Journal of Global Optimization **20:** 195–212, 2001.

SPT: a stochastic tunneling algorithm for global optimization

E.M. OBLOW
Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364, USA (e-mail: emo@icx.net)

Abstract. A stochastic approach to solving unconstrained continuous-function global optimization problems is presented. It builds on the tunneling approach to deterministic optimization presented by Barhen and co-workers (Bahren and Protopopescu, in: *State of the Art in Global Optimization*, Kluwer, 1996; Barhen et al., Floudas and Pardalos (eds.), TRUST: a deterministic algorithm for global optimization, 1997) by combining a series of local descents with stochastic searches. The method uses a rejection-based stochastic procedure to locate new local minima descent regions and a fixed Lipschitz-like constant to reject unpromising regions in the search space, thereby increasing the efficiency of the tunneling process. The algorithm is easily implemented in low-dimensional problems and scales easily to large problems. It is less effective without further heuristics in these latter cases, however. Several improvements to the basic algorithm which make use of approximate estimates of the algorithms parameters for implementation in high-dimensional problems are also discussed. Benchmark results are presented, which show that the algorithm is competitive with the best previously reported global optimization techniques. A successful application of the approach to a large-scale seismology problem of substantial computational complexity using a low-dimensional approximation scheme is also reported.

Key words: Global optimization, Tunneling, Stochastic, Pijavskij

1. Introduction

In two recent papers, Barhen and co-workers (Barhen and Protopopescu, 1996; Barhen et al., 1997) reported a significant improvement in solving low-dimensional global optimization benchmark problems. They developed a code called TRUST, which employs a descent-tunneling methodology characterized by non-Lipschitzian descent and one-dimensional (1-D) search. We will continue to refer to these 1-D searches as tunneling, when necessary, only to make clear the link to Barhen's work. The essence of this approach is to break the solution of an optimization problem down into two distinct phases: (1) local descent and (2) tunneling (using 1-D search). The descent phase makes use of local derivative information and the wealth of algorithmic experience in finding local minima in continuous functions. The search phase utilizes a generalization of the recent concept of tunneling Levy and Montalvo (1985) to find a new region in which to begin to descend again. Instead of searching for zeros of a polynomial as in (Levy and Montalvo, 1985), however, a one-dimensional search scheme is used to tunnel to the basin of at-

traction of a new local minimum. What makes this combined approach novel and efficient is that by descending into a local minimum before beginning to search for a new descent region, the method avoids repeated descents to the same local minimum. The final solution is thus, in effect, a single descent interspersed with 1-D searches to find the descent basins of new local minima. A key characteristic of this method, using 1-D search to find new basins, is an attractive feature because in general the ratio of a basin's perimeter to its volume favors a 1-D search over multi-dimension (volume) search for anything but a hyperspherical basin (Törn and Zilinskas, 1989). Both of these characteristics seem to explain the improvements Barhen et al. (1997) reported in benchmark testing using the TRUST methodology for a paper in *Science*.

Starting with the Barhen et al. descent-tunneling methodology as the basis for our research efforts, it is clear that for higher dimensional problems the search phase of the algorithm needs improvement. In high-dimensional spaces, search is much more costly in terms of function evaluations than descent. In order to move from benchmarks to realistic high-dimensional problems, therefore, further improvements must be made in TRUST's search methodology. The most limiting characteristic of search, however, is that *local* derivative information is of little (or, in principle, no) use in locating a new basin of attraction different from those already found. We must, therefore, explore what other information can be used to help in this search process. Most recent research indicates that branch and bound approaches address this problem theoretically but are complex and costly to implement in practice (see for example Horst and Tuy, 1993; Törn and Zilinskas, 1989). The bounding process allows certain regions of the search space to be eliminated from further function evaluations.

It is also widely understood that practical global optimization problem cannot be solved in general without imposing resolution or probabilistic convergence constraints on its solution or adding additional information (such as a Lipschitz constant) with which to bound the solution (Wood, 1992; Stephens and Baritompa, 1998). Without such constraints or information, it is impossible to know how to halt an algorithm and declare that the lowest-to-date minimum is the global minimum. Only an exhaustive bounded search is guaranteed to find a global minimum, since no local information can ever be used to characterize the current best answer as definitively being the global minimum. From a practical standpoint, we found that probabilistic constraints and added global information are easier to implement and have, therefore, concentrated our efforts in this area. In this approach only a probabilistic guarantee can be given that the best local minimum found is the global minimum. There is always a chance that another lower minimum will be found, albeit within a user specified probability constraint.

The comments above lead us to believe that a stochastic approach to Barhen's approach, coupled with some limited knowledge of function bounds and Lipschitz-like constants, can be made the basis of a powerful yet simple optimization algorithm. To make maximum use of such bounding information, a low (or one)

dimensional stochastic tunneling approach to covering a high-dimensional space seems to be necessary. Since we are committed to a *stochastic* reformulation of Barhen's method, we reiterate that only a probabilistic guarantee can be made on deciding whether a global minimum has been found. A halting algorithm which stochastically characterizes the solution is thus our aim. We will use a simple PAC (Probabilistically Approximately Correct) criterion successfully developed to halt machine learning algorithms (Oblow, 1992). Without stochastic characterization the only other alternative would be to solve the problem within prescribed resolution bounds using a fixed grid and an associated fixed number of function evaluations.

We propose, therefore, to only make use of global bounding information in a low-dimensional stochastic algorithm to significantly improve the tunneling phase of Barhen's TRUST algorithm. In doing so we utilize the already powerful breakup of the problem into local descent and tunneling. To accomplish this, we adapt the classic Pijavskij one-dimensional bounding procedure (Pijavskij, 1972) to higher dimensions to treat tunneling as a classic stochastic search problem. This coupling of search and descent eliminates many of the problems encountered in the past with bounding approach by using a stochastic version of Pijavskij's method useful exclusively in the search phase. The new approach is called SPT: (S)tochastic (P)ijavskij (T)unneling, suitable for the tunneling-search phase of Barhen's algorithm. This methodology is closely allied with similar extensions of bounded-covering algorithms published by Wood (1992), Shubert (1972), Evtushenko (1985), Mladineo (1986) and Baritompa (1953) and borrows heavily from these references.

