Simulated Annealing for Constrained Global Optimization*

H. EDWIN ROMEIJN
Rotterdam School of Management, Erasmus University Rotterdam, Rotterdam, The Netherlands

and

ROBERT L. SMITH
Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109-2117, U.S.A.

(Received: 24 August 1992; accepted: 7 October 1993)

Abstract. Hide-and-Seek is a powerful yet simple and easily implemented continuous simulated annealing algorithm for finding the maximum of a continuous function over an arbitrary closed, bounded and full-dimensional body. The function may be nondifferentiable and the feasible region may be nonconvex or even disconnected. The algorithm begins with any feasible interior point. In each iteration it generates a candidate successor point by generating a uniformly distributed point along a direction chosen at random from the current iteration point. In contrast to the discrete case, a single step of this algorithm may generate *any* point in the feasible region as a candidate point. The candidate point is then accepted as the next iteration point according to the Metropolis criterion parametrized by an *adaptive* cooling schedule. Again in contrast to discrete simulated annealing, the sequence of iteration points converges in probability to a global optimum regardless of how rapidly the temperatures converge to zero. Empirical comparisons with other algorithms suggest competitive performance by Hide-and-Seek.

Key words. Continuous simulated annealing, adaptive cooling, random search, global optimization, Monte Carlo optimization

1. Introduction

Consider the following constrained global optimization problem:

$$\max f(x)$$

$$\text{subject to } x \in S \qquad (P)$$

where S is a compact body in $\mathbb{R}^d$, i.e., S is a nonempty, bounded subset of $\mathbb{R}^d$ equal to the closure of its interior, the boundary of S has Lebesgue measure equal to zero, and f is a continuous real-valued function defined on S. In other words, the problem is to find a point $x^* \in S$ such that

$$f^* \equiv f(x^*) \geqslant f(x)$$

* This material is based on work supported by a NATO Collaborative Research Grant, no. 0119/89.

Journal of Global Optimization **5**: 101–126, 1994.

for all $x \in S$. Note that we do not require that f be differentiable nor that S be convex or even connected.

Restricting ourselves to stochastic methods for solving global optimization problems, we can divide the methods for solving versions of (P) into two classes:

- Two-phase methods
- Random search methods

Algorithms from the first class start by generating a random (usually uniform) sample of points in the feasible region S. Then, a local search procedure is applied to all (or a subset of the) points in this sample. If necessary, this procedure is repeated. The best local optimum found is used as the estimate of the global optimum. Examples of methods from this class are: (i) Multistart, for *constrained* global optimization (see Rinnooy Kan and Timmer, 1984); (ii) Multi Level Single Linkage, for *unconstrained* global optimization (see Rinnooy Kan and Timmer, 1987a,b).

The second class consists of algorithms which generate a sequence of points in the feasible region following some prespecified probability distribution (which can change during the course of the algorithm). Examples of methods from this class are Pure Random Search (see e.g. Rinnooy Kan and Timmer, 1984), Pure Adaptive Search (see Patel *et al.*, 1988, and Zabinsky and Smith, 1992), and methods based on Simulated Annealing.

Simulated Annealing is a sequential search technique that avoids getting trapped in local maxima by accepting, in addition to transitions corresponding to an increase in function value, transitions corresponding to a *decrease* in function value. The latter is done in a limited way by means of a probabilistic acceptance criterion. In the course of the maximization process, the probability of accepting deteriorations descends slowly towards zero. These "deteriorations" make it possible to move away from local optima and explore the feasible region S in its entirety.

Simulated Annealing originated from an analogy with the physical annealing process of finding low energy states of a solid in a heat bath (see Metropolis *et al.*, 1953). Pincus (1970) developed an algorithm based on this analogy for solving discretizations of continuous global optimization problems. Most of the other applications to date have been to discrete combinatorial optimization problems (see e.g. Kirkpatrick *et al.*, 1983, Aarts and Korst, 1988, and Aarts and Van Laarhoven, 1989). Formulations of the Simulated Annealing algorithm for continuous optimization have also been proposed (see e.g. Bohachevsky *et al.*, 1986, Corana *et al.*, 1987, and Dekker and Aarts, 1991). Many of these formulations however, with the notable exception of the last one, lack theoretical foundations.

One of the principal problems in the practical implementation of Simulated Annealing is the choice of a cooling schedule for the temperature parameter (which parametrizes the decrease of the acceptance probabilities for deteriorations as described above). For discrete simulated annealing Hajek (1988) has provided necessary and sufficient conditions for a cooling schedule that guarantee convergence to the global optimum, for the case of *a deterministic* cooling schedule, i.e., when the sequence of temperatures is known in advance. In this paper we derive an *adaptive* cooling schedule for Simulated Annealing for continuous optimization, where the temperatures employed depend upon the real-time progress of the algorithm. This algorithm, called Hide-and-Seek, was introduced in Romeijn and Smith (1990). Bélisle (1992) proved that, under this cooling schedule, convergence to the global optimum of (P) is guaranteed. It is interesting that in contrast to the discrete case, convergence in this continuous case is guaranteed regardless of how rapidly the sequence of temperatures converges to zero. This is an important distinction since the required rate of cooling in the discrete case is rarely computable in practice, thus imparting a heuristic character to the algorithm in that case.

The paper is organized as follows. In Section 2 we introduce a general class of random walks for generating a sequence of feasible points having the property that its limiting distribution equals a prespecified target distribution π. Again in contrast to generators in the discrete case that typically restrict moves to neighboring lattice points, this continuous walk can span the entire feasible region in a single step. These random walks are similar in spirit to the Hit-and-Run algorithms discussed in Smith (1984), Berbee *et al.* (1987), and Bélisle *et al.* (1993), as well as the Shake-and-Bake algorithms introduced in Boender *et al.* (1991). What is different is the use of an acceptance criterion to insure reversibility with respect to the limiting distribution π. In Section 3 we present the Hide-and-Seek algorithm. It is based on the idea of running the random walk of Section 2 with a target distribution that concentrates around the global optimum. An adaptive cooling schedule is derived and the algorithm is compared to other versions of simulated Annealing for continuous optimization. We conclude in Section 4 with a comparison of the performance of various versions of Hide-and-Seek with each other and with methods from the literature on a set of standard test problems.

2. Reversible Random Walks

2.1. A CLASS OF RANDOM WALKS

Let S be a compact body in $\mathbb{R}^d$ with boundary of Lebesgue measure zero. Let g be a strictly positive continuous density on S and set the *target distribution* π to be the absolutely continuous probability measure with density g, that is,

$$\pi(A) = \int_A g(x)\,\mathrm{d}x \quad \forall A \in \mathcal{B}$$

where $\mathcal{B}$ denotes the Borel σ-field on S. Let β be a function on S^2 satisfying the following conditions:

(i) $0 < \delta \leqslant \beta(x, y) \leqslant 1$ for all $x, y \in S$, for some $\delta > 0$.
(ii) $\beta(x, y)$ is jointly measurable on S^2.
(iii) $\beta(x, y)g(x) = \beta(y, x)g(y)$ for all $x, y \in S$.

We will call β the *acceptance probability function*, for reasons that will become clear later. Finally, let D denote the d-dimensional unit sphere centered at the origin, let ∂D be its topological boundary, and let ν be an absolutely continuous probability measure on ∂D, with density h with respect to the uniform distribution on ∂D which is uniformly bounded away from zero. Consider the following

Random Walk Algorithm
Step 0: Choose a starting point x_0 in the interior of S and set $k = 0$.
Step 1: Choose a direction θ_k on ∂D with distribution ν.
Step 2: Choose λ_k from the uniform distribution on $\Lambda_k = \{\lambda \in \mathbb{R} : x_k + \lambda\theta_k \in S\}$.
Step 3: *Set* $y_{k+1} = x_k + \lambda_k\theta_k$. With probability $\beta(x_k, y_{k+1})$, set $x_{k+1} = y_{k+1}$. Otherwise, set $x_{k+1} = x_k$.
Step 4: Set $k = k + 1$. Go to step 1.

