR LPLS ON WITCH'S HATS IN DIMENSION ONE

Theorem 4.1 gave a theoretical upper bound on the average number of iterations until the localization becomes the level set for the witch's hat. Table II compares this with the average observed number of iterations, over 1000 runs, thus showing that the theoretical bound is roughly ten times too large. The observed values of $E[N]$ are conservative estimates for β, as demonstrated in Corollary 4.1. Empirical tests have shown that β is roughly one third of $E[N]$.

TABLE II. A comparison of the observed and the theoretical average number of iterations until the localisation becomes the level set, for the witch's hat with varying values of h

h	Observed	Theoretical
1	4.8	32
1/2	7.4	58
1/3	9.8	84
1/4	12.1	110
1/8	21.4	214

OBSERVED RESULTS FOR SIMPLICIAL PLS ON WITCH'S HAT IN HIGHER DIMENSIONS

One higher dimensional analogue of the witch's hat is the upward facing simplicial cone defined over a simplicial domain. For this function with Simplicial PLS, the localization, as in dimension one, becomes the level set.

TABLE III. The behaviour of Simplicial PLS on higher dimensional analogues of the witch's hat

	Average observed number of iterations to convergence	
Dimension	Localisation = level set	Tolerance of 0.1
1	4.8	4.9
2	15	9.6
3	37	17.4
4	–	29.0

The second column of Table III shows the average observed number of iterations, for 100 trials, for this to happen when the dimension is 1, 2 and 3. The third column of Table III shows the average observed number of iterations for convergence, to within $\epsilon = 0.1$ of the global minimum, for Simplicial PLS for dimensions

1, 2, 3 and 4. Evidently these numbers are not linear in dimension. For PAS it is shown in [16] that this quantity is linear in dimension, so this implementation of Simplicial PLS is not "uniformly" close enough to PAS to maintain linearity in dimension. Empirically it appears that Simplicial PLS is SAS for the witch's hat in dimensions greater than one, but evidence suggests that the β values are unbounded. If there were a bound, then Simplicial PLS would be linear in dimension (Corollary 2.1). However, if the β values prove to be bounded by a function that is polynomial in dimension, then polynomial complexity of Simplicial PLS would result.

Following submission of this paper, Hansen *et al.* in [7, 8] have published an extended version of the Piyavskii–Shubert algorithm. It finds a set of disjoint intervals, containing only points with globally ϵ-optimal values, whose union contains all globally optimal points. It would be of interest to compare LPLS with this algorithm.

Acknowledgements

For encouragement and helpful discussions, the authors would like to thank Murray Smith, Stephen Samuels and Roger Wets.

References

1. F. Archetti and B. Betro (1978), A priori analysis of deterministic strategies for global optimisation problems, In *Towards Global Optimisation 2*, L.C.W. Dixon and G.P. Szegö, eds., North Holland: Amsterdam.
2. W. Baritompa (1993), Customizing methods for global optimization – a geometric viewpoint, *J. Global Optimization* **3**, 193–212.
3. P. Basso (1982), Iterative methods for the localization of the global maximum, *SIAM J. Numer. Anal.* **19**, 781–792.
4. H.E. Romeijn and R.L. Smith (1994), Simulated annealing for constrained global optimization, *J. Global Optimization* **5**, 101–126.
5. O.R. Chuyan and A.G. Sukharev (1990), On adaptive and nonadaptive stochastic and deterministic algorithms, *J. of Complexity* **6**, 119–127.
6. S. Graf, R. Daniel Mauldin and S.C. Williams (1986), Random homeomorphisms, *Advances in Mathematics* **60**, 239–359.
7. P. Hansen, B. Jaumard, and Shi-Hui Lu (1992), Global optimization of univariate Lipschitz functions: I. Survey and properties, *Mathematical Programming* **55**, 251–272.
8. P. Hansen, B. Jaumard, and Shi-Hui Lu (1992), Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison, *Mathematical Programming* **55**, 273–292.
9. R.H. Mladineo (1986), An algorithm for finding the global maximum of a multimodal, multivariate function, *Mathematical Programming* **34**, 188–200.
10. N. Patel, R.L. Smith and Z.B. Zabinsky (1988), Pure Adaptive Search in Monte Carlo Optimization, *Mathematical Programming* **43**, 317–328.
11. S.A. Piyavskii (1972), An algorithm for finding the absolute extremum of a function, *USSR Comp. Math. and Math. Phys.* **12**, 57–67.
12. B.O. Shubert (1972), A sequential method seeking the global maximum of a function, *SIAM J. Numer. Anal.* **9**, 379–388.

13. F.J. Solis and R.J-B. Wets (1981), Minimization by random search techniques, *Mathematics of Operations Research* **6**, 19–30.
14. G.R. Wood (1991), Multidimensional bisection applied to global optimisation, *Computers and Mathematics with Applications* **21**, 161–172.
15. G.R. Wood (1992), The bisection method in higher dimensions, *Mathematical Programming* **55**, 319–337.
16. Z.B. Zabinsky and R.L. Smith (1992), Pure Adaptive Search in global optimization, *Mathematical Programming* **53**, 323–338.
17. Z.B. Zabinsky, R.L. Smith, J.F. McDonald, H.E. Romeijn and D.E. Kaufman (1992), Improving Hit-and-Run for global optimization, *J. Global Optimization* **3**, 171–192.