We apply the stochastic improvements developed directly to the combined descent and tunneling TRUST algorithm. This allows us to revisit the benchmark calculations Barhen et al. (1997) reported in *Science*, to see what improvements can be made in these already good results. We will also solve a new high-dimensional seismic problem similar to the one reported in the *Science* paper to demonstrate the method's scalability.

2. Problem Formulation

The generic global optimization problem considered in this paper is defined as follows. Let $f(\mathbf{x}) : \mathcal{D} \rightarrow \mathcal{R}$ be a bounded piecewise-continuous differentiable function and $\mathbf{x}$ be an n-dimensional state vector. The function $f(\mathbf{x})$ is referred to as the objective function, and $\mathcal{D}$ its domain. The optimization goal is to find the state vector $\mathbf{x}^*$ which minimizes $f(\mathbf{x})$ in $\mathcal{D}$. Without loss of generality, we shall take $\mathcal{D}$ to be the hyperparallelopiped $\mathcal{D} = \{x_i \,|\beta_i^- \leqslant x_i \leqslant \beta_i^+ \;;\;\; i = 1, 2, \ldots, n\}$,where β_i^- and β_i^+ denote, respectively, the lower and upper bound of the i-th state variable.

We address this unconstrained global optimization problem by breaking its solution into two phases. The first, is a standard local descent phase in which local derivative information is used to converge a descent algorithm to a local minimum. Any efficient descent algorithm can be chosen for this phase of the work.

Many finite difference methods, as well as Jacobian or Hessian based schemes, are available for algorithmic implementation. Once a local minimum has be found, the second phase will employ a stochastic search to find a new local minimum basin of attraction with a lower local minimum to descend into. This procedure is employed to avoid costly repeated descents into already identified local minima, a characteristic weakness of multiple random start algorithms (Ratschek and Rokne, 1988). It should be noted that this step represents an algorithmic decision that might not be optimal for all problem classes. Specifically, this approach performs less efficiently in problems characterized by the existence of a hierarchy of local minima that has increasing numbers of narrower and narrower minima as the function values decrease. New minima become more difficult to find because of decreasing basin sizes in such cases.

To improve the search phase of Barhen's algorithm we use a new stochastic version of Pijavskij's one-dimensional bounding algorithm. The deterministic version of this algorithm (Pijavskij, 1972) has been widely studied in global optimization work in the past. Its geometric complexity in higher dimensions and its slow convergence even in one-dimensional cases have often been cited as the reasons for its limited usefulness in most practical problems (Ratschek and Rokne, 1988). By developing a stochastic version of this traditional algorithm and using it only in the search phase of our methodology, we hope to eliminate these problems entirely. The newly developed SPT (Stochastic Pijavskij Tunneling) algorithm is easily implemented in one-dimension and scales well to higher dimensional problems, although its usefulness decreases exponentially in this latter case.

3. A Stochastic Pijavskij Algorithm

Since Pijavskij's one-dimensional deterministic algorithm is well known (Pijavskij, 1972), only a few words are needed here to highlight the improvements afforded by a new stochastic implementation. The principle used by Pijavskij in his original paper (Pijavskij, 1972) and also employed by Shubert (Shubert, 1972), was to utilize the Lipschitz constant to determine a lower bound for the global minimum of a one-dimensional function. We assume the function f is continuously differentiable and define the Lipschitz constant as: $L = (|f(x) - f(y)|/|x - y|)_{\max}, \forall x, y \in \mathcal{D}, x \neq y$. In practice, it specifies the largest rate of descent a function can have over a region of interest. If the function is known for two values of x, and the maximal descent rate (called an L-line for future reference) makes it possible to estimate a lower bound f_L for the function in the region between the two given values of x. Using a zeroth-order estimate of this lower bound (i.e. f_L^0) and the point at which it occurs $x_{\min}^0$, Pijavskij's method prescribes that a guess for the position of the global minimum x^* should be taken at $x^* = x_{\min}^0$, the intersection of the two L-lines. That is, $f(x_{\min}^0)$ should be evaluated to make a new estimate of the global minimum of the function with the guarantee that $f^* \geqslant f_L^0$. The process can now be iterated to produce a sawtooth lower bounding curve for the function.

As is well documented, because no descent is performed Pijavskij's scheme becomes increasingly inefficient for flatter regions of the search space (i.e., as the global minimum is approached). Local descent making use of local derivative information is clearly much more efficient in such regions. In addition, the extension of the method to higher dimensions makes finding new points to evaluate much more difficult. Multidimensional surface intersections are needed in these cases to find the lower bound points which prescribe the next evaluation point. Several approaches have been used to derive more useful alternatives to this difficult multidimensional task (Shubert, 1972; Evtushenko, 1985; Mladineo, 1986; Ratschek, 1988; Wood, 1992; Baritompa, 1993; Stephens and Baritompa, 1998) but they are all more complex and inefficient in the high-dimensional problems of practical interest. These later solutions are characterized by using multidimensional rectangular or conical elements. We will make use of the conical elements and refer to them as P-cones in the rest of this paper. Clearly an approach that combines descent and bounded search will produce a more efficient scheme than just the simple extensions of Pijavskij's method published to date.