Our assumptions on the region S and on the direction distribution ν imply that, with probability 1, the uniform distribution in step 2, and therefore the random walk, is well defined.

The random walk generates three random sequences: the direction vectors $(\Theta_n; n \geqslant 0)$, the candidate points $(Y_n; n \geqslant 1)$, and the iteration points $(X_n; n \geqslant 0)$. Our main interest is in the sequence of iteration points. Clearly $(X_n; n \geqslant 0)$ is a Markov chain with state space S and with stationary transition probabilities. We will show that the target distribution π is a stationary distribution for this Markov chain and moreover that for every starting point x_0 in S the Markov chain converges in total variation and hence in distribution to π.

Note that if $\beta(x, y) = 1$ for all $x, y \in S$, then π is the uniform distribution on S and the random walk becomes the Hit-and-Run algorithm. In that case stationarity of π and convergence of X_n to π are known. (See Smith, 1984, Berbee *et al.*, 1987, or Bélisle *et al.*, 1993).

2.2. REVERSIBILITY AND STATIONARITY

Let us recall some standard definitions. A *measurable space* is a pair $(S, \mathscr{B})$ where S is an arbitrary set and where $\mathscr{B}$ is an arbitrary σ-field on S. A *Markov kernel* on $(S, \mathscr{B})$ is a nonnegative function P defined on $S \times \mathscr{B}$ and such that

(i) for all $x \in S$, $P(x, \cdot)$ is a probability measure on $\mathscr{B}$,
(ii) for all $A \in \mathscr{B}$, $P(\cdot, A)$ is a measurable function of S (see, e.g., Nummelin, 1984).

A Markov kernel P is said to be *reversible* with respect to a probability measure π if

$$\int_A P(x, B)\pi(\mathrm{d}x) = \int_B P(x, A)\pi(\mathrm{d}x) \quad \forall A, B \in \mathcal{B}.$$

A probability measure π on $(S, \mathcal{B})$ is said to be *stationary* for the Markov kernel P if

$$\int_S P(x, B)\pi(\mathrm{d}x) = \pi(B) \quad \forall B \in \mathcal{B}.$$

The following preliminary lemmas pertain to arbitrary Markov kernels on general measurable spaces.

LEMMA 2.1. *Suppose that a Markov kernel P is of the form*

$$P(x, A) = \int_A p(x, y)\pi(\mathrm{d}y) + 1_A(x)\Big(1 - \int_S p(x, y)\pi(\mathrm{d}y)\Big)$$

for all $x \in S$, $A \in \mathcal{B}$, for some jointly measurable symmetric *function $p(x, y)$ on S^2 and some probability measure π on $(S, \mathcal{B})$, where symmetric means $p(x, y) = p(y, x)$ $\forall x, y \in S$ and $1_A(\cdot)$ is the indicator function of the set A. Then P is reversible with respect to π.*

Proof. Under the given assumptions, an application of Fubini's theorem yields:

$$\begin{aligned}
&\int_A P(x, B)\pi(\mathrm{d}x) = \\
&= \int_A \Big(\int_B p(x, y)\pi(\mathrm{d}y) + 1_B(x)\cdot\Big(1 - \int_S p(x, y)\pi(\mathrm{d}y)\Big)\Big)\,\pi(\mathrm{d}x) \\
&= \int_A \int_B p(x, y)\pi(\mathrm{d}y)\pi(\mathrm{d}x) + \int_A 1_B(x)\cdot\Big(1 - \int_S p(x, y)\pi(\mathrm{d}y)\Big)\,\pi(\mathrm{d}x) \\
&= \int_B \int_A p(y, x)\pi(\mathrm{d}x)\pi(\mathrm{d}y) + \int_B 1_A(y)\cdot\Big(1 - \int_S p(y, x)\,\pi(\mathrm{d}x)\Big)\pi(\mathrm{d}y) \\
&= \int_B \Big(\int_A p(y, x)\pi(\mathrm{d}x) + 1_A(y)\cdot\Big(1 - \int_S p(y, x)\pi(\mathrm{d}x)\Big)\Big)\,\pi(\mathrm{d}y) \\
&= \int_B P(y, A)\,\pi(\mathrm{d}y)
\end{aligned}$$

for all $A, B \in \mathcal{B}$. Thus P is reversible with respect to π. □

LEMMA 2.2. *Suppose that a Markov kernel P is reversible with respect to a probability measure π. Then π is a stationary distribution for P.*

Proof. Using reversibility we obtain

$$\begin{aligned}\int_S P(x, A)\,\pi(\mathrm{d}x) &= \int_A P(x, S)\pi(\mathrm{d}x)\\ &= \int_A \pi(\mathrm{d}x)\\ &= \pi(A)\end{aligned}$$

for all $A \in \mathscr{B}$. Thus π is stationary for P. □

We now return to the Markov chain $(X_n; n \geqslant 0)$ of the random walk of Section 2.1 and we write

$$P(x, A) = \Pr[X_{n+1} \in A \mid X_n = x] \quad x \in S,\ A \in \mathscr{B}.$$

The following proposition provides an explicit expression for this Markov kernel.

PROPOSITION 2.3. *The Markov kernel associated with the random walk of Section 2.1 can be written as*

$$P(x, A) = \int_A p(x, y)\pi(\mathrm{d}y) + 1_A(x)\Big(1 - \int_S p(x, y)\pi(\mathrm{d}y)\Big)$$

for all $x \in S$, $A \in \mathscr{B}$, where π is the target distribution,

$$p(x, y) = \frac{(h((y-x)/\|y-x\|) + h((x-y)/\|y-x\|))\beta(x, y)}{C_d\|x-y\|^{d-1}d(x, y)g(y)},$$

C_d denotes the surface area of ∂D, and $d(x, y)$ is the diameter of S along the line through x and y.

Proof. analogous to the proof of Proposition 3(b) of Bélisle *et al.* (1993). Note that the first part of the expression for $P(x, A)$ is the probability of *moving* from x to a point in A, and the second part of the expression is the probability of *staying* at x. □

PROPOSITION 2.4. (a) *The Markov kernel P associated with the random walk of Section* 2.1 *is reversible with respect to the target distribution π.*

(b) *The target distribution π is a stationary distribution for the Markov kernel P.*

Proof. Part (a) follows from Lemma 2.1 and Proposition 2.3. Part (b) follows from Lemma 2.2 and part (a).

2.3. CONVERGENCE TO THE TARGET DISTRIBUTION

We now can show that the sequence $(X_n; n \geqslant 0)$ of iteration points of the reversible random walk of Section 2.1 converges to the target distribution π.

THEOREM 2.5. *For every starting point x_0 in S, the Markov chain $(X_n;\ n \geqslant 0)$*

defined by the reversible random walk algorithm of Section 2.1 *converges in total variation to the target distribution* π, *i.e.*

$$\lim_{n\to\infty} \Pr[X_n \in B \mid X_0 = x_0] = \pi(B) \quad \forall x_0 \in S, \quad \forall B \in \mathscr{B}.$$

Proof. Using the representation of $P(x, A)$ given in Proposition 2.3, Theorem 2.5 follows easily via Condition (D′) in Doob (1953), Chapter V, Section 5, case (b). □

Although the Markov chain $(X_n; n \geq 0)$ will be restricted in this paper to generating the Boltzman distribution for use in a simulated annealing context, Theorem 2.5 validates its use as a Monte Carlo procedure for generating samples from *any* absolutely continuous distribution π. Its use of an acceptance criterion β for providing reversibility with respect to π can be significantly more efficient for peaked distributions (such as the Boltzman family for low temperatures), then the device of conditionalization utilized in Bélisle *et al.* (1993).

