Stochastic Techniques for Global Optimization: A Survey of Recent Advances

FABIO SCHOEN
Dept. Information Sciences, University of Milano, via Moretto da Brescia, 9, I-20133 Milano, Italy.
E-mail: schoen@imiuccaunimi.it

(Received: 4 April 1990; accepted: 4 December 1990)

Abstract. In this paper stochastic algorithms for global optimization are reviewed. After a brief introduction on random-search techniques, a more detailed analysis is carried out on the application of simulated annealing to continuous global optimization. The aim of such an analysis is mainly that of presenting recent papers on the subject, which have received only scarce attention in the most recent published surveys. Finally a very brief presentation of clustering techniques is given.

Key words. Stochastic algorithms, simulated annealing, clustering.

Introduction

In this paper stochastic techniques for the optimization of multi-modal functions are presented. Within this class are indeed the best-performing algorithms for the global optimization problem. One of the main reasons for this to happen is that the problem of globally optimizing a real-valued function is inherently intractable (unless hard restrictions are imposed on the objective function) in that no practically useful characterization of the global optimum is available (like, for example, in the well known case of smooth local optimization, conditions on the gradient and hessian of the objective function); this fact has given rise to different streams of research which may be very roughly classified as "deterministic" and "stochastic". To the first class belong those algorithms which implicitly search all of the function domain and thus are *guaranteed* to find the global optimum; algorithms within this class are forced to deal with severely restricted classes of functions (e.g. Lipschitz continuous functions with *known* Lipschitz constant). Unfortunately, in most cases it is not sensible to assume a particular structure on the objective function; moreover, even if it is known that the objective function is, e.g., Lipschitzian, it is often computationally infeasible to search for a guaranteed global optimum, as the number of computation required increases exponentially with the dimension of the feasible space. To overcome the inherent difficulties of guaranteed-accuracy algorithms, much research effort has been devoted to algorithms in which a stochastic element is introduced; this way the *deterministic* guarantee is relaxed into a confidence measure; the rationale behind this approach is mainly due to the very nature of actual global optimization problems: deterministic global optimization algorithms tune their behaviour on

Journal of Global Optimization **1**: 207–228, 1991

the worst possible objective function, while stochastic algorithms tend to smooth out pathological functions, being more sensible to some kind of an "average" objective function (where the word average should be interpreted in a very loose sense).

Within the class of global optimization algorithms in which some stochastic element is introduced, the following broad distinction can be made:

- algorithms which assume a *stochastic model.* Typically in this class the objective function itself is seen as a sample path of some stochastic process; information on the objective function is sequentially updated via Bayes' rule. In some approaches a stochastic model is given only on the local optimizers and/or the associated function value. A detailed review of algorithms within this class is given in [5].
- algorithms which are stochastic in nature, i.e. algorithms in which the placement of observations is based on the generation of random points in the domain of the objective function. This paper will present algorithms within this class.

Within both frameworks several successful algorithms have been proposed in which probabilistic as well as heuristic techniques are mixed.

Obviously a definite statement about the claimed superiority of stochastic algorithms over deterministic ones is impossible to give; however, theoretical considerations as well as practical experience suggest that for problems of moderate to high dimension the use of stochastic technique is the only feasible approach.

Recently a very good and comprehensive survey of global optimization appeared [39]: the interested reader is addressed to this reference for a thorough coverage of all the aspects of global optimization. The paper presented here (differently from the cited survey) will be strongly biased towards stochastic algorithms; in order not to duplicate the work in the cited survey, in this paper some aspects of stochastic techniques will be presented as well as recent advances (not included in the book of Törn and Žilinskas) discussed. In particular, recent results and new directions in the area of the application of simulated annealing to the optimization of multimodal functions will be presented. This is an area of quite active research which deserved too little attention in recent surveys, so it seems natural to devote to it a significant part of this paper.

1. The Global Optimization Problem

In this section the basic definitions and notation will be set up. The optimization problem hereon called "global optimization problem" is defined as that of finding

$$f^* = \min_{x \in A} f(x) \tag{1}$$

where:

- A is a compact subset of $\mathbb{R}^N$, with $N > 0$. Even if not strictly necessary, in the following it will be tacitly assumed that A is the unit hypercube in $\mathbb{R}^N$; this way the optimization problem will be "almost unconstrained", i.e. only simple bounded; moreover the random generation of a point in A, which is a fundamental step in most of the stochastic algorithms, will be an easier task (although far from trivial, particularly for large N).

- $f: A \mapsto \mathbb{R}$ is the objective function; its smoothness is assumed to be that required by the algorithm: for example while simple random sampling of f over A requires almost no condition on f, algorithms based on performing some local searches usually require f to be twice continuously differentiable. As a matter of facts, it will be generally assumed that f is at least continuous, but sometimes only measurability will be required.

- f^* is the global optimum of f over A; under the above conditions of compactness of A and continuity of f a finite f^* is guaranteed to exist. If the continuity assumption is dropped, the definition (1) is changed to the following

  $$f^* = \inf\{y : \mu(x \in A : f(x) < y) = 0\}$$

  where $\mu(\cdot)$ is a positive measure on the Borel sets of $\mathbb{R}^N$ (usually the Lebesgue measure). This way the problem is redefined as that of finding the *essential infimum* of f over A.

The weak assumptions stated above on the objective function f and its domain A are required only to separate the very problem of global optimization from the (well known, but far from being easy) subproblems of *local* optimization and of random vector generation.

It is to be noticed also that usually algorithms for global optimization produce, as an output, not only an estimate, say $\hat{f}$, of the global optimum, but also a point $\hat{x} \in A$ such that $f(\hat{x}) = \hat{f}$. This does *not* mean that $\hat{x}$ is a good estimate of the optimizer, in the sense that there is no guarantee that $\hat{x}$ is close to the set

$$X^* = \{x^* \in A : f(x^*) \leq f(x) \forall x \in A\} .$$

Indeed it is easily shown that the problem of determining an accurate estimate of a global optimizer is mathematically ill-posed, in the sense that very similar objective functions (where similarity is meant with respect to some distance measure) may have very distant global optimizers.