To address this point, we propose to use the key characteristics of Pijavskij's algorithm, the L-lines (and, as needed in higher dimensions, P-cones), in just the search phase of an optimization algorithm to prescribe the next evaluation point. In Figure 1 we show how to use a sequence of five randomly selected points (labeled 1–5) to implement this new scheme. To start the algorithm we select the first point labeled 1 and descend from it until the first local minimum is reached. This minimum has a value of $f^0_{\min}$ and is located at $x^0_{\min}$ in the figure. Noting the value of $f^0_{\min}$ as the lowest value so far evaluated, we now do a random search of x values trying to find a new value of $f(x)$ which is lower than $f^0_{\min}$. In the figure this first search phase ends with the selection of point 4 which has a value lower than $f^0_{\min}$. Before we find point 4, however, we sample x values randomly, successively dropping cones with slopes corresponding to the L-lines from each newly selected points to exclude regions of the x-axis from further sampling. The intersection of these lines with the line corresponding to $f^0_{\min}$ defines these exclusion regions. The darkened regions along the $f^0_{\min}$ line where the cones from point 1 (and all of its descent points), point 2, and 3 intersect $f^0_{\min}$ prescribe these excluded x-regions in the figure. These are valid exclusion regions because they cannot produce f values lower than $f^0_{\min}$ if the L-lines truly represent a Lipschitz constant for the region being searched. Once point 4 is sampled, we descend from it until we reach $f^1_{\min}$ at $x^1_{\min}$. The cones from all points evaluated through point 4 can now be used to extend the cones down to intersect $f^1_{\min}$ and the random search phase can begin again. Since a larger range of x-values is now excluded from search, point 5 is chosen next and we descend from there to $x^2_{\min}$. This process can be continued indefinitely until the whole x region between a and b is excluded and $f^1_{\min}$ at $x^2_{\min}$ is found to be the global minimum in this case.

This figure thus illustrates that while Pijavskij uses the L-lines to prescribe the next point to evaluate f, we use them with reverse reasoning to define the concept

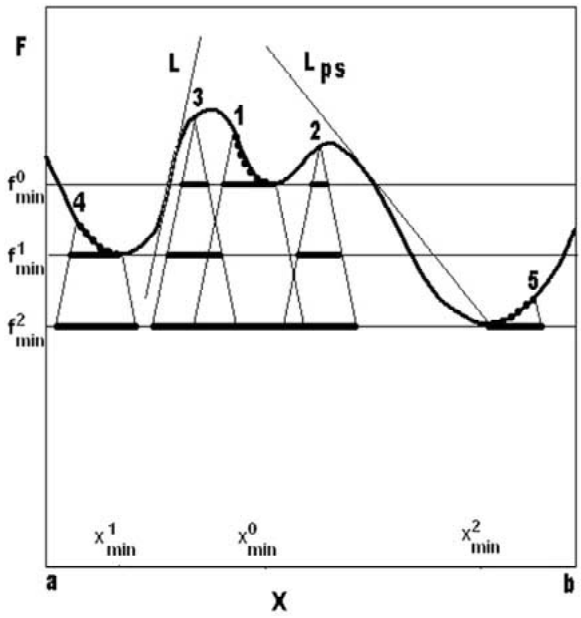

Figure 1. SPT iteration scheme.

of exclusion regions. In doing so we provide the basis for developing a stochastic search version of Pijavskij's basic scheme for finding new basins of attraction for a descent algorithm to work in. To produce a basic stochastic algorithm then, we can use a *region-of-possibility* for finding a point lower than the current lowest local minimum to define a sampling region for selecting a new point for evaluation. Pijavskij's algorithm essentially takes the midpoint of the lowest subregion of this region. A stochastic algorithm with a uniform distribution over the whole subregion would have Pijavskij's point as its average value. Sampling over all possible subregions is simpler, however, and such a procedure should, on average, follow a sequence of Pijavskij-prescribed evaluations.

A complementary algorithm using a rejection technique can also be used to meet this end. This approach would randomly sample over the whole x-domain and reject sample points which lie outside the *region-of-possibility* (i.e. those in the excluded region) until an acceptable point is found. Both techniques have their advantages and disadvantages. They both require search and storage operations to determine intervals of possibility or exclusion and they are the complements of each other in an algorithmic sense. Because of the difficulties entailed in extending a direct sampling method to higher dimensions where function evaluations are usually the most costly component of an optimization algorithm we chose the rejection approach for the SPT algorithm. The natural availability of a halting test based on the probability of finding a point lower than any one found before, also favors this rejection technique.

4. The General SPT Algorithm

The general stochastic multi-dimensional SPT algorithm can therefore be described as follows:

(1) Initialize with constants $N_{\max} = \log \delta / \log \epsilon$, L, and $N = 0$. Here, L is the Lipschitz constant and N counts the number of function evaluations the algorithm makes.

(2) Select a random starting point $\mathbf{x} = \mathbf{x}^1$ and set $f^* = f(\mathbf{x}^1)$. Increment $N = N + 1$.

(3) Use a descent algorithm to converge to local minimum incrementing N for each lower value of f found in the process. Set $f^* = f^N = f(\mathbf{x}^N)$, where x^N is the point at which f attains the lowest value found by the descent algorithm (i.e., f^N).

(4) Compute Pijavskij P-cone radii r_i for all N points evaluated so far as follows: $r_i = (f^i - f^*)/L, \forall \mathbf{x}^i, i = 1, \ldots, N$.

(5) Iteratively select new random points from the region outside of all P-cones intersecting f^*, searching for a value of f less than f^*. Increment N and update all P-cones for each new point if $f^N > f^*$. Halt this procedure if more than $N_{\max}$ points are selected without finding a value of f lower than f^* or all x are excluded.

(6) Go to step (3) if step (5) is successful in finding an $f < f^*$, otherwise halt and report $f^* = f_{\text{globalmin}}$.

In general this algorithm attempts to make a stochastic ϵ-cover of the **x**-domain with confidence δ (see Oblow, 1992) using the only information on the Lipschitz constant L to exclude certain unnecessary function evaluations. The larger a function value is compared to the lowest value of f found to date, the larger the exclusion region is for finding new function evaluation possibilities. The exclusion process is thus aided by large function values and lower minimum values. At some point before N_{max} is reached it is possible that the exclusion region will cover the whole **x**-space and the algorithm will halt having found the global minimum. If N_{max} is exceeded, the algorithm will halt and the global minimum might not be found. In this case only a probabilistic statement can be made about the f^*. That is, the probability of choosing an x value whose $f(x)$ is lower than f^* is less than ϵ with confidence δ. This constitutes only a PAC (probabilistically approximately correct (Oblow, 1992)) solution to the problem for the chosen ϵ-cover.