3. The Hide-and-Seek Algorithm

3.1. HIDE-AND-SEEK

We now return to the global optimization problem posed in Section 1:

$$\max_{x\in S} f(x)$$

where S is a compact body in $\mathbb{R}^d$ whose boundary has Lebesgue measure zero and where f is continuous on S. Following Pincus (1968), Rubinstein (1981) suggests for approximating the global optimum that we generate points from the distribution π_T with density

$$g_T(x) = \frac{e^{f(x)/T}}{\int_S e^{f(z)/T}\,dz} \tag{1}$$

where T is a "small" positive number. This is appropriate because for small T the distribution π_T will "concentrate near the global maximum". To make this idea precise, for $\varepsilon > 0$, define the ε-level set as follows:

$$S_\varepsilon = \{x \in S : f(x) \geq f^* - \varepsilon\}$$

where f^* is, as before, the global maximum of f on S. Then we have

PROPOSITION 3.1. *For all* $\varepsilon > 0$,

$$\lim_{T\to 0} \pi_T(S_\varepsilon) = 1\,. \tag{2}$$

Proof. Fix $\varepsilon > 0$. Then

$$\begin{aligned}\pi_T(S_\varepsilon) &= 1 - \pi_T(S - S_\varepsilon) = 1 - \frac{\int_{S-S_\varepsilon} \mathrm{e}^{f(x)/T}\,\mathrm{d}x}{\int_S \mathrm{e}^{f(z)/T}\,\mathrm{d}z} \\ &\geqslant 1 - \frac{\int_{S-S_\varepsilon} \mathrm{e}^{f(x)/T}\,\mathrm{d}x}{\int_{S_{\varepsilon/2}} \mathrm{e}^{f(z)/T}\,\mathrm{d}z} \geqslant 1 - \frac{\mathrm{e}^{(f^*-\varepsilon)/T}\varphi(S - S_\varepsilon)}{\mathrm{e}^{(f^*-(\varepsilon/2))/T}\varphi(S_{\varepsilon/2})} \\ &= 1 - \mathrm{e}^{-\varepsilon/(2T)}\frac{\varphi(S - S_\varepsilon)}{\varphi(S_{\varepsilon/2})}\end{aligned}$$

where φ denotes the Lebesgue measure on $\mathbb{R}^d$. (Note that our assumptions on S and f guarantee that $\varphi(S_{\varepsilon/2}) > 0$.) Thus $\lim_{T\to 0} \pi_T(S_\varepsilon) = 1$. □

REMARKS ON PROPOSITION 3.1.

1. Pincus (1968) has shown that if the global maximum x^* of f is unique, then the mean of the distribution π_T converges to x^* as T converges to 0. This result also follows easily from Proposition 3.1.
2. Proposition 3.1 suggests that, as T converges to 0, π_T converges weakly to a distribution that concentrates on the set

   $$S^* = \{x \in S : f(x) = f^*\}\,.$$

   If S^* consists of a single point x^*, then Proposition 3.1 directly implies that π_T converges weakly to the point mass at x^*. If S^* has positive Lebesgue measure then it is also easy to show that π_T converges weakly to the uniform distribution on S^*. In the general case, weak convergence of π_T is a very delicate issue. See Hwang (1980).

Now fix $T > 0$ and hence the target distribution π_T. The random walk of Section 2.1 can, by Theorem 2.5, be used to generate a sequence $(X_k(T);\ k \geqslant 0)$ with the property that for every $\varepsilon > 0$

$$\lim_{k\to\infty} \Pr[X_k(T) \in S_\varepsilon] = \pi_T(S_\varepsilon)\,. \tag{3}$$

A choice consistent with properties (i) through (iii) of Section 2.1 for the acceptance probability function β is the so-called Metropolis criterion, given by

$$\beta_T(x, y) = \min\{1, g_T(y)/g_T(x)\}$$

which, after substitution of (1), yields the acceptance probability

$$\beta_T(x, y) = \min\{1, \mathrm{e}^{(f(y)-f(x))/T}\} \tag{4}$$

of classical simulated annealing. In adherence with simulated annealing terminology, T will be called the *temperature parameter*. Combining (2) and (3) we get

$$\lim_{T\to 0}\lim_{k\to\infty} \Pr[X_k(T) \in S_\varepsilon] = 1 \quad \forall \varepsilon > 0\,. \tag{5}$$

The Hide-and-Seek algorithm is motivated by equation (5). It consists of generating a sequence $X_0, X_1, X_2, \ldots$ using the random walk algorithm of Section 2.1 with target distribution π_T with density $g_T(x)$ given in (1), with direction distribution ν, and with acceptance probability function $\beta_T(x, y)$ given by the Metropolis criterion (4), except that now the temperature parameter T will decrease to 0 as we proceed. The cooling schedule corresponds to a sequence of temperatures $(T_k;\, k \geqslant 0)$ that are *adpative*, i.e. for each $k \geqslant 0$, T_k is a function of $X_0, X_1, \ldots, X_k$. More formally, the *cooling schedule* is a sequence $t_0, t_1, t_2, \ldots$ where, for each k, t_k is a jointly measurable function from S^{k+1} to $[0, \infty]$. The temperature T_k is then given by $T_k = t_k(X_0, X_1, \ldots, X_k)$. In Section 3, we will offer a specific cooling schedule. For now we simply require that the temperatures decrease to 0 in probability as $k \to \infty$.

A formal mathematical construction of the Hide-and-Seek algorithm $(X_n;\, n \geqslant 0)$ can be achieved as follows. We are given a compact body S in $\mathbb{R}^d$, a continuous function f defined on S, a starting point $x_0 \in S$ and a cooling schedule $(t_k;\, k \geqslant 0)$. On an appropriate probability space we construct a sequence $(\Theta_k;\, k \geqslant 0)$ of i.i.d. random variables with distribution ν on ∂D, and two sequences, say $(U_k;\, k \geqslant 0)$ and $(V_k;\, k \geqslant 0)$, of i.i.d. random variables with uniform distribution on $[0, 1]$. The three sequences are independent of each other. The *Hide-and-Seek Algorithm* $(X_k;\, k \geqslant 0)$ is defined by setting $X_0 = x_0$ and by proceeding recursively: Given $X_0, X_1, \ldots, X_k$, set

$$Y_{k+1} = X_k + \lambda_k \Theta_k$$

with $\lambda_k = F^{-1}_{(X_k, X_k+\Theta_k)}(U_k)$, where $F_{(a,b)}$ is the c.d.f. of the uniform distribution over the set $\Lambda(a, b) = \{\lambda \in \mathbb{R} \colon a + \lambda(b - a/\|b - a\|) \in S\}$, and $F^{-1}_{(a,b)}$ is the usual right continuous inverse of $F_{(a,b)}$. Finally, set

$$X_{k+1} = \begin{cases} Y_{k+1} & \text{if } V_k \in [0, \beta_{T_k}(X_k, Y_{k+1})] \\ X_k & \text{if } V_k \in (\beta_{T_k}(X_k, Y_{k+1}), 1] \end{cases}$$

where $T_k = t_k(X_0, X_1, \ldots, X_k)$.