2. Stochastic Algorithms

A general stochastic algorithm for global optimization consists of three distinct major steps: a sampling step, an optimization step, a check of some stopping criterion. More explicitly, the sampling step basically consists of generating random points in the domain A and computing the associated function value; the optimization step consists of applying a local optimization routine to some (possibly none or all) of the sampled points; the stopping criterion, which often represents the most critical part in the design of an algorithm, is used to stop the algorithm when there is sufficient evidence that the global optimum has been detected or that the "cost" connected with the search for a better estimate of the global optimum would be too high, or that some kind of "resource" has been exhausted, like, for example, computer time or number of function evaluations. In particular it is possible to distinguish some major classes of algorithms based on different choices of these steps; without pretending to be exhaustive, a tentative classification could be made as follows:

- Sampling step:
  Cardinality of the sample:
  - a fixed number of points per sample are generated
  - a sample is generated whose cardinality is sequentially and adaptively determined by the algorithm

  Sampling strategy
  - Points are drawn in A as independent, identically distributed (i.i.d.) random vectors (usually uniform on A)
  - Points are drawn in A as random vectors following a distribution whose support is a prescribed neighborhood of the current point (often uniform on a sphere or on a hypercube centered at the current point or normal centered at the current point with prescribed variance/covariance matrix)
  - New points are generated according to a distribution which depends explicitly on previously generated points and/or associated function values (not only on the current point)

- Optimization step:
  - no local search is performed
  - a local search is started from a selected number n of sampled points
  - a local search is started from each of the sampled points

- Stopping rule:
  - stopping occurs after a fixed number of steps or function evaluations
  - stopping occurs when no improvement has been reported in the last few iterations
  - stopping occurs when an *a posteriori* estimate of the probability that no unobserved local optimum exists exceeds a threshold
  - stopping occurs when an *a posteriori* measure of the expected benefit in continuing the algorithm (measured in terms of the trade-off between the possible gain connected with the discovery of a new local optimum

and the computational cost to be paid for such a discovery) falls below some threshold.

3. From Pure Random Search to Multistart

In this section some of the simplest random search-like algorithms will be briefly reviewed. These are the most basic schemes for global optimization and they lend themselves quite naturally to detailed theoretical analysis. Indeed some quite general result is available for classes of random search-like methods (most of them based on considerations related to the Borel–Cantelli lemma). Despite the apparent simplicity of the algorithms presented in this section, there are still some critical issues on the practical implementation for which satisfactory answers are still lacking; the most critical of this is the problem of stopping rules. On the other side, some of these simple algorithmic schemes, when implemented, have proven to be remarkably reliable, even when compared with more advanced and refined methods, like those which will be introduced in the successive sections. The material in this section is quite standard and is reported here only for completeness.

```
1. let $f^* := +\infty$
2. if a stopping criterion is satisfied then stop;
   otherwise let $x \sim \mathcal{U}(A)$ be a uniform random vector in $A$.
3. let $f^* = \min(f^*, f(x))$
4. go to 2
```

Fig. 1. Pure random search.

Pure random search is the easiest implementation of a Monte Carlo algorithm for global optimization. Its limited practical usefulness is mainly due to the fact that most of the information gathered during the execution of the algorithm is lost, as no use is made of function values and of function structure. On the other side it is the easiest algorithm to analyze and to implement (even, trivially, on a parallel computer). The algorithm shares with many other random search algorithms the basic weakness of consuming a lot of computational power in trying to obtain, through random exploration, points with relatively good function values. However, if the objective function is even moderately smooth, the task of improving over current function values is much more efficiently performed by local optimization routines; in other words, what seems to be lacking in most random search type algorithms is a clear distinction between the task of locally improving the function values and the task of exploration, which is peculiar to the random nature of the algorithms considered in this paper. For more details and references to this simple, algorithm refer to the cited book [39], or to other less recent surveys [33, 3].

```
1. let $f^* := +\infty$
2. if a stopping criterion is satisfied then go to 5
   otherwise let $x \sim \mathcal{U}(A)$ be a uniform random vector in $A$.
3. if $f^* > f(x)$ then let $f^* := f(x)$ and $x^* := x$
4. go to 2
5. Performing a local optimization starting from $x^*$
```

Fig. 2. Single start.

Single start differs from pure random search in that a single local optimization is performed at the end of the sampling procedure from the most "promising" point; again most of the information gained during the execution is lost.

```
1. let $f^* := +\infty$
2. if a stopping criterion is satisfied then stop;
   otherwise let $x \sim \mathcal{U}(A)$ be a uniform random vector in $A$.
3. perform a local optimization starting from $x$:
   let $\bar{x}$ be the local optimizer and $\bar{f} := f(\bar{x})$
4. let $f^* := \min(f^*, \bar{f})$
5. go to 2
```

Fig. 3. Multistart.

Multistart is, in some sense, on the opposite side of pure random search with respect to the use of local information. In this algorithm a local search routine is started from *each* sampled point. Let the local search routine be considered as a mapping $\mathcal{L}(\cdot)$ from A into itself which to each starting point x associates a local optimizer $\bar{x}$ (in which case x is said to belong to the "region of attraction" of $\bar{x}$, i.e. $\mathcal{L}^{-1}(\bar{x})$). Then Multistart can be seen as a pure random search applied to the piecewise constant function

$$F(x) = f \circ \mathcal{L}(x) = f(\mathcal{L}(x)) . \tag{2}$$

The main difference with respect to Pure Random Search is that here function evaluation is very expensive, so that the obvious disadvantage of Multistart is that much effort is spent in doing local searches which eventually lead to already discovered local minima; it is the purpose of clustering methods to try to overcome this difficulty, as will be seen in Section 5. Anyway, Multistart deserves consideration as its behaviour, when a sufficiently accurate stopping rule is used, is remarkably good despite the great simplicity of the algorithm itself. Thorough investigation on the behaviour of Multistart is reported in [43, 9, 6, 7, 8].

```
1. let $D: A \times \mathbb{R}^N \mapsto A$ be given
2. let $x \in A$
3. let $f^* := f(x))$
4. if a stopping criterion is satisfied then stop;
   otherwise let $F$ be a probability distribution function
5. let $\xi$ be a random vector sampled from $F$
6. let $x := D(x, \xi)$
7. let $f^* := \min(f^*, f(x))$
8. go to 4
```

Fig. 4. Random directions.