Based on the description above, this new procedure can best be described as an importance sampling technique for stochastically finding a value of $f(\mathbf{x})$ lower than an existing estimate of f^*. The bias of the sampling is toward regions of the x space that have lower function values than the current estimate of the global minimum f^*. Used in conjunction with a local descent algorithm, it goes from local minimum to local minimum to try to find the global minimum or the deepest minimum within prescribed stochastic confidence limits.

5. Lower Bound Approximations

A striking feature of the algorithm given in the last section is the fact that the larger the function values are compared to the current minimum value, the more x-space is excluded from future function evaluation. Thus, larger function values or lower minimum values make the method more efficient. This feature can be used to great advantage by adding a single additional piece of information to the algorithm to improve its efficiency. This piece of information is an estimate of the lowest value the function can attain in the domain of interest (i.e., f_M, an estimate of the global minimum). Instead of letting the algorithm develop local minima estimates of the global minimum as it proceeds, a good low estimate can be input right from the start. This added parameter will immediately increase the efficiency of the algorithm by increasing the exclusion region of potential function evaluations. This efficiency increase will always improve the original algorithm until a local minimum is found which has a value lower than the estimate.

This estimation process can be made formal and adaptive by the following modification of the original algorithm:

(1) Initialize with constants $N_{\max} = \log \delta / \log \epsilon$, L, f^M, and $N = 0$. Here, any convenient method for estimating f^M, a lower bound for $f_{\text{globalminimum}}$, will do.

(2) Select a random starting point $\mathbf{x} = \mathbf{x}^1$ and set $f^* = f(\mathbf{x}^1)$. Increment $N = N + 1$.

(3) Use a descent algorithm to converge to local minimum incrementing N for each lower value of f found in the process. Set $f^* = f^N = f(\mathbf{x}^N)$, where x^N is the point at which f attains the lowest value found by the descent algorithm (i.e., f^N).

(4) Compare f^M and f^* and choose the lower value (i. e. set $f^M = \min(f^M, f^*)$).

(5) Compute Pijavskij P-cone radii r_i for all N points evaluated so far using f^M as follows: $r_i = (f^i - f^M)/L$, $\forall \mathbf{x}^i$, $i = 1, \ldots, N$.

(6) Iteratively select new random points from the region outside of all P-cones intersecting f^M, searching for a value of f less than f^*. Increment N and update all P-cones for each new point if $f^N > f^*$. Halt this procedure and report $f^* = f_{\text{global min}}$ if more than $N_{\max}$ points are selected without finding a value of f lower than f^*.

(7) Go to step (3) if step (6) is successful in finding an $f < f^*$, otherwise set $f^M = (f^M + f^*)/2$ and go to step (5).

The purpose of using f^M is simply to increase the exclusion regions of the original algorithm forcing it to look for potentially much lower values of f earlier than it would normally have. A lower bound on the global minimum derived from the x region boundaries and the Lipschitz constant would serve this purpose, but any reasonably small value will also work if altered adaptively as prescribed in step (4). If an overestimate of f^M caused the P-cones computed in step (5) to exclude the whole x region, then reducing its size in step (4) costs no additional

function evaluations and the algorithm could continue as usual. So long as function evaluations dominate the computational effort required for this algorithm, using even unrealistically low values of f^M will introduce only negligible additional computations and yet the efficiency of the original algorithm will be improved.

6. Pseudo-Lipschitz Approximations

As stated in Section 2 the general algorithm requires an estimate of the Lipschitz constant as the only additional piece of information used to increase search efficiency. Since analytic computer evaluated functions are our focus, the availability of a good Lipschitz constant is assumed to exist. A closer examination of the current algorithm however, reveals that a true Lipschitz constant is not really required for this algorithm to work. What is required is a pseudo-Lipschitz constant, defined as the smallest magnitude slope of any cone drawn from the global minimum which remains just above the graph of the function in the basin of attraction of that global minimum. This constant called L_{ps} guarantees that no function evaluation will ever cause the global minimum point to be excluded from potential function evaluation. The point at which this $L - ps$ cone just touches the graph is the point at which the global minimum is finally excluded from evaluation by a P-cone drawn from that point. It should be noted that even a square-well with infinite Lipschitz constant still has a finite L_{ps} provided only a single global minimum is to be found out of a potential infinity of possible values. This concept is thus very powerful in practice if a way can be found to estimate L_{ps} rather than L.

The difficulty with this new concept is, however, precisely its estimation. In principle this cannot be done without knowing a great deal of information about the size and shape of the global minimum basin, the very thing we presumably are trying to find. In practice however we can use an adaptive procedure to estimate this parameter since we are ultimately willing to accept only a probabilistic answer to the global optimization problem anyway. A possible procedure for estimating this constant is given as follows:

(1) Initialize with constants $N_{\max} = \log \delta / \log \epsilon$, f^M, and $N = 0$.
(2) Select a random starting point $\mathbf{x} = \mathbf{x}^1$ and set $f^* = f(\mathbf{x}^1)$. Increment $N = N + 1$.
(3) Use a descent algorithm to converge to local minimum incrementing N for each lower value of f found in the process. Set $f^* = f^N = f(\mathbf{x}^N)$, where x^N is the point at which f attains the lowest value found by the descent algorithm (i.e., f^N).
(4) If f^* is first minimum found, construct L_{ps} from its definition using all evaluated points in the descent to f^*.
(5) Compute Pijavskij P-cone radii r_i for all N points evaluated so far using f^* as follows: $r_i = (f^i - f^*)/L, \forall \mathbf{x}^i, i = 1, \ldots, N$.
(6) Iteratively select new random points from the region outside of all P-cones intersecting f^*, searching for a value of f less than f^*. Increment N and

update all P-cones for each new point if $f^N > f^*$. Halt this procedure and report $f^* = f_{\text{global min}}$ if more than $N_{\max}$ points have been selected without finding a value of f lower than f^*.

(7) Go to step (3) if step (6) is successful in finding an $f < f^*$, otherwise set $L_{ps} = 2 * min(L_{ps}, L)$ and go to step (5).