3.2. CONVERGENCE OF HIDE-AND-SEEK

Although successive iterations of Hide-and-Seek may experience deteriorations in objective function value, the following theorem, due to Bélisle (1992) states that, roughly speaking, these effects are transient and Hide-and-Seek will eventually be absorbed in arbitrarily small neighborhoods of the global maximum regardless of the rate of cooling.

THEOREM 3.2 (cf. Bélisle, 1992). *Consider the sequence $(X_n;\, n \geqslant 0)$ generated by the Hide-and Seek Algorithm using an adaptive cooling schedule $(T_n;\, n \geqslant 0)$. Assume that for every starting point x_0, the cooling schedule converges to 0 in* *probability. Then the sequence of function values* $(f(X_n);\ n \geqslant 0)$ *converges in probability to the global maximum* f^*. *That is, for all* $\varepsilon > 0$ *and* $x_0 \in S$,

$$\Pr[f(X_n) < f^* - \varepsilon \mid X_0 = x_0] \to 0 \text{ as } n \to \infty .$$

Theorem 3.2 is perhaps somewhat surprising in that the discrete case notion of "depth" and corresponding restriction on the cooling schedule (see Hajek, 1988) play no role in this continuous case. The fundamental difference here is due to the fact that Hide-and-Seek executes *global*, as oppose to local, reaches over S within each iteration.

In practice one would keep track of the current record value

$$f_n^* = \max_{0 \leqslant k \leqslant n} f(X_k) .$$

The following corollary is an immediate consequence of the theorem.

COROLLARY 3.3. *In Hide-and-Seek,*

$$f_n^* \to f^* \text{ almost surely, as } n \to \infty .$$

The convergence rate of the algorithm to the global optimum is of course dependent on the way in which the temperature is lowered to zero. The special case where the temperature is fixed at 0, reduces to the Improving Hit-and-Run algorithm (see Zabinsky *et al.*, 1993). Incidentally, if the temperature were fixed at the other extreme of ∞, we would have ordinary Hit-and-Run with the uniform distribution as its target distribution. This would effectively correspond to so-called pure random search. In the following section we derive a specific cooling schedule which begins with Hit-and-Run and asymptotically approaches Improving Hit-and-Run.

3.3. A HEURISTICALLY MOTIVATED ADAPTIVE COOLING SCHEDULE

In Patel *et al.* (1988), Pure Adaptive Search is introduced. The algorithm consists of generating a sequence of points with the property that each successor point is conditionally uniformly distributed in the region of points with superior objective function values. In Zabinsky and Smith (1992), it is proven that Pure Adaptive Search exhibits linear time computational complexity in dimension for the class of global optimization problems whose objective function possesses a finite Lipschitz constant and whose feasible region is convex. However, the algorithm was posed purely as a theoretical construct, since no efficient implementation was known.

Our choice of cooling schedule is motivated by the goal of approximating Pure Adaptive Search via Hide-and-Seek. Specifically, we choose the next temperature so that a point generated according to the limiting distribution π_T of Hide-and-Seek under that temperature is very likely to be in the region of points with superior objective function value. Of course, π_T is not conditionally uniform on this region; however it promises to be stochastically superior in value to uniform.

Therefore, under this cooling schedule, we hope to inherit the favorable complexity properties exhibited by Pure Adaptive Search. Preliminary empirical evidence offers support for this hope as we will see in Section 4.

We begin by considering a second-order Taylor series approximation of the objective function f at a point $x_0 \in S$. We assume that the point x_0 is near a local maximum and that the corresponding Hessian matrix H is a positive definite matrix. In particular, we approximate f by

$$f(x) = -(x - x_0)'H(x - x_0) + 2r'(x - x_0) + f(x_0)$$

where H is a $d \times d$-dimensional positive definite matrix and r is a $d \times 1$ vector. Suppose now for convenience and without loss of generality for the argument to follow that $x_0 = 0$ and $f(x_0) = 0$, and assume that S is a level set of this approximation to f. Our optimization problem becomes

$$\max_{x \in S} f(x)$$

where $f(x) = -x'Hx + 2r'x$ and $S = \{x \in \mathbb{R}^d : f(x) \geqslant f^* - c\}$ where $c \in \mathbb{R}^+$. Note that we can rewrite the objective function as follows:

$$\begin{aligned} f(x) &= r'H^{-1}r - (x - H^{-1}r)'H(x - H^{-1}r) \\ &= f^* - (x - H^{-1}r)'H(x - H^{-1}r)\,. \end{aligned}$$

The expression for the feasible region now reduces to

$$S = \{x \in \mathbb{R}^d : (x - H^{-1}r)'H(x - H^{-1}r) \leqslant c\}\,.$$

Given an iteration point x, we will choose the temperature T in such a way that generating a point from the exact distribution π_T would give an improvement in function value over the current iteration point with probability at least p. As an alternative notation for the level sets of f, write:

$$\mathcal{S}_x = \{z \in S : f(z) > f(x)\} = S_{f^* - f(x)}\,.$$

Let Y be a random variable with distribution π_T. We then have:

$$\Pr\{f(Y) > f(x)\} = \frac{\int_{\mathcal{S}_x} \mathrm{e}^{-f(z)/T}\, \mathrm{d}z}{\int_S \mathrm{e}^{-f(z)/T}\, \mathrm{d}z} \geqslant \frac{\int_{\mathcal{S}_x} \mathrm{e}^{-f(z)/T}\, \mathrm{d}z}{\int_{\mathbb{R}^d} \mathrm{e}^{-f(z)/T}\, \mathrm{d}z}\,.$$

Now perform the transformation $u = \sqrt{2/T}H^{1/2}(z - H^{-1}r)$ in both the numerator and the denominator. We then get:

$$\Pr\{f(Y) > f(x)\} \geqslant \frac{\int_{\mathcal{S}'_x} \mathrm{e}^{-u'u/2}\, \mathrm{d}u}{\int_{\mathbb{R}^d} \mathrm{e}^{-u'u/2}\, \mathrm{d}u}$$

where $\mathcal{S}'_x = \{u \in \mathbb{R}^d : u'u \leqslant 2(f^* - f(x))/T\}$. The latter ratio is now equal to

$$\Pr\{\|U\|^2 \leqslant 2(f^* - f(x))/T\}$$

where $\|U\|^2$ has a chi-squared distribution with d degrees of freedom. So this probability is equal to p if

$$2(f^* - f(x))/T = \chi^2_{1-p}(d)$$

or, equivalently, if

$$T = \frac{2(f^* - f(x))}{\chi^2_{1-p}(d)}$$

where $\chi^2_{1-p}(d)$ satisfies:

$$\Pr\{\|U\|^2 \leqslant \chi^2_{1-p}(d)\} = p\,.$$

Summarizing, we have that

$$\Pr\{f(Y) > f(x)\} \geqslant p$$

if

$$T = \frac{2(f^* - f(x))}{\chi^2_{1-p}(d)}$$

where $\chi^2_{1-p}(d)$ is the $100(1-p)$ percentile point of the chi-squared distribution with d degrees of freedom. This result gives us an update formula for the temperature. It is perhaps surprising, and certainly fortunate, that the formula does not involve H or r, Thus simplifying its calculation enormously. The remaining question is now: *when* should the temperature be updated. In the interests of efficiency, instead of running Hide-and-Seek at this temperature T for a sufficient number of iterations to nearly reach the limiting distribution π_T, we instead stop and update the temperature whenever the original goal of finding an improving point is achieved. Note that we are assured a sequence of temperatures which is adaptive and decreasing to zero, so that the results of Section 3.2 and in particular the conclusions of Theorem 3.2 and Corollary 3.3 hold.