In the method of random directions, analyzed in the present form in [35], at each iteration a probability distribution function F is chosen, possibly as a function of the current point x, and a trial vector ξ (usually, but not necessarily, belonging to A) is generated according to such a distribution; the new iterate is now obtained through the application of a transformation $D: A \times \mathbb{R}^N \mapsto A$. Through different choices of the probability distribution and of the mapping D a whole range of stochastic algorithms is obtained. Notice, in particular, that choosing F to be the uniform distribution over A and D to be defined as $D(x, \xi) = \arg\min\{f(x), f(\xi)\}$, the pure random search is obtained, while the choice $D(x, \xi) = \mathscr{L}(\xi)$ corresponds to Multistart. In many variants of the basic scheme, the probability distribution F is chosen as the uniform distribution over a sphere centered at x with prescribed radius or as a gaussian distribution with mean x and a prefixed variance-covariance matrix. The name of the algorithm comes from the fact that x and ξ together identify a random line and, in many instances, the mapping D is chosen as some kind of line search along the direction $\xi - x$. In particular there are many variations of the basic algorithm in which

$$
\begin{aligned}
D(x, \xi) &= x + \alpha^*(\xi - x) \\
\alpha^* &= \arg\min_{\alpha} f(x + \alpha(\xi - x))
\end{aligned}
$$

the most commonly used of which are those in which α is constrained to range over one of the sets $[0, 1]$, $[0, 2]$, $[0, +\infty)$, or even the discrete set $\{0, 1\}$.

It is shown in [35] that converge in probability of the sequence generated by the algorithm to a set

$$R_\varepsilon = \{x \in A : f(x) < f^* + \varepsilon\}$$

for every choice of $\varepsilon > 0$ is achieved under the conditions that

- the mapping D ensures that no strict ascent steps are performed (i.e., $f(D(x, \xi)) \leq f(x))$, and
- for every Borel set B of positive (Lebesgue) measure, denoting by P_k the probability measure chosen at the kth iteration, then

$$\prod_{k=1}^{\infty} (1 - P_k(B)) = 0 ;$$

this way the possibility of repeatedly missing any Borel set of positive measure is ruled out.

Methods in this class generally are quite simple to implement, even if sensible stopping rules are very difficult to derive: in [35] the search for such stopping rules is dubbed as *doomed to fail*. Even if we are not so pessimistic, it should be evident that the choice of an appropriate stopping criterion is the crucial step in all of the algorithms discussed in this paper.

As a general remark on methods based on random directions, again it is to be stressed the fact that almost no use is made of the powerful theory and algorithms of local optimization: the methods thus appear to be well suited only for a special class of optimization problems characterized by the presence of a huge number of local optima and/or the unavailability of information on the derivatives of the objective function.

4. Simulated Annealing

Recently quite a lot of research effort has been devoted to variants of a Monte Carlo technique which corresponds to the simulation of the physical process of annealing, i.e. the process of driving a physical system to a minimal energy configuration by means of a slow reduction in the temperature of the system; if the "cooling" process is carried out sufficiently slowly, the system is allowed to skip over locally stable (minimum energy) configurations. Noticing the analogy between configurations of a physical system and feasible solutions of an instance of an optimization problem, and between the energy function and the cost to be optimized, in the early 80's two papers [24] and [13], independently opened the way to a large field of theoretical as well as applied research, mainly devoted to combinatorial optimization problems; for extensive surveys see [26] and [1]. Recently a comprehensive annotated bibliography appeared [15] on the theory and application of simulated annealing both to combinatorial and continuous optimization problems.

Apart from the analogy with the behaviour of physical systems, which, though appealing, can give at most an heuristic justification for the algorithm, the main characteristics of this approach lie in the possibility of escaping from the region of attraction of poor local optima by means of a controlled acceptance/rejection rule which selectively admits ascent steps in the course of the optimization process.

The general scheme of simulated annealing is sketched in Figure 5.

The crucial step in the algorithm is the decision whether or not to accept a new "configuration" y in place of the current one x: acceptance of the new configuration is made with probability

$$\min\{1, \exp\left(-\frac{f(y) - f(x)}{T}\right)\} \tag{3}$$

```
1. let $x \in A$
2. let $f^* := f(x)$
3. let $T > 0$ be the "initial temperature"
4. if a stopping criterion is satisfied then stop;
   otherwise do
   (a) if "equilibrium is reached" then exit this loop;
   (b) let $y$ be a random neighbor of $x$
   (c) let $U \sim \mathcal{U}([0,1])$ be a uniform random number in
       $[0,1]$
   (d) if $\exp(-(f(y)-f(x))/T) > U$ then let $x := y$
   (e) go to 4a
5. let $T$ be a new temperature value
6. go to 4
```

Fig. 5. Simulated annealing.

where T is the current value of the control parameter ("temperature"). This way every descent step is accepted, but it is also possible, albeit to a limited extent, to perform "up-hill" moves.

Several decisions have to be made in order to let the conceptual algorithm described in the figure become an implementable one; the steps involved are the following

- choice of an initial "temperature": a rule has to be given in order to obtain a starting value for the control parameter. Choosing a value which is too high slows down considerably the algorithm, while choosing a value too close to 0 will tend to exclude the possibility of ascent steps, thus losing the global optimization feature of the method.

- choice of an adequate stopping rule: as it has already been pointed out, this is almost always the crucial and most difficult part of the algorithm, and its definition may have dramatic influence on the overall performance.

- choice of a criterion for detecting equilibrium: this is some sort of another stopping rule; ideally, for each value of the control parameter, the inner loop of the conceptual algorithm should be executed infinitely often in order to let the "system reach equilibrium". More formally, as the sequence of points generated by the algorithm inside the inner loop can be seen as a realization of time-homogeneous Markov chain, then the execution of the loop is a simulation run of the Markov chain which, under ergodicity conditions, asymptotically reaches equilibrium.