Here, we estimate L_{ps} from the first minimum found and proceed using all the procedures used in the original SPT algorithm. In all cases we will reach a point where $L_{ps} = L$ and the algorithm will almost be identical to the original. The difference is that if the real L_{ps} was smaller than L the sampling sequence will have a higher probability of finding the real global minimum than the original procedure. If the real Lipschitz constant had been used as the basis for a first cut estimate of L_{ps} then regions that would not have be sampled using the real L_{ps} will not be excluded, thus increasing the probability not finding the real global minimum. Using L itself will either cause the algorithm to halt on the exclusion condition or the $N_{\max}$ limit. If the real $L_{ps} \leqslant L$, we have pruned the sampling space more efficiently with a better estimate of chance of opening up further sampling regions by decreasing the estimate of L_{ps} below that of L_{ps}. As before, if the $N_{\max}$ limit is reached then the problem can be said to be solved only to the prescribed probabilistic confidence.

It should be noted here that if this adaptive algorithm halts on the $N_{\max}$ limit being reached, it does not mean that $N_{\max}$ function evaluations have been made. Only $N_{\max}$ x values have been chosen. The real efficiency of the algorithm will depend on the size of the final exclusion region after halting takes place. In practice orders of magnitude fewer function evaluations might be needed to achieve a probabilistic confidence equivalent to a fixed grid of $O(N_{\max})$. For a relatively flat (or even constant) function, $N_{\max}$ function evaluations will probably be approached, but few other algorithms can be conceived of that will do much better than this on such a problem.

7. One-Dimensional Approximations

In practice this algorithm suffers from one limiting characteristic: its efficiency compared to pure random search diminishes as the dimension of the x-space increases. It becomes increasingly difficult to cover higher dimensional spaces with P-cones based on a single Lipschitz-like constant. In practice the exclusion fraction approaches zero in high dimensional problems and the algorithm is effectively reduced to a pure random search. We expect the method to be useful in low-dimensional problems (as will be demonstrated in the next benchmark testing section) but practically only random search is achieved in very large-dimensional problems.

A little further analysis reveals, however, that the maximum efficiency of the method can be extended to higher dimensional problems by accepting a very useful approximation to basic SPT approach. This approximation is based on the

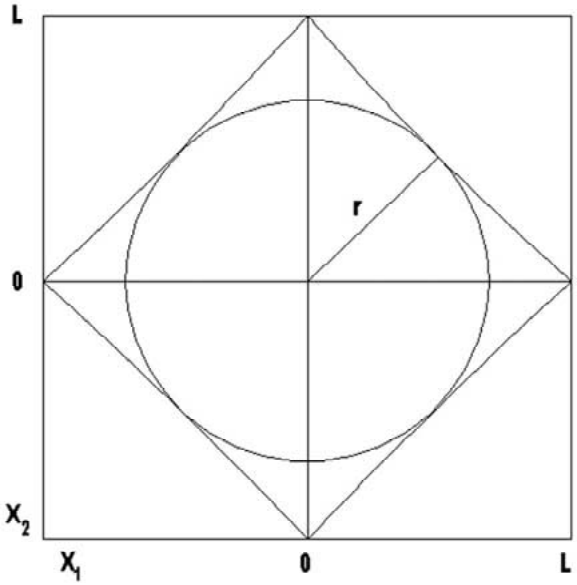

Figure 2. Two-dimensional exclusion regions.

construction of a multidimensional Lipschitz constant from partial derivative information. This construction is illustrated here by assuming that in each individual x-direction the maximum Lipschitz constant $L_x, \forall x$ can be estimated and it is roughly equal for all coordinate directions. The real Lipschitz constant in d-dimensions for this illustrative case is thus: $L = O(\sqrt{d}L_x)$. A graphical picture of this process and the resulting P-cone in two-dimensions is shown in Figure 2.

This case an be generalized by noting that L is always larger than any the individual $L_x's$ and thus the P-cone is smaller than that derived from any individual L_x including the maximum L_x for any direction. If we take the real shape of the excluded region for a single x value based on the maximum coordinate derivatives in each of the coordinate directions we get a starfish shaped region. As much higher dimensional problems are considered, L becomes the smallest P-cone that can be inscribed in the starfish exclusion region and the starfish itself becomes almost a collection of one-dimensional exclusion problems with virtually no measurable exclusion volume in d-dimensions. This means in practice that the algorithm would be no better than pure random search, since a Lipschitz constant based P-cone would have almost negligible measure. In this case the algorithm will halt when a maximum number of sample points ($N_{\max}$) limit is reached (rather than a halt based on any exclusion region limit).

The above facts cannot be avoided in principle. A heuristic approach which utilizes the availability of L_x information, however, can be proposed. We simply recast the algorithm into a search of each individual x-coordinate direction starting from a random point in d-dimensions. In this fashion, td very efficient one-dimensional

searches can be used to solve the larger problem with some significant efficiency advantages accruing because of availability of L_x information. The maximum efficiency of one-dimensional problem exclusions regions can thus be transferred to higher dimensional problems with only minimal approximations being used. A d-dimensional problem reduced to a collection of d uncoupled one-dimensional problems is the most efficient use of Lipschitz or pseudo-Lipschitz information.

The one-dimensional implementation of the SPT approach is the one we are, therefore, recommending for higher dimensional problems. It effectively produces a random line cover of a d-dimensional space with an underlying grid of density $N^d_{\min}$ points. Of these points, only a fraction of them are required for function evaluation depending on the 1-D exclusion regions that develop iteratively as more function evaluations are made. In this sense it is guaranteed to be at least as efficient as a fixed $N^d_{\min}$ grid evaluation of a d-dimensional function. The simplicity and speed of the algorithm should give it a competitive advantage over most existing approaches to global optimization in high dimensional problems. This advantage is demonstrated well in the 314-dimensional seismic benchmark we present later in this paper.

8. Science Benchmark Results

This section presents results of benchmarks carried out to assess the new SPT algorithm using several standard multidimensional test functions taken from the literature. A description of each test function can be found in Barhen and Protopopescu (1996); Cvijovic and Klinowski (1995). In Table 1, the performance of SPT is compared to the best competing global optimization methods including the previous version of TRUST published in *Science* (Barhen et al., 1997). Here the term 'best' indicates the best widely reported reproducible results the authors could find for the particular test function. The criterion for comparison is the number of function evaluations. All SPT results were converged to a convergence criteria of $\epsilon = 10^{-5}$.