3.4. SUMMARY AND COMPARISON WITH OTHER METHODS

Recall that the Hide-and-Seek algorithm proceeds as follows: from a starting point in the feasible region S, generate a direction vector from the uniform distribution over the unit sphere. The candidate for the next iteration point is then chosen uniformly from the intersection of S with the line defined by the current iteration point and the direction vector generated. The Metropolis criterion is then used for determining the next iteration point.

The algorithm introduced in Bohachevsky *et al.* (1986) also generates a uniform direction vector from an interior point of the feasible region. Then, the candidate for the next iteration point is however a fixed step size in the direction of this vector. The determination of this step size is not formalized, and no convergence properties of the algorithm are discussed.

Another algorithm from the class Simulated Annealing is discussed in Corana *et al.* (1987). In this case the coordinate directions are chosen as direction vectors, each one in turn, and the step size is chosen uniformly from an interval which only depends on the particular direction vector chosen. Again, no convergence properties are derived.

A third Simulated Annealing algorithm is introduced by Dekker and Aarts (1991). In this paper a similar Markov chain approach is taken as in Section 2 of this paper. The existence of a limiting distribution is proven in the case where the acceptance probability function is the Metropolis criterion. However, convergence of the algorithm to an optimal solution is not established for the inhomogeneous Markov chain, i.e., the case where the temperature changes during the course of the algorithm. The implementation of the algorithm is as follows. In every iteration, with probability $t \leqslant 1$ a point is generated from the uniform distribution over the feasible region S. With probability $1-t$ a local search procedure is started from the current point in the Markov chain, yielding an improvement over the function value of the current iteration point. The temperature remains fixed for a number of iterations depending on the dimension of the problem, before lowering it according to some cooling schedule. One of the main problems in implementing this algorithm is that the generation of the candidate iteration points, being uniform in the feasible region S, is a difficult and time consuming task when the feasible region is more complicated than for instance a hyperrectangle or hypersphere.

4. Implementation and Numerical Results

In this section we will discuss implementation issues and numerical test results for Hide-and-Seek applied to global optimization problems of the form:

$$\max_{x \in S} f(x)$$

where f is a continuous function over S, and where S is a compact body in $\mathbb{R}^d$, whose boundary has Lebesgue measure zero, and where S can be expressed as

$$S = S_1 \cap S_2 \cap S_3$$

where

$$S_1 = \{x \in \mathbb{R}^d : \ell \leqslant x \leqslant u\}$$
$$S_2 = \{x \in \mathbb{R}^d : a_j'x \leqslant b_j, j = 1, \ldots, m_L\}$$
$$S_3 = \{x \in \mathbb{R}^d : c_j(x) \leqslant 0, j = 1, \ldots, m_{NL}\}$$

where ℓ, $u \in \mathbb{R}^d$, $a_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ $(j = 1, \ldots, m_L)$, and c_j is a continuous function on $S_1 \cap S_2$ $(j = 1, \ldots, m_{NL})$.

4.1. IMPLEMENTATION OF HIDE-AND-SEEK

4.1.1. *Starting point*

Step 0 of the Hide-and-Seek algorithm consists mainly of finding a feasible starting point in S. We find this starting point by using the Hide-and-Seek algorithm to solve the following auxiliary optimization problem:

$$\max_{x \in S_1} \left(\sum_{j=1}^{m_L} \min(0, b_j - a_j' x) + \sum_{j=1}^{m_{NL}} \min(0, -c_j(x)) \right).$$

Assuming that there exists a feasible solution to the original problem, the global maximum of the auxiliary problem will have value zero. The set of globally optimal solutions to the auxiliary problem is equal to S, the feasible region of the original problem.

4.1.2. *Direction distribution*

We will use and compare the following direction distributions:

D1. Uniform on the boundary of the unit hypersphere, which is equivalent to $\Theta \sim N(0, I)$.

D2. Uniform on the boundary of the unit hypersphere after *scaling* the variables in such a way that the lowerbound is 0 and the upperbound is 1. This is equivalent to $\Theta \sim N(0, B)$, where $B = \mathrm{diag}((u_1 - \ell_1)^2, \ldots, (u_d - \ell_d)^2)$, in the original problem.

D3. Θ_i uniformly in $(\ell_i - u_i, u_i - \ell_i)$ $(i = 1, \ldots, d)$. This choice is motivated by Kaufman and Smith (1991). They show that this direction distribution is optimal for accelerating the Hit-and-Run algorithm on S_1 with target distribution equal to the uniform distribution on S_1.

4.1.3. *Step size*

In general the set Λ_k will be difficult to determine. In practice we will implement Step 2 by first computing

$$\tilde{\Lambda}_k = \{\lambda \in \mathbb{R} : x_k + \lambda \theta_k \in S_1 \cap S_2\}.$$

This is an easy task since all constraints in S_1 and S_2 are linear. It is clear that $\Lambda_k \subset \tilde{\Lambda}_k$, so we can generate a stepsize uniformly from Λ_k using the acceptance/rejection method on $\tilde{\Lambda}_k$.

4.1.4. *Cooling schedule*

We will use and compare the following cooling schedules:

C1. $t_k(x_0, \ldots, x_k) = \tau(\max_{0 \leq i \leq k} f(x_i))$, where

$$\tau(y) = \frac{2(f^* - y)}{\chi^2_{1-p}(d)} .$$

C2. $t_k(x_0, \ldots, x_k) = \tau(\max_{0 \leq i \leq k} f(x_i), \hat{f}(x_0, \ldots, x_k))$, where

$$\tau(y, \hat{f}) = \frac{2(\hat{f} - y)}{\chi^2_{1-p}(d)}$$

and where $\hat{f}(x_0, \ldots, x_k)$ is a consistent estimator of f^*.

C3. $\tau_k \equiv (0.99)^k$.

C4. $\tau_k \equiv \dfrac{1}{\ln(k+1)}$.

C5. $\tau_k \equiv 0$.

REMARKS.

1. For cooling schedules C1 and C2 we choose the parameter p equal to 0.01, i.e. the probability that the next point is improving is set to 0.99.

2. For cooling schedule C2 we need a consistent estimator of f^*. In accordance with the Adaptive Search algorithms we only update the temperature at a new record value, so that the estimator $\hat{f}$ will also only be updated at a new record value. Denote the sequence of iteration points generated by the algorithm by $X_0, X_1, \ldots$, and the corresponding sequence of function values by $Y_0, Y_1, \ldots$. Let $Y_{(k)}$ and $Y_{(k-1)}$ be the largest and second largest order statistics of the sequence $Y_0, \ldots, Y_k$. Then we choose the estimator as follows:

$$\hat{f}(X_0, \ldots, X_k) = Y_{(k)} + \frac{Y_{(k)} - Y_{(k-1)}}{(1-q)^{-d/2} - 1} .$$

This estimator is the right endpoint of a $100(1-q)\%$-confidence interval for the maximum value of the sequence $f(X_0)$, $f(X_1), \ldots$, estimated on the basis of the first $k+1$ observations, under some conditions on the random variables $f(X_i)$ (see De Haan, 1981). In the general case this estimator is consistent.

3. The Hide-and-Seek algorithm with cooling schedule C5 corresponds to Improving Hit-and-Run (see Zabinsky *et al.*, 1993).

4.2. EMPIRICAL TEST RESULTS

In the Appendix we describe our testproblems in detail. These test problems consist of a combination of linearly and nonlinearly constrained problems taken from Aluffi-Pentini *et al.* (1985), Ballard *et al.* (1974), Dixon and Szegö (1978) and Timmer (1984). This section shows the results of running the various versions of the Hide-and-Seek algorithm discussed above on problems 1–19 in the Appendix. The results are all averages over 20 runs, where each run was stopped if a point was found whose function value was within 1% of the optimal value (if the optimal value was 0, we used a critical value of −0.01). Tables I–III show the problem number 'p', average number of function evaluations 'f.e.', constraint function evaluations 'c.e.' (if applicable), new points in the sequence 'new' (i.e.