  The choice at this step is intrinsically linked to the

- choice of the temperature decrement strategy: again, too fast the decrement, the higher the probability of being trapped in a poor local minimum; on the other side a slow decrement rule causes the algorithm to be unacceptably slow. Taking into account the outermost loop and considering the fact that the innermost loop is in practice executed a finite number of times, the whole sequence of points generated can be seen as a time inhomogeneous Markov chain. Based upon this characterization, several theoretical results concerning the convergence properties and finite time behaviour of simulated annealing have appeared in the literature (see the references in the cited books [26, 1] and more recent analyses in [40, 25, 14, 19]), as well as hints on the choice of the decrement rule given. From the point of view of the practical application of simulated annealing to optimization problems, the last, but not least, crucial step is the

- choice of an appropriate neighborhood structure. While general results are available for all of the critical issues discussed above, the way in which a "neighbor" y of the current solution x is defined is clearly problem-dependent. In continuous global optimization usually such a problem corresponds both to the generation of a direction and of a step length along such a direction.

From a theoretical point of view, the interest of simulated annealing for global optimization is easily motivated. In fact let the Boltzmann distribution be defined as

$$\pi^T(x) \propto \exp(-f(x)/T) \quad \forall x \in A \tag{4}$$

where $T > 0$ is the temperature; under mild assumptions (4) is easily seen to converge, as $T \downarrow 0$, to a uniform distribution over the set of global minimizers of f on A. Of course it is not feasible to sample directly from (4) (this would require the observation of f on every $x \in A$); when T is fixed, the Metropolis algorithm [29] can be employed, which consists in the simulation of a sample path from a Markov process (dealing, for simplicity, with a continuous time simulation) whose intensity is given by

$$R_T(x, y) = e^{[f(y)-f(x)]^+/T} S(x, y) \quad \forall x, y \in A \tag{5}$$

where S is any symmetric irreducible intensity matrix. It is immediately seen that the inner loop in Figure 5 is a discrete time implementation of this algorithm, with a prescribed temperature.

In the following we will present and briefly discuss the bibliography devoted to the study of the application of simulated annealing and similar techniques to the problem of *continuous* global optimization. The relative length of this section is by no means a measure of the author's confidence on the method, but is only due to the strong interest that the general approach of simulated annealing has risen in these last few years and to the relatively scarce discussion of its efficiency on continuous global optimization problems.

Vanderbilt and Louie, 1984

The algorithm introduced in [41], while retaining the basic scheme of Figure 5, generates random trial points according to a probability distribution which tries to take into account the local structure of the objective function. In particular,

$$y = x + Qu$$

where u is a random uniform vector on $[-\sqrt{3}, \sqrt{3}]^N$ (this way u's component are i.i.d. uniform random variables with 0 mean and unit variance), and Q is an appropriate transformation matrix. It is easily seen that the variance/covariance matrix S of the random step $y - x$ is related to the matrix Q by the relation

$$S = QQ^T .$$

The approach in [41] is to compute Q from an estimate of S based on the accepted trial points generated during the innermost loop in Figure 5; the estimator used for S is the sample variance/covariance matrix of the points accepted during a loop at constant temperature.

While the method appears to be interesting, as it tries to combine "intelligent" random search with the acceptance/rejection rule of simulated annealing, there are quite a lot of parameters which have to be adjusted in order for the algorithm to work and whose assessment is far from being easy; moreover the numerical results reported by the authors on a set of standard test functions for global optimization (see [18]) are not particularly encouraging: the number of function evaluations required until stopping is very high, and, in particular, it is consistently higher than that reported by [17] (which itself is nowadays outperformed by a variety of algorithms). Even more surprisingly, the reliability of the algorithm, when measured in terms of the estimated probability of detecting the global optima, computed over 100 independent runs of the algorithm over the same test function, is very poor: in 3 out of the 7 test functions the global optimum is discovered roughly 60% of the times; a simple Multistart algorithm equipped with a smart stopping rule in [6] was able to find the global optimum with 100% accuracy on the same set of test functions (which are indeed quite simple from the point of view of global optimization).

Bohachevsky et al., 1986

In [11] a quite simple implementation of simulated annealing is presented for continuous optimization. The approach followed is basically that of a random directions method, in which, at each step, a random (uniform) point is generated on the surface of a sphere centered in the current point with a prefixed radius. Then the most basic scheme of simulated annealing is applied, with the main differences being that the temperature decrement is, in some sense, autoregulating; that is the choice of the parameter T for the acceptance/rejection rule (3) is given (in our notation) by

$$T \propto (f(x) - f^*)^g \tag{6}$$

where $g > 0$. This way the parameter T is automatically driven to zero, with a speed regulated by g, as the point x approaches the global optimum. It is immediately clear that the implementation of the method requires the knowledge of the solution f^* itself. The authors state that, should the global optimum value not be known in advance, an estimate $\hat{f}^*$ could perhaps be given whose value is necessarily updated only when the current value $f(x)$ is lower than such an estimate, that is they propose to use

$$T \propto \max(0, f(x) - \hat{f}^*)^g .$$

Moreover, assuming knowledge of the global optimum, the stopping rule consists just in a verification of some distance criterion between the current best solution and the known optimum value.

The method, quite simple-minded, is severely invalidated by this strong assumption and also by the necessity (shared by most of the simulated annealing type algorithms) of setting a number of parameters whose "physical" meaning is a little evanescent and whose assessment is surely cumbersome. On the other side, the authors are successful in showing that a straightforward implementation of a random direction method in a simulated annealing setting is not very efficient, unless some modification to the acceptance/rejection rule is provided.

A slight modification of this algorithm appears in [12] where the temperature parameter is set to

$$T \propto f(x) \tag{7}$$

(compare with (6)). Again, the acceptance/rejection rule is made dependent on the actual value of the objective function, but there is no more explicit dependency on the global optimum value f^*. The performance of the method is displayed over the 7 classical test functions introduced in [18]; the numerical results look quite interesting, both for the good performance of the method, and for the reliability in detecting the global optimum. Unfortunately, also in this paper the global optimum value is used for the stopping criterion, so that the comparison with other algorithms which make no use of such an information, is strongly biased.