In Table 1, the benchmark labels BR (Branin), CA (Camelback), GP (Goldstein-Price), RA (Rastrigin), SH (Shubert), and H3 (Hartman), refer to the test functions considered. The following abbreviations are also used: SDE is the stochastic method of Aluffi-Pentini (Alluffi-Pentini et al., 1985); EA denotes the evolution algorithms of Yong, Lishan, and Evans (Yong et al., 1995), or Schneider (Schneider et al., 1996); MLSL is the multiple level single linkage method of Kan and Timmer (Kan and Timmer, 1985); IA is the interval arithmetic technique of Ratschek and Rokne (Ratschek and Rokne, 1988); TUN is the tunneling method of Levy and Montalvo (Levy and Montalvo, 1985); and TS refers to the Taboo Search scheme of Cvijovic and Klinowski (Cvijovic and Klinowski, 1995) and TRUST refers to Barhen's TRUST code (Barhen et al., 1997). The results demonstrate that TRUST is substantially faster than these state-of-the-art method, albeit needing a number of parameters to be optimized for each benchmark. The SPT results are comparable to

Table 1. Number of function evaluations required by different methods to reach a global minimum of several Standard Test Functions.

Method	BR	CA	GP	RA	SH	H3
SDE	2700	10822	5439	–	241215	3416
EA	430	–	460	2048	–	–
MLSL	206	–	148	–	–	197
IA	1354	326	–	–	7424	–
TUN	–	1469	–	–	12160	–
TS	492	–	486	540	727	508
TRUST	55	31	103	59	72	58
SPT	67	26	123	140	150	75

the TRUST numbers and use only a single adjustable parameter and are averaged over 100 random (vs 2^d deterministic) runs.

9. Large-scale Application

To assess the performance of SPT for a large-scale practical application, the high-dimensional problem of residual statics corrections for seismic data was chosen. A description of the residual statics problem appears in Barhen et al. (1997) but for ease of understanding the brief description in Barhen et al. (1997) is repeated here.

Basically the optimization problem arises from the seismic energy measurement process. This energy is detected by receivers that are regularly spaced along a grid that covers the domain being explored. A source is positioned at some grid location to produce a shot. Time series data is collected from the detectors for each shot, then the source is moved to another grid node for the next shot.

Degradation of seismic signals usually arises from near-surface geologic irregularities (Rothman, 1985; Yilmaz, 1988; DuBose, 1993) which include uneven soil densities, topography, and significant lateral variations in the velocity of seismic waves. The most important consequence of such irregularities is the recording of a distorted image of the subsurface structure due to delays travel times of seismic waves in a vertical neighborhood of every source and receiver. To improve the quality of the seismic analysis, timing adjustments (i.e., 'statics corrections') must be performed. This problem has generally been formulated in terms of global optimization, and, to date, Monte-Carlo techniques (Sen and Stoffa, 1995) (e.g., simulated annealing, genetic algorithms) have provided the primary tools for seeking a potential solution. Such an approach is extremely expensive, and a major need has existed for better methods which would allow the accurate and efficient solution of large-scale problems.

The statics correction optimization problem can be summarized as follows. Acoustic signals are shot from N_s source locations and received by N_r sensors. Each signal reflects at a midpoint k, $k = 1, ..., N_k$. A trace t corresponds to seismic energy traveling from a source s_t to a receiver r_t via a midpoint k_t. We denote by D_{ft} the (complex) Fourier coefficient of frequency f ($f = 1, .., N_f$) for trace t, $t = 1, ..., N_t \leqslant N_r N_s$. Ideally, after they have been corrected for normal move out (Yilmaz, 1988), all traces corresponding to the same midpoint carry coherent information . If there were no need for statics corrections, all signals, stacked by their common midpoint, should be in phase and yield a maximum for the total power

$$E(\mathbf{S}, \mathbf{R}) = \sum_k \sum_f |\sum_t \exp[2\pi i\ f\ (S_{s_t} + R_{r_t})] D_{ft} \delta_{kk_t}|^2\ . \tag{9.1}$$

In (1), the statics corrections $\mathbf{S} = (S_1, .., S_{N_s})$, and $\mathbf{R} = (R, .., R_{N_r})$ are now considered independent variables. Their optimum values are found by maximizing the power E. This expression highlights the multimodal nature of E which, even for relatively low dimensional **S** and **R**, has been found to exhibit a very large number of local minima.

A smaller problem in a 154-dimensional space has already been reported in the Barhen *Science* paper. To assess the performance improvements of the SPT algorithm, we considered a much larger problem involving N_s = 100 shots and N_r = 216 receivers yielding a 316-dimensional space. A data set consisting of N_t = 4776 synthetic seismic traces folded over N_k = 423 common midpoint gathers was obtained from CogniSeis Corporation (DuBose). It uses N_f = 118 Fourier components for data representation. This set is somewhat typical of collections obtained during seismic surveys by the oil industry and is thus representative of the complexity underlying generic residual statics problems.

To derive a quantitative estimate of SPT's impact, the next two figures show the original subsurface map without statics corrections and the same map derived after solving the statics optimization problem. The SPT results illustrated in Figure 4 show significant improvement over the uncorrected results shown in Figure 3. These results also show considerable improvement over an attempt to solve this same problem with industry standard methods (see discussion in Barhen et al., to appear). The optimization run used an adaptive method for estimating the pseudo-Lipschitz constant which resulted in a value of $L_{ps} = 50$. A one-dimensional grid of $N_{\max} = 100,000$ was used to converge each individual one-dimensional ray. The approximate global minimum found by SPT was $f_{\min}$=−2442 compared with an industry best result of f_{min}=−2230. The industry method required close to 40 h of computer time to converge compared with less than one hour for SPT on comparable computing hardware. TRUST itself, was unable to solve the problem in its full dimensional entirety. A stack decomposition method similar to that used to solve the original *Science* paper statics 154-dimensional problem was needed to achieve a result of $f_{\min}$=−2320.