Table I. Results for nonlinearly constrained problems

	Hide-and-Seek with alternate cooling C and direction distribution D							Best other	
p	f.e.	c.e.	new	rec	time	best		f.e.	c.e.
1	0.0	10.1	4.3	3.5	0.10				
	530.6	1,866.2	24.9	14.8	9.87				
	530.6	1,876.3	29.2	18.3	9.97	C1	D3		
2	0.0	158.1	19.2	14.8	1.17				
	281.9	13,735.0	128.8	30.7	44.30				
	281.9	13,893.1	148.0	45.5	45.46	C1	D3		
3	0.0	10.6	1.8	1.8	0.04				
	197.5	541.0	10.8	10.8	3.16				
	197.5	551.6	12.6	12.6	3.20	C5	D1	487[1]	1,057
4	0.0	10.4	2.2	2.2	0.05				
	105.6	1,500.5	31.9	31.9	8.06				
	105.6	1,510.9	34.1	34.1	8.11	C5	D3	739[1]	1,322
5	0.0	8.2	1.6	1.3	0.04				
	288.4	541.2	14.7	11.0	4.65				
	288.4	549.4	16.2	12.3	4.69	C1	D2	454[1]	440

[1] Multi Level Single Linkage, using a penalty function for handling the constraints (see Timmer, 1984).

Table II. Results for linearly constrained problems

p	Hide-and-Seek with alternate cooling C and direction distribution D						
	f.e.	c.e.	new	rec	time	best	
6	0.0	3.6	0.5	0.5	0.02		
	174.4	0.0	36.4	36.4	2.05		
	174.4	3.6	36.9	36.9	2.07	C5	D1
7	0.0	2.6	1.1	1.1	0.02		
	79.0	0.0	18.4	18.4	0.56		
	79.0	2.6	19.5	19.5	0.58	C5	D3
8	0.0	2.5	0.8	0.8	0.02		
	95.7	0.0	19.9	19.9	0.96		
	95.7	2.5	20.8	20.8	0.98	C5	D3

Table III. Results for box-constrained problems

	Hide-and-Seek with alternate cooling C and direction distribution D						Best Other
	f.e.	new	rec	time	best		f.e.
9	79.0	22.4	10.8	1.16	C1	D1	197[1]
10[2]	353.2	57.5	23.5	7.40	C1	D3	487[1]
10[3]	394.1	34.7	34.7	9.49	C5	D1	
11	621.2	14.2	12.4	5.58	C3	D2	148[1]
12	717.6	14.9	11.1	5.37	C1	D3	160[4]
13	547.4	29.0	13.0	4.69	C1	D2	
14	237.6	35.3	21.3	2.61	C1	D3	
15	900.5	9.7	9.7	9.37	C5	D3	26,893[5]
16	301.8	21.4	7.2	2.26	C4	D3	5,393[5]
17	738.1	10.2	10.2	5.49	C3	D3	51,285[5]
18	421.1	11.1	11.1	3.02	C5	D3	10,286[5]
19	114.9	11.8	8.4	0.86	C1	D3	3,402[5]

[1] Multi Level Single Linkage, using a penalty function for handling the constraints (see Timmer, 1984).

[2] The global optimum was not found in 35% of the runs. The averages have been taken over the remaining runs.

[3] The global optimum was naot found in 25% of the runs. The averages have been taken over the remaining runs.

[4] Random direction method (see Bremmerman, 1970).

[5] Method based on stochastic differential equations (see Aluffi-Pentini *et al.* 1985).

number of iterations in which the candidate point was accepted), number of record values 'rec' and time (in seconds) on an Olivetti M24 personal computer. The results are only shown for the best algorithm on each problem, characterized by 'best' cooling schedule and direction distribution. For the nonlinearly and linearly constrained problems the first line in the table corresponds to the first phase (of finding a feasible solution), and the second line to the second phase (i.e. solving the actual problem). The third line is an aggregation of the first two. Where available, we have also shown the number of function (and constraint) evaluations for the best results reported in the literature for a particular problem. We should be careful in comparing the algorithms in this way however, since different algorithms use different stopping rules, and since each algorithm is usually tailored for a restricted class of optimization problems. From Tables I–III we can conclude that the algorithm is remarkably robust to problem type and performs quite well in comparison with other methods from the literature. In all but one case the optimal solution was found to within the desired accuracy in a modest amount of time and function evaluations. The only testproblem where the global optimum was not always found is the Hartman 6 testproblem. The objective function of this testproblem has only 4 local optima, and all of them correspond to very sharp peaks in the graph of the function. In general, we can say that problems of this type pose a special problem for the algorithm. If the sequence of points moves in the direction of one of the non-global local optima, it proves to be very difficult (although theoretically not impossible: we know that we will get arbitrary close to the global optimum eventually) to escape from there, since it requires a very large decrease in function value *or* a large number of small decreases. As to the cooling schedules, C1 or C5 clearly performed best. The good performance of cooling schedule C1 in comparison with cooling schedules C3 and C4 suggest that much can be gained from using an adaptive cooling schedule, which takes into account the value of the best solution found. However, cooling schedule C2 did not perform as well as might be expected. The only reason for this can be in the accuracy of the estimator of the global maximum. Therefore, an important subject of future research is finding better estimators of the global maximum. The good performance of the simple cooling schedule C5 seems remarkable. At this point we cannot explain this behaviour.

4.3. EFFECTS OF PROBLEM DIMENSIONALITY

In this section we consider Testproblem 20 (see the Appendix). This testproblem is posed for general dimension, so we can use this testproblem to investigate the behavior of the algorithms for increasing dimension. In particular, we can test whether the observed number of record values grows linearly in the dimension of the problem. From the complexity result of the Pure Adaptive Search algorithm we can hope for this. However, as noted before, the Hide-and-Seek algorithm is only an approximate implementation of this algorithm. Table IV shows, for various dimensions, the average number of record values (over 10 runs) for the Hide-and-Seek algorithm with cooling schedules C1 and C5. In all cases we used the uniform distribution on the boundary of the unit sphere as direction distribution. For $d = 5$ the algorithms were stopped as soon as a point with value larger than -0.01 was found (the optimal value of the problem is 0). Since the range of the objective function increases approximately linear in the dimension of the problem, we also let this critical value change linearly in the dimension of the problem. Linear regressions of the data in Table IV gives the following least squares fits:

$$N_1(d) = 2.45 + 4.62d$$
$$N_5(d) = -10.2 + 6.29d$$

where $N_i(d)$ is the number of records found for a problem of dimension d, using cooling schedule Ci. The linear fit for both equations is excellent, yielding a

Table IV. Results for problem 20

d	C1	C5
5	23.2	23.1
10	52.8	54.5
15	70.4	77.8
20	94.3	115.5
25	117.9	149.8

percent variation explained of 99.5% for each. Thus, for this testproblem, the property of linear complexity in record values appears to hold, suggesting that Hide-and-Seek is a successful implementation of Pure Adaptive Search.

5. Concluding Remarks

In this paper we have introduced a new simulated annealing algorithm for global optimization. Experiments indicate that our method performs quite well as compared to other algorithms from the literature on a set of standard testproblems. Moreover, the Hide-and-Seek algorithm is easily implemented for problems having non-convex, or even disconnected, feasible regions. This is in contrast with many other methods, which often require for example problem specific local search algorithms. Another conclusion from the experiments is that the algorithm appears to inherit some of the attractive theoretical properties of the Pure Adaptive Search algorithm, and in particular linear time complexity in improving points in the dimension of the problem.