Corana et al., 1987

Also Corana *et al.* [16] propose a mixture of the standard Simulated Annealing scheme with a random search method. In their approach new candidate points are generated by perturbing the current one along a single coordinate direction; the perturbation is obtained through a uniform random variable whose support is chosen by the algorithm. In particular, the new candidate point y is given by

$$y = x + Ue_h \tag{8}$$

where e_h is the hth unit vector in $\mathbb{R}^N$ and U is a random variable uniformly distributed over the interval $[-v_h, v_h]$; the quantity $v_h > 0$ is determined in an adaptive way following the criterion that during the course of the inner loop (the one with the temperature held constant) the ratio of accepted *vs*. rejected moves should be approximately 1:1. This seems to be the original contribution of the paper, which otherwise consists in a quite standard implementation of the basic scheme of simulated annealing. However it is questionable whether such a ratio is indeed "optimal": usually such a ratio depends strongly on the temperature value, in that at "high" temperatures almost all moves are likely to be accepted, while the contrary happens at low temperatures. Moreover, the ratio of accepted-to-rejected moves is often used to detect the equilibrium condition necessary to exit the innermost loop in Figure 5 (see, e.g. [23]). It is clear that the desire of the authors is to try to adapt the range of exploration to the local form of the objective function, concentrating the sample around good local optima, but this could be also done in different, and more efficient ways, taking into account the local structure of f, e.g. by using second order information (see [41]).

Test results are provided in the paper, part of which based on the well-known Rosenbrock function, which is a classical test for *local* optimization; the appearance of such a test is again a proof of a basic misunderstanding upon which many random search algorithms are based, namely that the most efficient use of randomness lies in trying to approximate local optima. However we strongly feel that, being this a job very efficiently performed by local optimization routines, the correct place of randomness is in detecting regions of attractions of new local minima, and/or in providing confidence about the fact that the global optimum has already been found. This criticism is also supported by the numerical results displayed in the paper: the number of function evaluations required for a standard precision is a number of 6 or 7 figures, both in the Rosenbrock function test, as well as in other tests made with a multimodal function.

Piccioni, 1987

After an accurate analysis of the Pure Random Search algorithm, seen in a simulated annealing context, in [30] an algorithm is proposed whose acceptance/rejection rule is used in order to determine whether or not to jump (like the pure random search method) to a different point, randomly generated in A. Letting T_i, $i = 1, 2, \ldots$ be i.i.d. random variables exponentially distributed with parameter 1, define the time instants S_i, $i = 1, 2, \ldots$ by

$$S_i = \sum_{k=1}^{i} T_i \quad i = 1, 2, \ldots \tag{9}$$

Then a Markov process is generated in which at time instants S_i a random (uniform) point y is generated in A and an acceptance/rejection criterion of the type employed in Figure 5 is implemented; if the new configuration is accepted, a jump is made to y, otherwise the current configuration is not altered. The interesting point is that, differently from a pure random search/simulated annealing algorithm, here during the time interval $[S_i, S_{i+1})$ the process follows a deterministic descent process, performing steps in a direction which has a non-null component in the direction of the anti-gradient. This way the algorithms combines the features of a descent method with the controlled possibility of jumping to a different point in A at random time instants; this possibility is gradually made more difficult as the "temperature" of the system is lowered, as in the standard simulated annealing algorithm.

The author through a detailed spectral analysis of the mixed random search/simulated annealing method, supports the superiority of this scheme to those based on the Langevin equation (see next section). The approach seems very interesting; unfortunately, no numerical result is provided so a direct comparison with other methods is not possible.

Lucidi and Piccioni, 1989

In this paper [28] a general acceptance/rejection scheme is analyzed; in particular it is assumed that an algorithm for local optimization is available and that a randomization device is used in order to control whether or not to start a local search from a given (randomly generated) point in A. The scheme of the algorithm is reported in Figure 6.

At each step a new local search is started with probability $p_k(\cdot)$, which is indeed a conditional probability in which it is possible to include all of the information collected during the execution of the algorithm, i.e., all of the starting points X_k and, when a local search has been performed, the local optimizer Y_k.

Under mild assumptions on the non-null measure of the region of attraction of the global minimum and on the asymptotic behaviour of the acceptance probability p_k for $k \mapsto \infty$, it is shown that the algorithm finds the global optimum in a finite number of steps with probability one. However this result is common to most of the algorithms presented in this paper and does not provide a clear understanding of the actual behaviour of the algorithm; moreover the efficiency of an algorithm should be measured not (or not only) in terms of the number of steps necessary to *discover* the global optimum, but in terms of the number of steps required to *confirm* that the global optimum has indeed been observed. These considerations

```
1. let $k := 1$
2. let $X_k \sim \mathcal{U}(A)$ be a uniform random vector in $A$
3. let $U \sim \mathcal{U}([0,1)]$
4. if $U < p_k(X_k \mid X_1, Y_1, \ldots, X_{k-1}, Y_{k-1})$
     then let $Y_k := \mathcal{L}(X_k)$
     else let $Y_k := \emptyset$
5. if a stopping criterion is satisfied then stop
6. let $k := k + 1$
7. go to 2
```

Fig. 6. Lucidi and Piccioni.

are obviously generally applicable to the algorithms presented in this paper; the reader is again referred to [5] for a discussion on the fundamental issues of stopping rules.

The conceptual algorithm of figure 6 is modified in order to produce a reasonable sequence of acceptance probabilities; in particular, let f_k^* be defined as the best observation obtained after k steps; then it is suggested that the acceptance probabilities are defined as

$$p_k(x \mid \cdots) = \exp(-\max(0, f(x) - f_k^*)/T_k) \tag{10}$$

where T_k plays a role similar to that of the temperature parameter in the general simulated annealing scheme. It is to be noted however that, independently on the temperature schedule, the algorithm converges to the global optimum.

The algorithm, though quite simple, is very interesting, as it provides a stochastic version of the so-called Tunneling algorithm [27], which can be seen as a limiting case of the algorithm of Lucidi and Piccioni when the temperature is 0. The fundamental drawback of the deterministic tunneling algorithm is the practical equivalence between the problem of stopping the algorithm and the global optimization problem itself, which gives the algorithm the appeal of nothing more then a good heuristic; on the contrary, the statistical setting of the algorithm here discussed lends itself naturally to the design of stopping rules. Unfortunately the authors do not, for the moment, consider any stopping criteria, so that the practical implementation of the algorithm is still far from possible; even the numerical evidence reported is only of limited usefulness as it records the numbers of steps required to observe for the first time the (known) global optimum of the test functions presented. Further research is necessary in this field, but in any case it can be safely affirmed that this is the correct way of using Monte Carlo techniques in global optimization: as an extension to, and not as a substitute for, local optimization.