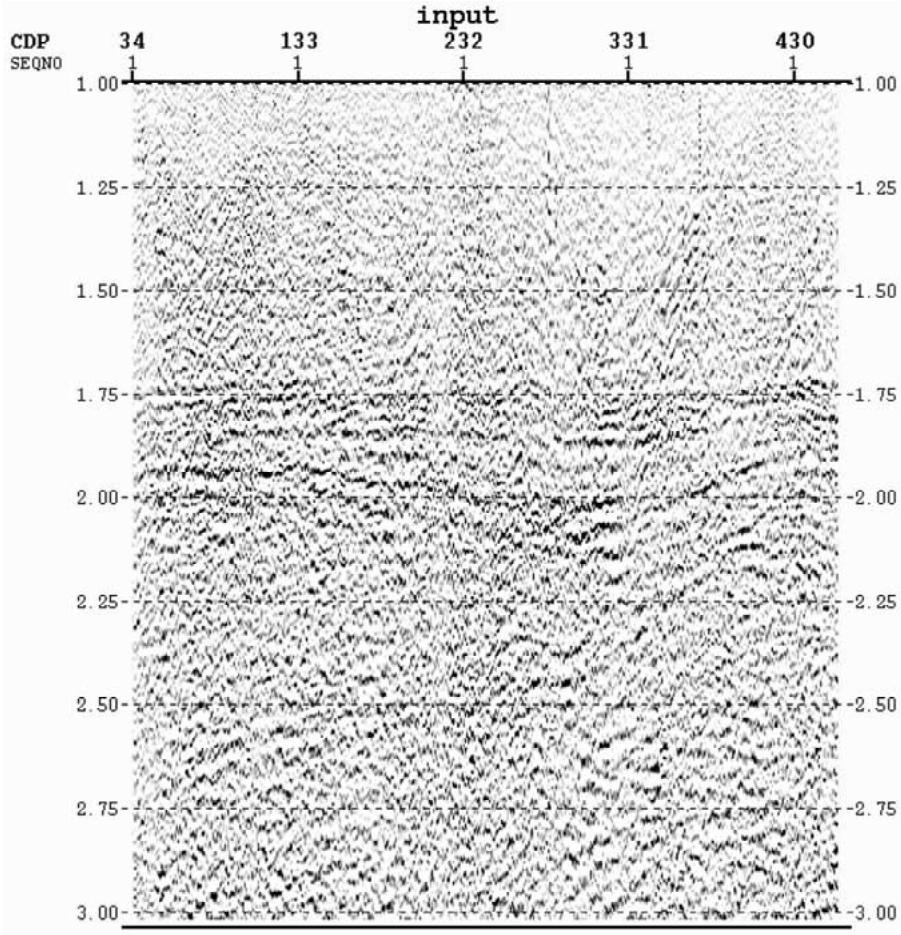

Figure 3. Unoptimized subsurface seismic map.

10. Conclusions

From the results presented, one can conclude that a stochastic approach to tunneling makes a considerable improvement in Barhen's TRUST approach to solving continuous-function global optimization problems. This new approach builds on the deterministic tunneling-descent methodology presented by Barhen et al. (Barhen and Protopopescu, 1996; Barhen and Protopopescu, 1997), as one of the best benchmark methods to date, but uses a rejection-based stochastic procedure to locate new local minima descent regions. The method employs a series of local descents interspersed with stochastic search to find new local minima, using a rejection-based stochastic procedure to prune the tunneling search space traversed in searching for new local minima descent regions. It uses a fixed Lipschitz-like constant to reject unpromising regions in this search space thereby increasing the efficiency of the tunneling process. The algorithm is most easily implemented in

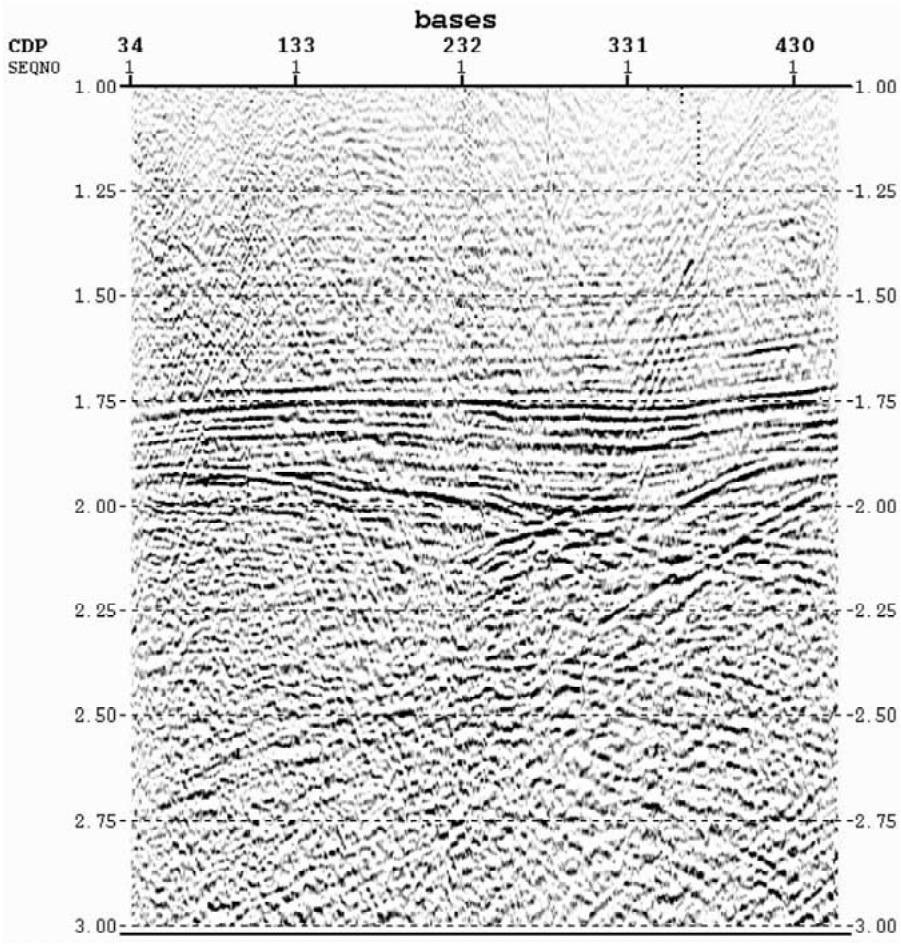

Figure 4. Optimized subsurface seismic map.