Acknowledgement

We are indebted to Claude Bélisle for many important insights.

Appendix: Test Problems

NONLINEARLY CONSTRAINED PROBLEMS

1. Objective function:

$$f(x) = -\frac{1}{x_1^2 x_2 x_3}$$

Feasible region:

$$\begin{aligned}&0.44098x_1 + 28.46x_1^2 + 6158.4x_1^2x_2 + 0.0037018x_3 + 5.4474x_3^2\\&\quad + 0.032236x_1x_3 + 2.92x_2x_3 + 0.44712x_2 + 37.964x_2^2 + 42.876x_1x_2 - 1 \leq 0\end{aligned}$$

$$0 \leq x_1 \leq 0.18745$$
$$0 \leq x_2 \leq 0.16230$$
$$0 \leq x_3 \leq 0.42846$$

2. Objective function:

$$\begin{aligned}f(x) = &-(0.0204 + 0.0607x_5^2)\cdot x_1x_4\cdot(x_1 + x_2 + x_3)\\&- (0.0187 + 0.0437x_6^2)\cdot x_2x_3\cdot(x_1 + 1.57x_2 + x_4)\end{aligned}$$

Feasible region:

$$\frac{2070}{x_1x_2x_3x_4x_5x_6} - 1 \leqslant 0$$

$$0.00062x_1x_4x_5^2 \cdot (x_1 + x_2 + x_3) + 0.00058x_2x_3x_6^2 \cdot (x_1 + 1.57x_2 + x_4) - 1 \leqslant 0$$

$$0 \leqslant x_1 \leqslant 10$$
$$0 \leqslant x_2 \leqslant 10$$
$$0 \leqslant x_3 \leqslant 15$$
$$0 \leqslant x_4 \leqslant 15$$
$$0 \leqslant x_5 \leqslant 1$$
$$0 \leqslant x_6 \leqslant 1$$

3. Objective function:

$$f(x) = \sum_{i=1}^{5} \frac{1}{a_i(x - p_i)'(x - p_i) + c_i}$$

Feasible region:

$$x_1 + x_2 \leqslant 5$$
$$x_1 - x_2^2 \leqslant 0$$
$$5x_1^3 - \tfrac{8}{5}x_2^2 \leqslant 0$$

$$-3 \leqslant x_1 \leqslant 10$$
$$-4 \leqslant x_2 \leqslant 7$$

Data:

i	a_i	p_i		c_i
1	0.5	0	5	0.125
2	0.25	2	5	0.25
3	1	3	2	0.1
4	$\frac{1}{12}$	4	4	0.2
5	2	5	1	$\frac{1}{12}$

4. Objective function:

$$f(x) = x_1^2 + x_2^2$$

Feasible region:

$$x_1 + x_2 \leqslant 2$$
$$x_1^2 - x_2 \leqslant 0$$
$$-3 \leqslant x_1 \leqslant 2$$
$$0 \leqslant x_2 \leqslant 5$$

5. Objective function:

$$f(x) = -(x_2 - 1.275x_1^2 + 5x_1 - 6)^2 - 10\left(1 - \frac{1}{8\pi}\right)\cos(\pi x_1) - 10$$

Feasible region:

$$-\pi x_1 - x_2 \leqslant 0$$
$$-\pi^2 x_1^2 + 4x_2 \leqslant 0$$
$$-1\tfrac{1}{2} \leqslant x_1 \leqslant 3\tfrac{1}{2}$$
$$0 \leqslant x_2 \leqslant 15$$

LINEARLY CONSTRAINED PROBLEMS

6. Objective function:

$$f(x) = -2x_1 - 6x_2 + x_1^3 + 8x_2^2$$

Feasible region:

$$x_1 + 6x_2 \leqslant 6$$
$$5x_1 + 4x_2 \leqslant 10$$
$$0 \leqslant x_1 \leqslant 2$$
$$0 \leqslant x_2 \leqslant 1$$

7. Objective function

$$f(x) = -\frac{1}{2}\left(\frac{0.1211}{x_2} + \frac{1.11 \cdot 10^{-6}}{x_1 x_2}\right)$$

Feasible region:

$$x_1 + x_2 \leqslant 0.12321$$
$$0 \leqslant x_1 \leqslant 0.12321$$
$$0 \leqslant x_2 \leqslant 0.12321$$

8. Objective function:

$$f(x) = (x_1 - \tfrac{3}{4})^2 + (\tfrac{1}{2}x_2 - \tfrac{3}{4})^2$$

Feasible region:

$$x_1 + \tfrac{1}{2}x_2 \leqslant 1$$

$$0 \leqslant x_1 \leqslant 1$$
$$0 \leqslant x_2 \leqslant 2$$

BOX-CONSTRAINED PROBLEMS

9. *Hartmann 3*

Objective function:

$$f(x) = \sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(\frac{1}{j} x_j - p_{ij}\right)^2\right)$$

Feasible region:

$$0 \leqslant x_j \leqslant j \quad j = 1, 2, 3$$

Data:

i	c_i	a_i			p_i		
1	1	3	10	30	0.3689	0.1170	0.2673
2	1.2	0.1	10	35	0.4699	0.4387	0.7470
3	3	3	10	30	0.1091	0.8732	0.5547
4	3.2	0.1	10	35	0.03815	0.5743	0.8828

10. *Hartmann 6*

Objective function:

$$f(x) = \sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(\frac{1}{j} x_j - p_{ij}\right)^2\right)$$

Feasible region:

$$0 \leqslant x_j \leqslant j \quad j = 1, \ldots, 6$$

Data:

i	c_i	a_i					
1	1	10	3	17	3.5	1.7	8
2	1.2	0.05	10	17	0.1	8	14
3	3	3	3.5	1.7	10	17	8
4	3.2	17	8	0.05	10	0.1	14

i	p_i					
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

11. *Goldstein/Price*
Objective function:

$$\begin{aligned} f(x) = -(1 + (x_1 + 2x_2 + 1)^2 \\ \times (19 - 14x_1 + 3x_1^2 - 28x_2 + 12x_1x_2 + 12x_2^2))(30 + (2x_1 - 6x_2)^2 \\ \times (18 - 32x_1 + 12x_1^2 + 96x_2 - 72x_1x_2 + 108x_2^2)) \end{aligned}$$

Feasible region:

$$-2 \leqslant x_1 \leqslant 2$$
$$-1 \leqslant x_2 \leqslant 1$$

12. *Branin*
Objective function:

$$f(x) = -(x_2 - 1.275x_1^2 + 5x_1 - 6)^2 - 10\left(1 - \frac{1}{8\pi}\right) \cos(\pi x_1) - 10$$

Feasible region:

$$-1\tfrac{1}{2} \leqslant x_1 \leqslant 3\tfrac{1}{2}$$
$$0 \leqslant x_2 \leqslant 15$$

13. Objective function:

$$f(x) = -(9x_1^2 + 36x_1x_2 + 52x_2^2 + 30(x_1^2 + 4x_2^2 + 2x_1 - 16)^2)$$

Feasible region:

$$-4 \leqslant x_1 \leqslant 4$$
$$2 \leqslant x_2 \leqslant 2$$

14. Objective function:

$$f(x) = -\left(2.1 \cdot 10^{-11} x_2^{2.55} + 6.29 \cdot 10^7 \frac{x_2^5}{x_3^6} + \frac{8.5 \cdot 10^{10}}{x_1^2 x_2 x_3^{0.2}} + 1.6 \cdot 10^5 \frac{x_1^{2.5} x_3}{x_2}\right)$$