4.1. THE LANGEVIN EQUATION

The so-called Langevin equation in $\mathbb{R}^N$ takes the form

$$\mathrm{d}x(t) = -\nabla f(x(t))\,\mathrm{d}t + \sqrt{2T(t)}\,\mathrm{d}w(t) \tag{11}$$

where ∇f is the gradient of the function f, $T(t)$ the temperature at time $t \in [0, \infty)$ and $w(t)$ is the standard Brownian motion in $\mathbb{R}^N$. Equation (11) appeared as a generalization of the law of Brownian motion to the case of a particle moving in a viscous fluid. Again, apart form its physical interpretation, the equation can be seen, from our point of view of looking for stochastic optimization algorithms, as the law of motion of a point in $\mathbb{R}^N$ whose movement is subject to two different components: one is the tendency to follow down-hill trajectories along the direction of the anti-gradient $-\nabla f$; the other is a random fluctuation whose amplitude is governed by the temperature parameter $T(t)$.

The Langevin algorithm consists in simulating the Markov diffusion $x(\cdot)$ in (11) letting $T(t) \to 0$ as $t \to \infty$ by means of the algorithmic scheme

$$x_{k+1} = x_k - a_k \nabla f(x_k) + b_k W_k \quad k = 0, 1, \dots \tag{12}$$

where $\{W_k\}$ are i.i.d. standard gaussian random vectors with identity variance/covariance matrix, $a_k = \Delta t_k$ is a finite time increment and $b_k = \sqrt{2a_k T(t_k)}$ (see [19] for more details and references).

It appears evident that the Langevin algorithm is, similarly to simulated annealing, a stochastic descent method, in which some up-hill moves are allowed in order to escape from the region of attraction of local minima. The analogy with simulated annealing goes further if it is observed that both algorithms admit as invariant distribution the Boltzmann distribution (4) for every fixed $T > 0$. The mathematical elegance of the approach based on the Langevin equation has inspired good theoretical research in this field. See in particular [21, 20, 25, 14, 19] for various results on the convergence in probability of (12) to the set of global minima of f. Unfortunately, besides the formal mathematical beauty, in the cited papers no hint is given on the practical realization of implementable algorithms; nor, *a fortiori*, any analysis of the actual behaviour of the algorithms on standard test functions is provided. To our knowledge, the only relevant reference to an actual algorithm directly based on the Langevin algorithm is [2] which is analyzed next.

Aluffi-Pentini et al., 1985

In [2] the Cauchy problem

$$\begin{cases} \mathrm{d}x(t) = -\nabla f(x(t))\,\mathrm{d}t + \sqrt{2T(t)}\,\mathrm{d}w(t) \\ x(0) = x_0 \end{cases} \tag{13}$$

is considered, where the notation is as above. The idea is to integrate numerically (13) following the paths of the stochastic differential equation. Using the Euler–Cauchy discretization method, letting $\Delta t_i > 0$ and

$$t_0 = 0$$

$$t_k = \sum_{i=0}^{k-1} \Delta t_i \quad k = 1, 2, \dots$$

the solution $x(t_{k+1})$ is approximated by the finite difference equation

$$\begin{cases} x(t_{k+1}) = x(t_k) - \Delta t_k \nabla f(x(t_k)) + \sqrt{2T(t_k)}(w_{k+1} - w_k) \\ x(t_0) = x_0 \,. \end{cases} \tag{14}$$

As a practical detail, the authors suggest not to follow a single path of (13), which would require too long a simulation run, but to generate simultaneously a fixed number of independent trajectories (with the possibility of exploiting advanced computer hardware like, e.g., MIMD parallel machines). After a fixed number of steps have been performed, the "worst" trajectory is discarded, while another one is split into two different ones, which is easily accomplished thanks to the stochastic nature of the finite difference scheme (14). The newly generated path is assigned a different, usually lower, temperature $T(\cdot)$.

The performance of the algorithm is numerically investigated on several classical and original test problems; the overall reliability of the algorithm is remarkably good. The number of function evaluations, as it is common in algorithms of this kind, is quite high, but no astronomical numbers show up.

Comments on Simulated Annealing

As a general remark on all of the approaches to global optimization based upon simulated annealing, it seems worth to notice that a very high computational cost is generally required. It is difficult to state a precise measure of the "inefficiency" of simulated annealing as no extensive computational/theoretical comparison with more traditional methods is available. The overall implementation of these algorithms seems to be very difficult and the necessity of tuning several parameters to the objective function often shows up. It is therefore difficult to suggest such an approach as a general and reliable global optimization algorithm. Moreover too much folklore has risen around the physical meaning of "annealing", thus leading to much research constrained to a curious tentative of imitation of "nature" (a similar misunderstanding is to be mentioned with regard with another class of stochastic algorithms which does under the appealing name of "genetic algorithms" [22] and which merely consist of a variation of random search). Hoping not to look too simplistic, we can state that simulated annealing is just a randomization device that, by means of an acceptance/rejection criterion which adapts its parameters in the course of the algorithm, allows some ascent step during the optimization process. This fact is clearly recognized in some of the papers here presented (e.g., [28]). Much research is still needed in order to detect an appropriate blend of random sampling, descent algorithms and acceptance/rejection rules which leads to a practical, efficient, reliable and implementable algorithm.

Several variants of the basic scheme of simulated annealing have appeared in the literature. Recently [44] proposed an algorithm which is similar in spirit to that of simulated annealing, but differs in that ascent moves are never allowed, while an acceptance/rejection scheme is used in order to forbid steps which decrease the objective function too much; this way, by not descending too fast, the algorithm has time to concentrate on the global optimum. The algorithm may be interesting, but, while sharing some of the drawbacks of simulated annealing, adds the disadvantage of being easily trapped in the region of attraction of local minima. However it is interesting to notice that the description of such an algorithm applied to problems of stochastic approximation, appeared in [42], well before the first papers on simulated annealing.

5. Clustering Methods

Among the best performing methods for global optimizations are those which mix local search procedures with the application of clustering techniques aiming at grouping together (and thus identifying) points in A belonging to the region of attraction of the same local optimum; the methods in this class try to identify the shape and location of the regions of attraction of local optima in order to decide whether a local search started from a given point in A will eventually end up in a previously observed local optimum.