low-dimensional problems, which allows it to be used efficiently as a heuristic in large scale problems. Several variations of the basic algorithm should allow for further improvements in efficiency by allowing approximations to be made to estimating the algorithms parameters. The benchmark results presented, show that the algorithm is competitive with the best previously reported global optimization techniques. A successful application of the approach (using the one-dimensional approximation search scheme) to a large-scale seismology problem of substantial computational complexity bears out the practical applicability of the approach.

Acknowledgements

This research was performed at the Center for Engineering Science Advanced Research (CESAR), Computer Science and Mathematics Division, Oak Ridge Na-

tional Laboratory. Funding was provided by the Engineering Research Program of the Office of Basic Energy Sciences and by the Advanced Computing Technology Initiative Program of the U.S. Department of Energy under contract DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp.

References

Aluffi-Pentini, F., Parisi, V. and Zirilli, F. (1985), 'Global Optimization and Stochastic Differential Equations,' *Journal of Optimization Theory and Applications* 47: 1.

Barhen, J. and Protopopescu, V. (1996), 'Generalized TRUST Algorithm for Global Optimization,' *State of the Art in Global Optimization*, C.A. Floudas and P.M. Pardalos eds., pp. 163–180. Kluwer.

Barhen, J., Protopopescu, V. and Reister, D. (1997), 'TRUST: A Deterministic Algorithm for Global Optimization,' 276: 1094–1097.

Barhen, J. Reister, D. Oblow, E. and DuBose, J. accepted for publication in Geophysics.

Baritompa, W. (1993), 'Customizing Methods for Global Optimization - A Geometric Viewpoint,' *J. of Global Optimization* 3: 193–212.

Cetin, B., Barhen, J. and Burdick, J. (1993), 'Terminal Repeller Unconstrained Subenergy Tunneling (TRUST) for Fast Global Optimization,' *Journal of Optimization Theory and Applications* 77: 97.

Cvijovic D. and Klinowski, J. (1995), 'Taboo Search: An Approach to the Multiple Minima Problem,' *Science* 267: 664.

DuBose, J. (1993), 'Practical Steps Toward Realizing the Potential of Monte Carlo Automatic Statics,' *Geophysics*, 58: 399.

Evtushenko, Yu.G. (1985), 'Numerical Optimization Techniques,' *Optimization Software Inc., New York*, p. 561.

Floudas, C.A. and Pardalos, P.M. (1996), *State of the Art in Global Optimization: Computational Methods and Applications*. Kluwer.

Horst R. and Pardalos, P.M. (1995), *Handbook of Global Optimization* Kluwer.

Horst, R., Pardalos, P.M. and Thoai, N.V. (1996), *Introduction to Global Optimization*. Kluwer.

Horst, R. and Tuy, H. (1993), *Global Optimization*, 2nd ed. Springer-Verlag.

Horst, R. and Tuy, H. (1993), *Global Optimization*, 2nd ed. Springer-Verlag.

Kan, A.H.G.R. and Timmer, G.T. (1985), 'A Stochastic Approach to Global Optimization,' in *Numerical Optimization*, eds. P. T. Boggs, R. H. Byrd, and R. B. Schnabel, pp. 245–262. SIAM.

Levy, A. and Montalvo, A. (1985), 'The Tunneling Algorithm for the Global Minimization of Functions,' *SIAM Journal on Scientific and Statistical Computing* 6: 15.

Mladineo, R.H. (1986), 'An Algorithm for Finding the Global Maximum of a Multimodal, Multivariate Function,' *Mathematical Programming* 34: 188–200.

Oblow, E.M. (1992), 'Implementing Valiant's Learnability Theory Using Random Sets,' *Machine Learning* 9: 45–73.

Pijavskij, S. (1972), 'An Algorithm for Finding the Absolute Extremum of a Function,' *USSR Computational Mathematics and Mathematical Physics* 12: 57–67.

Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization* Ellis Horwood.

Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization* Ellis Horwood.

Rothman, D. (1985), 'Nonlinear Inversion, Statistical Mechanics and Residual Statics Estimation,' *Geophysics* 50: 2784.

Schneider, G. Schuchhardt, J. and Wrede, P. (1996), 'Evolutionary Optimization in Multimodal Search Space,' *Biological Cybernetics* 74: 203.

Sen, M. and Stoffa, P.L. (1995), *Global Optimization Methods in Geophysical Inversion* Elsevier.

Shubert, B.O. (1972), 'A Sequential Method Seeking the Global Extremum of a Function,' *SIAM Journal of Numerical Analysis* 9: 379–388.
Stephens, C.P. and Baritompa, W. (1998), 'Global Optimization Requires Global Information,' *J. of Global Optimization* 96: 575–588.
Törn, A. and Zilinskas, A. (1989), *Global Optimization* pp. 66–67. Springer-Verlag.
Törn, A. and Zilinskas, A. (1989), *Global Optimization* pp. 30–38. Springer-Verlag.
Törn, A. and Zilinskas, A. (1989), *Global Optimization* Springer-Verlag.
Wood, G.R. (1992), 'The Bisection Method in Higher Dimensions', *Mathematical Progrramming* 55: 319–337.
Yilmaz, O. (1988), *Seismic Data Processing*, Society of Exploration Geophysicists Tulsa.
Yong, L., Lishan, K. and Evans, D.J. (1995), 'The Annealing Evolution Algorithm as Function Optimizer,' *Parallel Computing* 21: 389.
The authors thank J. DuBose for providing the benchmark problem.