Feasible region:

$$0 \leqslant x_1 \leqslant 1$$
$$0 \leqslant x_2 \leqslant 6 \cdot 10^5$$
$$0 \leqslant x_3 \leqslant 3 \cdot 10^5$$

15. *Two-dimensional Shubert function*
Objective function:

$$f(x) = -\left(\sum_{i=1}^{5} i \cos((i+1)x_1 + 1)\right)\left(\sum_{i=1}^{5} i \cos(5(i+1)x_2 + 1)\right)$$

Feasible region:

$$-10 \leqslant x_1 \leqslant 10$$
$$-2 \leqslant x_2 \leqslant 2$$

16. *Camel function*
Objective function:

$$f(x) = -(4 - 2.1x_1^2 + \tfrac{1}{3}x_1^4)x_1^2 - x_1x_2 + 4(1 - x_2^2)x_2^2$$

Feasible region:

$$-4 \leqslant x_1 \leqslant 4$$
$$-4 \leqslant x_2 \leqslant 4$$

17. Objective function:

$$f(x) = -10^5 x_1^2 - x_2^2 + (x_1^2 + x_2^2)^2 - 10^{-5}(x_1^2 + x_2^2)^4$$

Feasible region:

$$-20 \leqslant x_1 \leqslant 20$$
$$-20 \leqslant x_2 \leqslant 20$$

18. Objective function:

$$f(x) = -\tfrac{1}{2}x_1^2 - \tfrac{1}{2}(1 - \cos(2x_1)) - x_2^2$$

Feasible region:

$$-10 \leqslant x_1 \leqslant 10$$
$$-10 \leqslant x_2 \leqslant 10$$

19. Objective function:

$$f(x) = -\tfrac{1}{4}x_1^4 + \tfrac{1}{2}x_1^2 - \tfrac{1}{10}x_1 - \tfrac{1}{2}x_2^2$$

Feasible region:

$$-2 \leqslant x_1 \leqslant 2$$
$$-2 \leqslant x_2 \leqslant 2$$

20. Objective function:

$$f(x) = -0.1 \cdot \left(\sin^2(3\pi x_1) + \sum_{j=1}^{d-1} (x_j - 1)^2(1 + \sin^2(3\pi x_{j+1})) + (x_d - 1)^2(1 + \sin^2(2\pi x_d))\right)$$

Feasible region:

$$-10 \leq x_j \leq 10 \quad j = 1, \ldots, d$$

References

Aarts, E. H. L. and J. H. M. Korst (1988), *Simulated Annealing and Boltzmann Machines*, Wiley, Chichester.

Aarts, E. H. L. and P. J. M. van Laarhoven (1989), Simulated annealing: an introduction, *Statistica Neerlandica* **43**, 31–52.

Aluffi-Pentini, F., V. Parisi, and F. Zirilli (1985), Global optimization and stochastic differential equations, *Journal of Optimization Theory and Applications* **47**, 1–16.

Ballard, D. H., C. O. Jelinek, and R. Schinzinger (1974), An algorithm for the solution of constrained generalized polynomial programming problems, *The Computer Journal* **17**, 261–266.

Bélisle, C. J. P. (1992), Convergence theorems for a class of simulated annealing algorithms on $\mathbb{R}^d$, *Journal of Applied Probability* **29**, 885–895.

Bélisle, C. J. P., H. E. Romeijn, and R. L. Smith (1993), Hit-and-run algorithms for generating multivariate distributions, *Mathematics of Operations Research* **18**, 255–266.

Berbee, H. C. P., C. G. E. Boender, A. H. G. Rinnooy Kan, C. L. Scheffer, R. L. Smith, and J. Telgen (1987), Hit-and-run algorithms for the identification of nonredundant linear inequalities, *Mathematical Programming* **37**, 184–207.

Boender, C. G. E., R. J. Caron, A. H. G. Rinnooy Kan, J. F. McDonald, H. E. Romeijn, R. L. Smith, J. Telgen, and A. C. F. Vorst (1991), Shake-and-Bake algorithms for generating uniform points on the boundary of bounded polyhedra, *Operations Research* **39**, 945–954.

Bohachevsky, I. O., M. E. Johnson, and M. L. Stein (1986), Generalized simulated annealing for function optimization, *Technometrics* **28**, 209–217.

Bremmerman, H. (1970), A method of unconstrained global optimization, *Mathematical Biosciences* **9**, 1–15.

Corana, A., M. Marchesi, C. Martini, and S. Ridella (1987), Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm, *ACM Transactions on Mathematical Software* **13**, 262–280.

De Haan, L. F. M. (1981), Estimation of the minimum of a function using order statistics, *Journal of the American Statistical Association* **76**, 467–469.

Dekker, A. and E. H. L. Aarts (1991), Global optimization and simulated annealing, *Mathematical Programming* **50**, 367–393.

Dixon, L. C. W. and G. P. Szegö (1978), *Towards Global Optimization 2*, North-Holland, Amsterdam, The Netherlands.

Doob, J. L. (1953), *Stochastic Processes*, Wiley, New York, NY.

Hajek, B. (1988), Cooling schedules for optimal annealing, *Mathematics of Operations Research* **13**, 311–329.

Hwang, C. (1980), Laplace's method revisited: weak convergence of probability measures, *The Annals of Probability* **8**, 1177–1182.

Kaufman, D. E. and R. L. Smith (1991), Optimal direction choice for Hit-and-Run sampling, Technical Report 90-08, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan.

Kirkpartick, S., C. D. Gelatt Jr., and M. P. Vecchi (1983), Optimization by simulated annealing, *Science* **20**, 671–680.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equations of state calculations by fast computing machines, *The Journal of Chemical Physics* **21**, 1087–1092.

Nummelin, E. (1984), *General Irreducible Markov Chains and Non-Negative Operators*, Cambridge University Press, Cambridge, U.K.

Patel, N. R., R. L. Smith, and Z. B. Zabinsky (1988), Pure adaptive search in Monte Carlo optimization, *Mathematical Programming* **43**, 317–328.

Pncus, M. (1968), A closed form solution for certain programming problems, *Operations Research* **16**, 690–694.

Pincus, M. (1970), A Monte-Carlo method for the approximate solution of certain types of constrained optimization problems, *Operations Research* **18**, 1225–1228.

Rinnooy Kan, A. H. G. and G. T. Timmer (1984), Stochastic methods for global optimization, *American Journal of Mathematical and Management Sciences* **4**, 7–40.

Rinnooy Kan, A. H. G. and G. T. Timmer (1987a), Stochastic global optimization methods; part I: clustering methods, *Mathematical Programming* **39**, 27–56.

Rinnooy Kan, A. H. G. and G. T. Timmer (1987b), Stochastic global optimization methods; part II: multilevel methods, *Mathematical Programming* **39**, 57–78.

Romeijn, H.E. and R. L. Smith (1990), Sampling through random walks. Technical Report 90-02. Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI.

Rubinstein, R. Y. (1981), *Simulation and the Monte Carlo Method*, Wiley, New York, NY.

Smith, R. L. (1984), Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions, *Operations Research* **32**, 1296–1308.

Timmer, G. T. (1984), *Global optimization: a stochastic approach.* PhD thesis, Erasmus University Rotterdam, Rotterdam, The Netherlands.

Zabinsky, Z. B. and R. L. Smith (1992), Pure adaptive search in global optimization, *Mathematical Programming* **53**, 323–338.

Zabinsky, Z. B., R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman (1993), Improving Hit-and-Run for global optimization, *Journal of Global Optimization* **3**, 171–192.