In this section a brief review of some clustering techniques will be presented; the material reported here is quite standard, as no significant improvement to clustering methods have been proposed in the literature in the last five years. The reasons for the decreasing interest in this approach are somewhat of a mystery, as the algorithms within this class are very well performing in practice, though several critical issues, both from a computational and from a theoretical point of view, still need to be addressed.

Most of the material here follows the development in the cited survey [39] and in [31, 32] (an almost untouched version of Timmer's Ph.D. thesis [36]).

The general scheme of clustering methods for global optimization can be represented in the following form (see [39]):

1. Sampling: draw at random a fixed number of points in A and observe the associated function values.

2. Concentration: transform the sample so that points belonging to regions of attraction of different local optima can be identified by the subsequent clustering analysis step; the most commonly used among the concentration techniques are those first introduced in [4] and [37]. In the former, the sample is reduced by eliminating a fixed percentage of points with higher function values; in the latter, a few steps of a descent algorithm are started from each sampled point.

3. Clustering: an appropriate clustering technique is employed in order to associate points to regions of attraction of local optima.

4. Stopping criterion: if some stopping condition is met, stop (possibly after performing some final computation, such a complete local optimization from selected points). Otherwise:

5. Transform the sample, retaining some or all of the points generated at step 2,

or retaining some of the "best" clusters as well as information on their shape, like, e.g., the radius of each cluster. Then repeat from step 1.

Clustering techniques are standard statistical methods aiming at grouping objects by means of a similarity criterion; often data can be represented as points in an Euclidean space and the similarity measure is taken to be the euclidean norm. A critical issue in implementing any kind of clustering technique is the choice of a threshold, which is used by the algorithm in order to decide if a point is "near" (i.e. within such a critical distance) to a representative point of the cluster (a so-called "seed" of the cluster).

The most commonly used clustering technique in the context of global optimization consists of partitioning the available observations into groups, sequentially assigning sampled points to clusters grown around "natural" seed points (in this framework natural seed points are usually local optimizers, or, in general, points with low function value). The simplest technique for deciding whether or not a new point should be added to a cluster is by means of comparison of the distance of such a point from the nearest seed point with a computed threshold: the clustering algorithm consists in building concentric hyperspheres, centered in a seed point, and by adding points to the cluster until the average density of points within the hypersphere is higher than the average density of sampled points. This clustering technique is used in [38]. In [34] clusters are grown around seed points based on the k-th nearest neighbor statistics; in this approach the critical distance is given by

$$\sqrt[n]{\beta_{j,1-\alpha}/\lambda} \quad j=1,\ldots,k$$

where $\beta_{j,1-\alpha}$ is the $(1-\alpha)$-quantile of the Beta distribution with parameters j and $n-j$ (n being the sample size), and $\alpha \in (0,1)$. In this as well as in most methods, the reduction technique of eliminating the worst points from the sample is used, despite the other one, based upon local descent, seems superior: the main reason for this is that this way, denoting by f^{**} the highest function value in the reduced sample, uniform distribution is still retained, but now on the level set

$$\mathscr{L}_{f^{**}} = \{x \in A : f(x) \leq f^{**}\} .$$

The choice of α in [34] roughly corresponds to the probability of misclassifying a point under the null hypothesis that the reduced sample is still uniformly distributed over the connected component of the level set $\mathscr{L}_{f^{**}}$.

In [10] a modification is proposed in which ellipsoidal-shaped clusters are grown instead of spherical ones: the idea is to try to approximate best the level sets of the objective function near local optima. In [36] a very simple criterion is used, by which a local search is not started from a point x if there exists another sampled point with lower function value within a critical distance given by

$$\sqrt{\pi}\sqrt[n]{\sigma\mu(A)\Gamma(1+N/2)\log k/k}$$

where σ is a constant and k represents the total number of sample/concentrate/cluster loops executed. The motivation for this formula is mainly heuristic, but it is easily proven that if $\sigma > 4$ the total number of local searches started by the algorithm is finite with probability one. Despite the simplicity of the approach and the lack of rigorous theoretical justification, the heuristic performs quite well in practice. The main difference between this criterion and that proposed in [34] is that in [10] thresholds are computed upon asymptotic considerations, whereas in [34] the exact finite sample distribution is considered.

As in most stochastic algorithms for global optimization, one of the most difficult and critical issues consists of choosing appropriate stopping rules; here the situation appears even more critical then elsewhere, as each time the stopping criterion is not met, a whole new sample with the associated local optimizations and clustering runs is started, so that late stopping produces very big amount of extra computation. Unfortunately still no satisfactory answer has been given on this very subject, which remains one of the most challenging research issues to be addressed in the context of global optimization.

Acknowledgement

This research has been partially supported by MURST Project 40% "Metodi di Ottimizzazione per le Decisioni".

References

1. Aarts, E. and Korst, J. (1989), *Simulated Annealing and Boltzmann Machines*, J. Wiley & Sons.
2. Aluffi-Pentini, F., Parisi, V., and Zirilli, F. (1985), Global Optimization and Stochastic Differential Equations, *J.O.T.A.* **47**, 1–16.
3. Archetti, F. and Schoen, F. (1984), A Survey on the Global Optimization Problem: General Theory and Computational Approaches, *Annals of Operations Research* **1**, 87–110.
4. Becker, R. W. and Lago, G. W. (1970), A Global Optimization Algorithm, in *Proceedings of the 8th Allerton Conference on Circuits and Systems Theory*, Monticello (Illinois), 3–12.
5. Betrò, B. (1990), Bayes Methods in Global Optimization, in B. Betrò, M. Gugiani and F. Schoen, *Monte Carlo Methods in Numerical Integration and Optimization*, Monografie di Matematica Applicata, CNR.
6. Betrò, B. and Schoen, F. (1987), Sequential Stopping Rules for the Multistart Algorithm in Global Optimisation, *Mathematical Programming* **38**, 271–286.
7. Betrò, B. and Schoen, F. (1988), *Optimal and Suboptimal Stopping Rules for the Multistart Method in Global Optimization*, report CNR-IAMI 88.11, Milano.
8. Betrò, B. and Schoen, F. (1990), A Stochastic Technique for Global Optimization, to appear in *Int. J. of Computers and Mathematics with Applications*.
9. Boender, C. G. E. and Rinnooy Kan, A. H. G. (1987), Bayesian Stopping Rules for Multistart Optimization Methods, *Mathematical Programming* **37**, 59–80.
10. Boender, C. G. E., Rinnooy Kan, A. H. G., Stougie, L., and Timmer, G. T. (1982), A Stochastic Method for Global Optimization, *Mathematical Programming* **22**, 125–140.
11. Bohachevsky, I. O., Johnson, M. E., and Stein, M. L. (1986), Generalized Simulated Annealing for Function Optimization, *Technometrics* **28**, 209–217.
12. Brooks, D. G. and Verdini, W. A. (1988), Computational Experience with Generalized Simulated Annealing over Continuous Variables, *American J. of Mathematical and Management Sciences* **8**, 425–449.

13. Černy, V. (1985), Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Šimulation Algorithm, *J.O.T.A.* **45**, 41–51.
14. Chiang, T.-S., Hwang, C.-R., and Sheu, S.-J. (1987), Diffusion for Global Optimization in R^n, *SIAM J. Control and Optimization in* **25**, 737–753.
15. Collins, N. E., Eglese, R. W., and Golden, B. L. (1988), Simulated Annealing – An Annotated Bibliography, *American J. of Mathematical and Management Sciences* **8**, 209–308.
16. Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987), Minimizing Multimodal Functions of Continuous Variables with the 'Simulated Annealing' Algorithm, *ACM Trans. Math. Software* **13**, 2–280.
17. De Biase, L. and Frontini, F. (1978), A Stochastic Method for Global Optimization: Its Structure and Numerical Performance, in *Towards Global Optimization 2*, L. C. W. Dixon and G. P. Szegö (eds.), North-Holland, Amsterdam, 85–102.
18. Dixon, L. C. W. and Szegö, G. P. (eds.) (1978), *Towards Global Optimization 2*, North-Holland, Amsterdam.
19. Gelfand, S. B. and Mitter, S. K. (1989), Simulated Annealing Type Algorithms for Multivariate Optimization, Technical Report LIDS-P-1845, M.I.T., Cambridge (MA).
20. Geman, S. and Hwang, C.-R. (1986), Diffusions for Global Optimization, *SIAM J. Control and Optimization* **24**, 1031–1043.
21. Gidas, B. (1985), Global Optimization via the Langevin Equation, *Proceedings of the 24th IEEE Conference on Decision and Control*, Ft. Lauderdale FL.
22. Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading (MA).
23. Huang, M. D., Romeo, F., and Sangiovanni-Vincentelli, A. (1986), An Efficient General Cooling Schedule for Simulated Annealing, *IEEE Int. Conf. Computer-Aided Design*, Santa Clara CA, 381–384.
24. Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M. P. (1982), Optimization by Simulated Annealing, *IBM Research Report RC 9355.*
25. Kushner, H. J. (1987), Asymptotic Global Behavior for Stochastic Approximation and Diffusions with Slowly Decreasing Noise Effects: Global Minimization via Monte Carlo, *SIAM J. Appl. Math.* **47**, 169–185.
26. van Laarhoven, P. J. M. and Aarts, E. H. L. (1987), *Simulated Annealing: Theory and Practice*, Reidel, Dordrecht.
27. Levy, A. V. and Montalvo, A. (1985), The Tunneling Algorithm for the Global Minimization of Functions, *SIAM J. Sci. Stat. Comput.*, **6**, 15–29.
28. Lucidi, S. and Piccioni, M. (1989), Random Tunneling by Means of Acceptance-Rejection Sampling for Global Optimization, *J.O.T.A.* **62**, 255–277.
29. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, M. (1953), Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics* **21**, 1087–1092.
30. Piccioni, M. (1987), A Combined Multistart-Annealing Algorithm for Continuous Global Optimization, Technical Research Report 87–45, Systems and Research Center, The University of Maryland, College Park MD.
31. Rinnooy Kan, A. H. G. and Timmer, G. T. (1987), Stochastic Global Optimization Methods. Part I: Clustering Methods, *Mathematical Programming* **39**, 27–56.
32. Rinnooy Kan, A. H. G. and Timmer, G. T. (1987), Stochastic Global Optimization Methods. Part II: Multi Level Methods, *Mathematical Programming* **39**, 57–78.
33. Rinnooy Kan, A. H. G. and Timmer, G. T. (1986), *Global Optimization*, report 8612/A, Erasmus University, Rotterdam.
34. Rotondi, R. (1987), A New Method for Global Optimization Based on the k-th Nearest Neighbor, CNR-IAMI report, Milano.
35. Solis, F. J. and Wets, R. J.-B. (1981), Minimization by Random Search Techniques, *Mathematics of Operations Research* **6**, 19–30.
36. Timmer, G. T. (1984), Global Optimization: A Stochastic Approach, Ph.D. thesis, Erasmus University, Rotterdam.
37. Törn, A. (1973), Global Optimization as a Combination of Global and Local Search, *Gothenburg Business Adm. Studies* **17**, 191–206, 1973.

38. Törn, A. (1977), Cluster Analysis Using Seed Points and Density Determined Hyperspheres as an Aid to Global Optimization, *IEEE Trans. Syst. Men and Cybernetics* **7**, 610–616.
39. Törn, A. A. and Žilinskas, A. (1989), *Global Optimization*, Springer-Verlag.
40. Tsitsiklis, J. N. (1986), A Survey of Large Time Asymptotics of Simulated Annealing Algorithms, Technical Report LIDS-P-1623, M.I.T., Cambridge (MA).
41. Vanderbilt, D. and Louie, S. G. (1984), A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables, *Journal of Computational Physics* **56**, 259–271.
42. Zieliński, Global Stochastic Approximation: A Review of Results and Some Open Problems, in *Numerical Techniques for Stochastic Systems*, North-Holland, 379–386, 1980.
43. Zieliński, R. (1981), A Statistical Estimate of the Structure of Multiextremal Problems, *Mathematical Programming* **21**, 348–356.
44. Zieliński, G. S. A. *vs.* S. R. A., private communication, 1990.